



Article

Integrity Authentication Based on Blockchain and Perceptual Hash for Remote-Sensing Imagery

Dingjie Xu ^{1,2,3} , Na Ren ^{1,2,3,4,*} and Changqing Zhu ^{1,2,3}

¹ Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing 210023, China

² State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing 210023, China

³ Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

⁴ Hunan Engineering Research Center of Geographic Information Security and Application, Changsha 410000, China

* Correspondence: 09359@njnu.edu.cn; Tel.: +86-136-1157-8959

Abstract: The integrity of remote-sensing image data is susceptible to corruption during storage and transmission. Perceptual hashing is a non-destructive data integrity-protection technique suitable for high-accuracy requirements of remote-sensing image data. However, the existing remote-sensing image perceptual hash-authentication algorithms face security issues in storing and transmitting the original perceptual hash value. This paper proposes a remote-sensing image integrity authentication method based on blockchain and perceptual hash to address this problem. The proposed method comprises three parts: perceptual hash value generation, secure blockchain storage and transmission, and remote-sensing image integrity authentication. An NSCT-based perceptual hashing algorithm that considers the multi-band characteristics of remote-sensing images is proposed. A Perceptual Hash Secure Storage and Transmission Framework (PH-SSTF) is designed by combining Hyperledger Fabric and InterPlanetary File System (IPFS). The experimental results show that the method can effectively verify remote-sensing image integrity and tamper with the location. The perceptual hashing algorithm exhibits strong robustness and sensitivity. Meanwhile, the comparison results of data-tampering identification for multiple landscape types show that the algorithm has stronger stability and broader applicability compared with existing perceptual hash algorithms. Additionally, the proposed method provides secure storage, transmission, and privacy protection for the perceptual hash value.

Keywords: perceptual hash; blockchain; remote-sensing image; integrity authentication; Hyperledger Fabric



Citation: Xu, D.; Ren, N.; Zhu, C. Integrity Authentication Based on Blockchain and Perceptual Hash for Remote-Sensing Imagery. *Remote Sens.* **2023**, *15*, 4860. <https://doi.org/10.3390/rs15194860>

Academic Editor: Chaowei Yang

Received: 1 August 2023

Revised: 26 September 2023

Accepted: 5 October 2023

Published: 7 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Remote-sensing imagery, which records the magnitude of electromagnetic waves from the Earth's surface, provides precise spatial location information and multispectral data. It has found widespread application in various civilian and military domains, such as disaster assessment, geological exploration, and target detection [1,2]. However, the security of remote-sensing images has emerged as a growing concern. During storage and transmission, these images are vulnerable to tampering, which can compromise the integrity of the data, including its accuracy and authenticity. The alteration of data containing sensitive information, such as national security or court evidence, can have significant consequences [3]. Therefore, it is imperative to ensure the security of remote-sensing images and provide technical support to prevent malicious tampering, enabling more precise and reliable analysis of remote-sensing data for scientific and practical applications.

There are currently three main techniques for data integrity authentication. The first technique is digital signature, in which the sender generates a digest of the data using a cryptographic hash function. The digest is then encrypted with a private key to produce a digital signature, which is sent along with the original data to the recipient. The recipient decrypts the digital signature using the sender's public key and compares the resulting digest to a digest generated from the original data. If the two digests match, the data integrity is verified [4]. However, the cryptographic hash function used in digital signatures has an avalanche effect, where any change in one bit of the data is considered tampering. This means that these methods only assess the consistency of the binary representation of the data, not the content consistency. In the process of storing and transmitting remote-sensing images, data undergo content-preserving operations, such as format conversion and compression, which can result in changes at the bit level but do not alter the underlying information. Digital signature technology is not robust to these content-preserving operations and cannot determine the location of tampering, making it unsuitable for remote-sensing image integrity authentication.

The second type of data integrity authentication technique is semi-fragile watermarking. This technique is designed to tolerate content-preserving image processing operations, such as compression and noise while being able to detect changes caused by malicious tampering. Some scholars have proposed some semi-fragile watermarking schemes specifically for remote-sensing image integrity authentication [5–8]. In particular, Serra-Ruiz proposed a series of semi-fragile watermarking algorithms using a tree-structured vector quantization (TSVQ) approach [6–8]. These algorithms involve tiling the original image into blocks of varying sizes, applying the discrete wavelet transform to each selected spectral band, and building a TSVQ tree for each block. An iterative algorithm is applied to modify the tree until it meets a required criterion, and a secret key produces a different criterion for each block to avoid copy-and-replace attacks. These algorithms can detect possibly forged blocks and their position in the whole image while also maintaining robustness against near-lossless compression attacks.

Although semi-fragile watermarking can provide reliable performance and precise tamper localization, it may not be the best approach for remote-sensing applications where high accuracy is critical. Remote-sensing image is characterized by high precision, which is a fundamental feature of geospatial data. High-precision remote-sensing images may contain sensitive information, such as national defense infrastructure, which must not suffer any accuracy loss. Unfortunately, the watermark embedding process modifies the original data, potentially leading to inaccuracies [9]. Therefore, alternative techniques that can preserve the high accuracy of remote-sensing images while also being efficient are essential for various scientific and practical applications.

The third technique for data integrity authentication is perceptual hashing (PH), which inherits the one-way, anti-collision, and digest properties of traditional cryptographic hash functions [10]. PH maps input data of arbitrary length into a compact hash value, making it computationally infeasible to find two inputs that produce the same output. Unlike cryptographic hash functions that perform authentication at the binary level, PH considers data changes from the perspective of the perceived content rather than the binary representation, making it robust to content-preserving operations such as format conversion and compression without modifying the original data. Tamper localization can be performed through image grid segmentation, providing a precise tamper location. In comparison to digital signatures and semi-fragile watermarking techniques, PH techniques satisfy the requirements for remote-sensing images for lossless accuracy, robustness to content-preserving operations, and precision in tamper localization.

In recent years, researchers have put forward a range of perceptual hash algorithms to achieve the integrity authentication of remote-sensing images. Zhang et al. introduced a robust high-resolution remote-sensing (HRRS) image integrity authentication algorithm based on perceptual hashing techniques that account for both global and local features [11]. The algorithm extracts global features through the efficient recognition capability of Zernike

moments for texture information, while FAST key points are used for local feature construction and tamper localization. Additionally, Ding et al. proposed various perceptual hash algorithms for remote-sensing image integrity authentication, including perceptual hash algorithms for multispectral remote-sensing images and HRRS images based on edge features [9,12]. Recently, the author presented a novel attention-based Asymmetric U-Net (AAU-Net) for subject-sensitive hashing of remote-sensing images [13]. The robustness of the AAU-Net-based subject-sensitive hashing algorithm is stronger than algorithms based on U-Net and MUM-Net previously proposed by the author [14,15]. Previous research on perceptual hashing of remote-sensing images has focused on enhancing algorithm robustness and achieving a good balance between robustness and tamper sensitivity.

However, the security of remote-sensing image authentication information, specifically the perceptual hash value itself, has not been addressed in the most recent research to date [16,17]. This oversight can lead to two primary security issues. First, malicious attackers can aim to deceive the receiver by tampering with the genuine message sent by the sender or impersonating the sender to transmit a false message. Second, a mutual distrust problem may arise between the sender and the receiver, leading to mutual deception, denial, and other related issues. It is crucial to address the security of authentication information to ensure the reliability and authenticity of communication between the sender and receiver. Hence, there is a pressing need for new technology to ensure the trusted transmission and security management of the perceptual hash value.

The advent of blockchain technology has established a novel trust paradigm characterized by the non-tamperability of data, transparency of information, decentralized collective maintenance, and traceability. These attributes provide a fresh approach to data integrity authentication and information security management [18]. It stores data in a distributed manner, eliminating the vulnerabilities of centralized storage and offering resistance to single points of failure. The inherent hash pointer structure of blockchain ensures data integrity, making it a suitable choice for securely storing perceptual hash values.

Blockchain has gained substantial traction in various industries, including healthcare, the Internet of Things, and financial activities [19]. However, the unique sensitivity of remote-sensing imagery in critical applications, such as national security and land management, demands specific industry standards for secure sharing platforms. Government and regulatory agencies are likely to mandate such standards for remote-sensing image authentication. Consequently, the direct application of existing approaches from other industries to remote-sensing data security may have limitations due to the domain's distinct considerations.

In this paper, a remote-sensing image integrity authentication method is proposed that combines blockchain technology with perceptual hashing. The method is designed to address the need for remote-sensing image data integrity authentication with robustness to content-preserving operations and to ensure the security of authentication information. The main contributions of the paper are as follows:

1. A Perceptual Hash Secure Storage and Transmission Framework (PH-SSTF) is designed to realize the secure storage and transmission of the original perceptual hash values by innovatively combining the private IPFS network and Hyperledger Fabric. This framework fills the gap in the existing research for the secure protection of remote-sensing image authentication information.
2. A prototype system was implemented using PH-SSTF, and its practicality and scaling value were verified through rigorous testing. The proposed framework enables a higher level of data security and privacy protection, faster data storage and retrieval, and elastic storage and capacity scaling. These results demonstrate the effectiveness and potential of the proposed approach.
3. A remote-sensing image perception hashing algorithm with wider applicability is proposed. The algorithm takes into account the unique multi-band features of remote-sensing images, uses the Non-Subsampled Contourlet Transform (NSCT) to achieve

multi-band feature fusion and feature extraction, generates an encrypted perceptual hash, and improves the efficiency of hash generation.

The rest of the paper is structured as follows: Section 2 provides an overview of the basic concepts and preliminaries, Section 3 details the proposed methods, Section 4 presents and analyzes the experimental results, Section 5 examines the effect of parameter choices on the method and conducts a security analysis and comparative discussion. Finally, Section 6 concludes the paper.

2. Basic Idea and Preliminaries

2.1. Basic Idea

In this paper, a novel remote-sensing image integrity authentication method that incorporates blockchain technology and perceptual hashing is proposed. The general concept of the method is illustrated in Figure 1. The proposed method consists of three stages: generation of the perceptual hash value, secure storage and transmission via the PH-SSTF, and integrity authentication of the remote-sensing image. First, the perceptual hash value is calculated by applying the perceptual hashing algorithm to the original remote-sensing image. During transmission or distribution, the resulting hash value is then bound with the metadata and transmission information of the image and stored on the PH-SSTF. After the transmission or distribution is completed, the perceptual hash value is recomputed and compared with the original hash value stored on the blockchain. If the difference between the two hash values is below a certain threshold T , the image data are considered to be intact; otherwise, tampering is detected, and the affected regions can be pinpointed. This process will be described in more detail in Section 3.

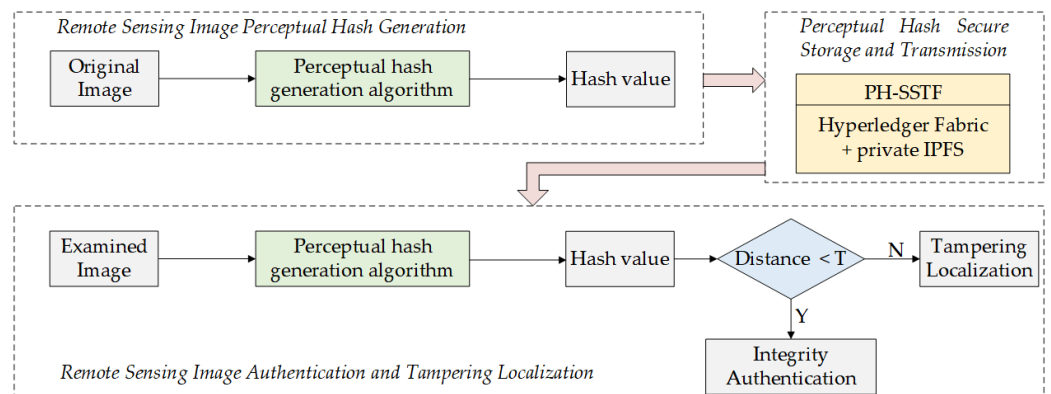


Figure 1. Remote-sensing image integrity authentication method.

2.2. NSCT-Based Remote-Sensing Image Perceptual Hash Feature Extraction

Compared to the conventional wavelet transform, the nonsubsampled contourlet transform (NSCT) is known for its superior ability to preserve the edge and texture information of images. NSCT is particularly effective in handling non-smooth signals due to its adaptive and multiscale representation of image features. The NSCT has gained significant recognition in the fields of remote-sensing image change detection and image data fusion, as demonstrated by numerous studies [20,21]. The NSCT transform retains the multiscale decomposition, multi-directionality, and anisotropy of the contour wave transform, eliminates the down-sampling and up-sampling operations in the decomposition and reconstruction of image information, and makes the size of each sub-band image identical to the original image. NSCT eliminates the Pseudo-Gibbs effect, thus making the transform translation invariant [22]. Therefore, this paper designs the perceptual hashing algorithm based on NSCT's method of extracting spatial features of remote-sensing images.

2.2.1. Spatial Feature Extraction of Remote-Sensing Images Based on NSCT

Texture features are essential spatial features of remote-sensing images that measure pixel relationships within a local area and reflect the distribution patterns of grayscale values of neighboring pixels. They provide crucial information about the structure and organization of object surfaces and their connection with the surrounding environment.

The NSCT low-frequency sub-band coefficients contain most of the information in the source image, concentrate the overall energy of the source image, and are an approximate representation of the source image [23,24]. It reflects the average features of the image and can determine the general contour of the image. In this paper, the low-frequency sub-band coefficients are used as the main carrier of remote-sensing image perception hash space feature extraction to improve the algorithm's robustness to resist content-preserving operations such as filtering and sharpening.

Additionally, the high-frequency sub-band coefficients also contain some detail and texture information of the image, and tampering with the high-frequency sub-bands may only lead to detailed or even texture changes in the reconstructed image by the inverse NSCT transform. However, from a perceptual standpoint, such attacks, unlike direct tampering of the original image, can be categorized as either invalid tampering or valid tampering based on the extent of change in the reconstructed image, where invalid tampering can be regarded as content-preserving operations. Specific experiments and discussions will be conducted in Section 5.1.3 of the paper.

Remote-sensing images are multi-band in nature, and detecting any tampering of valuable content within each band is essential. In practical scenarios, the tamperer may try to modify only certain bands of the image without drawing too much attention to hide or camouflage their tampering behavior. Some applications may be more sensitive to information in specific bands (e.g., near-infrared band applications with water body identification), so tampering with specific bands can be more difficult to detect but may have a significant impact on the application results. In the case of tampering at the band granularity level, the existing perceptual hashing algorithm needs to authenticate and detect each band once, which is time-consuming and inefficient for multispectral or even hyperspectral images. To address this issue, this study adopts the NSCT transform for multi-band feature fusion as a preprocessing step, taking into account the multispectral nature of remote-sensing images. This preprocessing step helps to reduce the redundancy of waveband information and enhances the computational efficiency of perceptual hashing.

Therefore, this study proposes the use of statistical features based on the fused NSCT low-frequency sub-band coefficients from each waveband of the remote-sensing image as texture features. The perceptual hash value is calculated based on the low-frequency sub-band coefficients after sub-band fusion using standard deviation and other statistical features.

2.2.2. NSCT Sub-Band Fusion of Multispectral Remote-Sensing Images

Unlike the traditional remote-sensing image fusion, which combines remote-sensing image data from different sensors or different resolutions into a new image to obtain more comprehensive and accurate information, the low-frequency sub-band coefficient fusion of NSCT in this paper is a preprocessing process to generate the perceptual hash of the image. For the tampering identification needs at the granularity level of remote-sensing image bands, the fusion method in this paper needs to balance the two requirements of high efficiency and sensitivity to tampering in each band. Therefore, compared with the common fusion methods such as Principal Component Analysis (PCA), IHS/HSV conversion, and Wavelet Transform, Weighted Average, which is more efficient, is chosen to meet the requirements of this paper, and the comparison of different methods will be expanded in Section 5.3.2. The weighted average method has more significant changes in the fusion results after the tampering of one band.

The weighted average method is commonly used for low-frequency sub-band fusion, but the existing weighting methods are not directly applicable to the specific purposes

of this paper. The sensitivity of band information tampering requires that bands with a greater impact on the fusion result should be assigned higher weights to better reflect the tampering in the fusion outcome. Hence, it is essential to identify statistical properties where different bands exhibit similar values in regions without tampering. However, tampering in a band could cause significant changes in its statistical values.

In general, images of different bands within the same region should demonstrate consistent statistical properties, including regional energy (RE) and variance. RE refers to the summation or averaging of pixel values, effectively characterizing the overall brightness or energy distribution of a specific region in an image [25]. When a region exhibits consistent surface features with similar reflectance across different bands, it results in a relatively uniform RE. Band tampering may lead to a change in the RE of a band. For example, an image region being modified or replaced may result in a significant change in the energy value of that region. On the other hand, variance reflects the dispersion of pixel values in an image, indicating the degree of difference or variation in these values. Although different bands within the same region usually exhibit similar variance, factors such as varying optical reflectance properties of features, atmospheric effects, and sensor noise may introduce variations when tampering is not present. Consequently, variance is relatively less stable than RE when considering weighting metrics. The stability of RE across bands enables its application as a valuable indicator in detecting potential tampering.

This paper proposes an improved sub-band fusion method that utilizes RE to quantify the significance of low-frequency sub-band coefficients in distinct bands across all band ranges. This method uses the RE of sub-band coefficients as weights to perform a weighted average of coefficients across different sub-bands, resulting in fused coefficients for extracting texture features. The expression for RE in the k th band (with an area size of $M \times N$) is given by Formula (1).

$$E_l^k(i, j) = \sum_{i \in M, j \in N} [C_l^k(i, j)]^2 \quad (1)$$

In the formula, $C_l^k(i, j)$ represents the l -th level low-frequency sub-band coefficients after the decomposition of the k th band, and M and N denote the total rows and columns of the low-frequency sub-band coefficients. $E_l^k(i, j)$ denotes the RE value of the k -th band.

This approach effectively preserves most of the energy of the source image and improves the efficiency of subsequent perceptual hash feature extraction. It should be noted that the region energy is only used as an indicator to assist band tampering detection; it does not directly determine the specific tampering type or location. Therefore, band fusion is only a preprocessing process, and the final integrity verification and localization of tampered regions require perceptual hashing algorithms for further implementation.

2.3. Hyperledger Fabric

Blockchains can be divided into public, private, and consortium blockchains. Table 1 shows the advantages and disadvantages of various blockchains. Private blockchains are under the complete control and management of a single entity or organization, whereas consortium blockchains are governed and controlled by multiple participants [26]. In contrast to private blockchains, consortium blockchains offer a higher degree of decentralization, enabling multiple entities to engage in the network through a distributed approach, therefore enhancing transparency and the system's credibility. Private blockchains are well-suited for managing internal business processes and data within a single entity. In contrast, consortium blockchains facilitate collaborative efforts among various entities across organizations, ensuring data security and consistency. Furthermore, consortium blockchains provide a mechanism for multiple participants to share and access shared data, fostering more efficient business cooperation.

Table 1. Blockchain Classification Summary.

Categories	Advantages	Disadvantages	Use Cases	Delegates
Public	+Independence +Transparency +Trust	–Performance –Scalability –Privacy Security	Cryptocurrency Document validation	BTC ETH Solana
Private	+Access control +Performance	–Trust –Auditability	Supply chain Asset ownership	Multichain
Consortium	+Access control +Scalability +Privacy Security	–Transparency	Banking Research Supply chain	Hyperledger Fabric Corda Quorum

Consortium blockchains provide enhanced data privacy and access control compared to public blockchains. They are proprietary networks involving selected participants from specific organizations, ensuring that only authorized members have access to transactions and data, thus offering strong privacy measures. This fine-grained access control is ideal for safeguarding sensitive information and tailoring access based on roles and identities in specific organizational contexts. In contrast, public blockchains operate as open networks, with all transactions and data being transparent to the public, resulting in weaker privacy measures [27]. Although public blockchains theoretically enhance security through decentralization, high-throughput public blockchains like Solana provide lower-latency data transmission and validation. However, for scenarios involving remote-sensing data, particularly in government contexts related to national security and land management, compliance with specific regulatory requirements is crucial. Consortium blockchains, with their selected and authorized participants, facilitate better traceability and auditing of transactions, ensuring data security and regulatory compliance.

Hyperledger Fabric, an open-source platform designed for constructing enterprise-level distributed applications through consortium blockchains, offers several features that make it highly suitable for secure storage and transmission of perceptual hash in practical applications. It supports pluggable consensus protocols and a common programming language for smart contracts, allowing customization to fit specific business scenarios and trust models. Hyperledger Fabric utilizes Go as the universal language for smart contract deployment, benefiting from its high performance, user-friendliness, concurrency support, and cross-platform compatibility, which provides an efficient, reliable, and flexible development environment for enterprise-level blockchain applications. Fabric's flexible identity verification and access control mechanisms ensure that only authorized participants can access and engage with the blockchain network. The platform's strong scalability enables it to handle large-scale transaction processing and network expansion, making it well-suited for complex enterprise environments. Furthermore, Fabric achieves cost reduction, higher efficiency, and economic benefits by sharing validation nodes and resources. As an open-source project with an active global community of contributors, Hyperledger Fabric continuously evolves, receiving timely security updates and bug fixes through community support. Its proven reliability and trustworthiness in various industries and organizations further validate its suitability for diverse use cases [28].

The architecture of Hyperledger Fabric is depicted in Figure 2. The platform offers gRPC APIs and encapsulated SDKs for applications, enabling users to access resources in the Fabric network such as ledger, transactions, chaincode, events, and permission management. This design abstracts the internal details of the Fabric framework and allows for simple invocation through the SDKs. The ledger, the central structure for recording transaction data, is based on core blockchain elements, including database and consensus mechanism. Chaincode, the smart contract of Hyperledger Fabric, implements the execution logic of each transaction and utilizes technologies such as Docker containers and state machines. Permission management, responsible for access control throughout the process, employs security technologies like PKI systems, digital certificates, and encryption/decryption. The bottom layer of the architecture comprises multiple nodes that

form a P2P network, interact through a gRPC channel, and use the Gossip protocol for synchronization. The hierarchical structure of the architecture enhances scalability and pluggability, making it easier for developers to work on a module-by-module basis. Given these features, this study selects Hyperledger Fabric as the underlying blockchain network for the secure storage and transmission of perceptual hash.

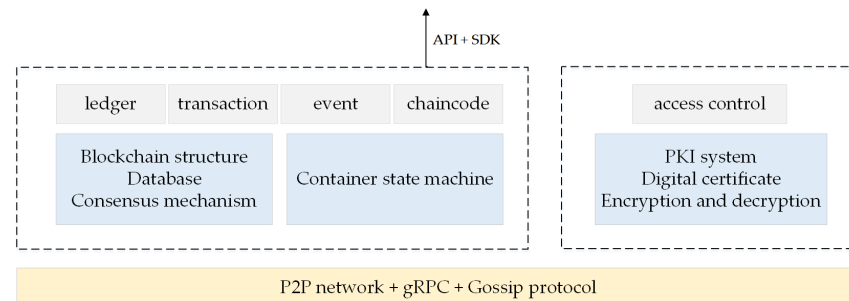


Figure 2. Hyperledger Fabric framework.

2.3.1. Feasibility of Hyperledger Fabric Applied to Perceptual Hash Secure Transmission

The feasibility of using Hyperledger Fabric for perceptual hash secure transmission in a remote-sensing image integrity authentication system is evident in the following aspects:

1. Member Licensing Mechanism for Authentication

Hyperledger Fabric utilizes an MSP (Membership Service Provider) mechanism, which manages members and verifies their identities through PKI (Public Key Infrastructure). In the remote-sensing image integrity authentication system, data organizations and users require permission to submit, transmit, and use data. The member licensing mechanism of Hyperledger Fabric sets strict protocols for network data protection, and the identity of members is disclosed upon accessing the data to identify the transmission objects.

2. Channel Mechanism for Secure Data Transmission

To ensure the security of the perceptual hash transmission process, the original perceptual hash must be protected from tampering during the transmission and acquisition process. The channel mechanism in Hyperledger Fabric provides secure point-to-point transmission and prevents uncontrolled data diffusion that may cause perceptual hash tampering [29]. In the transmission process, the peer nodes maintained by the sender and receiver join the corresponding channel and maintain the ledger data as members of the consortium blockchain. If sensitive data are involved, the regulatory authorities can also participate in the relevant channel and assume supervisory responsibilities.

3. Modular Architecture for Efficient System Construction

Hyperledger Fabric's modular architecture, with its ability to support hot plugging, makes it suitable for transforming existing systems at a relatively low cost. The mature architecture of the perceptual hash application for integrity authentication allows for a quick application of the modular Fabric network to access the perceptual hash algorithm and reduce the cost of building a perceptual hash blockchain storage system.

Hyperledger Fabric was selected as the preferred consortium blockchain platform over Corda or Quorum due to its demonstrated advantages in flexibility, security, and efficiency within the member licensing mechanism, channel mechanism, and modular architecture. In conclusion, Fabric is better suited to meet the requirements of the remote-sensing image integrity authentication system, ensuring a secure and efficient operation.

2.3.2. Necessity of Combining Hyperledger Fabric with IPFS

The storage of perceptual hash data often poses challenges for direct storage on Hyperledger Fabric due to capacity limitations. Thus, this study proposes the use of the IPFS (InterPlanetary File System) network as an auxiliary decentralized storage organization in combination with Hyperledger Fabric to ensure secure off-chain storage of large amounts of

perceptual hash values. IPFS is a secure and efficient distributed storage and transmission protocol that utilizes content addressing. It constructs a decentralized storage system with hash verification and data integrity through distributed hash addressing [30]. All nodes that run the IPFS protocol are interconnected, enabling quick location and download of data resources through directed acyclic graphs and distributed hash tables. IPFS offers benefits such as data security, tamper resistance, rapid access, no single point of failure, and firewall restriction immunity.

Compared to other decentralized storage protocols, the integration of IPFS with Hyperledger Fabric is a more effective approach. This integration has been demonstrated to be applicable to various data storage scenarios, such as land ownership, electronic medical records (EMR), and healthcare data [31–34]. However, current methods employ the public IPFS network directly as the off-chain decentralized storage center for the blockchain, which is accessible to anyone. In contrast, a private IPFS network is built on the IPFS protocol and only allows authorized users to access and share content. Compared to the public IPFS network, a private IPFS network is limited to internal use by a group or organization, and it provides higher security and privacy. Only nodes possessing the shared swarm.key can access and share content, and others cannot access information about the content. For sensitive data transmission and storage scenarios, such as remote-sensing image perceptual hash, it is necessary to use a private IPFS network to store perceptual hash data for data integrity proof. This scheme can help mitigate the risk of data tampering resulting from perceptual hash leakage and protect users' privacy and data security.

This study utilizes Kubo (go-ipfs) and the IPFS-Cluster to establish a private IPFS network and integrate it with Hyperledger Fabric to build a Perceptual Hash Secure Storage and Transmission Framework (PH-SSTF). By restricting access to nodes with the same shared key, the authenticity and security of perceptual hash value storage are ensured. At the same time, a wide range of file and information types can be stored with fewer restrictions, eliminating file redundancy and storage duplication, and improving storage network performance.

3. Methods

3.1. Remote-Sensing Image Perceptual Hash Generation Algorithm

The remote-sensing image perceptual hashing algorithm designed in this paper employs the NSCT, which is shown in Figure 3. It is composed of two stages: preprocessing and perceptual hash value generation. In the preprocessing stage, the one-layer NSCT is applied to each band of the original image. The multi-band NSCT-transformed low-frequency sub-bands are then fused according to the RE-based weighted average method, resulting in the fused low-frequency sub-band coefficients C_F . In the second stage, the statistical texture features are extracted from each grid cell of C_F , and the perceptual features are serialized to obtain the grid perceptual hash value. The perceptual hash value of the remote-sensing image is generated by concatenating and encrypting the perceptual hash values of the grids.

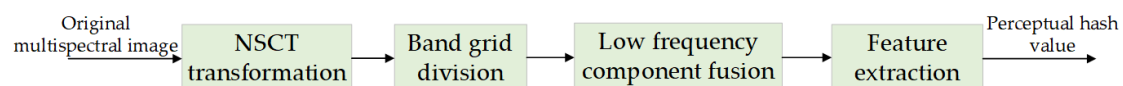


Figure 3. Remote-sensing image perceptual hash algorithm process.

3.1.1. Preprocessing Stage

The preprocessing includes the first three processes of the algorithm: NSCT transform, grid division, and multi-band NSCT low-frequency sub-band fusion. The specific steps are as follows:

1. The image k bands are each subjected to an NSCT decomposition of scale 1 to obtain k low-frequency sub-band coefficients $C^i, i \in [1, k]$.

2. The C^i is divided into $W \times H$ invisible grid cells of equal size and without overlapping. The divided grid cells are denoted as $C^i(w, h)$, where w and h identify the corresponding positions of the grid.
3. The regional energy (RE) values of all bands at the (w, h) position are calculated by Formula (1), where $E^i(w, h)$ represents the RE value of the i -th band. The RE of all bands $\{E^1, E^2, \dots, E^k\}$ will be recorded in the blockchain simultaneously with the perceptual hash for use in the preprocessing stage of tampering localization. Furthermore, the weight α_i of the sub-band coefficients of the i -th band can be calculated, and its expression is given in Formula (2).

$$\alpha_i = \frac{E^i(w, h)}{E^1(w, h) + \dots + E^i(w, h) + \dots + E^k(w, h)} \quad (2)$$

4. The weighted average of the low-frequency sub-band coefficients at each band (w, h) position, $C_F(w, h)$, is regarded as the sub-band fusion result. The specific calculation method is shown in Formula (3).

$$C_F(w, h) = \alpha_1 * C^1(w, h) + \dots + \alpha_i * C^i(w, h) + \dots + \alpha_k * C^k(w, h) \quad (3)$$

3.1.2. Perceptual Hash Value Generation Stage

The steps for perceptual feature extraction and the generation of the final perceptual hash value are as follows.

1. The fused low-frequency sub-band coefficients $C_F(w, h)$ of the grid cells are denoted as $G_{w,h}$, and each grid cell is further divided into $M \times N$ sub-blocks of equal size. Calculate the standard deviation σ_t of each sub-block, and calculate the mean μ_t and standard deviation σ_t by Formulas (4) and (5):

$$\mu_t = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N C_t(i, j) \quad (4)$$

$$\sigma_t = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (C_t(i, j) - \mu_t)^2} \quad (5)$$

In the formula, M and N are, respectively, $G_{w,h}$ the total number of rows and columns of the sub-block; $C_t(i, j)$ represents the low-frequency coefficient matrix of the t -th sub-block; σ_t represents the standard deviation of the low-frequency coefficient of the t -th sub-block with scale 1, $t \in (1, 2, \dots, M \times N)$. After the above processing, the characteristic sequence of the grid cell $G_{w,h}$ consisting of the standard deviation of the first-order low-frequency sub-bands of each sub-block NSCT transform is obtained as $H_{w,h} = (\sigma_1, \sigma_2, \dots, \sigma_{M \times N})$.

2. The mean value $\bar{\sigma}$ of the feature sequence $H_{w,h}$ is calculated and quantized according to Formula (6) to obtain the perceptual hash value $H_S^{w,h}$ of the grid cell.

$$H_S^{w,h} = \begin{cases} 1, & \sigma_t \geq \bar{\sigma} \\ 0, & \sigma_t < \bar{\sigma} \end{cases} \quad t \in (1, 2, \dots, M \times N) \quad (6)$$

3. Logistic mapping, as a typical chaotic system, can be expressed as a nonlinear iterative equation, as shown in Formula (7).

$$x_{l+1} = \mu x_l (1 - x_l), \quad x_l \in (0, 1) \quad (7)$$

The logistic mapping is chaotic when $\mu \in (3.56994564, 4]$. Chaotic sequences can be generated by logistic mapping. The obtained chaotic sequence is very sensitive to the

initial value and is non-periodic and non-convergent. The obtained chaotic sequence can be converted into binary numbers in the encryption process. Let each bit of it be an inner product with each bit of the original sequence of quantized perceptual hash, respectively, to obtain the final encrypted perceptual hash value $H_{St}^{w,h}$. In this process, the initial value in chaotic encryption is used as the key K, which is shared between the sender and receiver and can improve the algorithm’s security.

The hash value $H_{St}^{w,h}$ of $G_{w,h}$ grid cells can be generated after the above three steps, and the final perceptual hash value of the whole image is obtained by concatenating the hash values of each grid cell, $PH = (H_{St}^{1,1}, H_{St}^{1,2}, H_{St}^{1,h}, \dots, H_{St}^{w,h})$.

3.2. Perceptual Hash Secure Storage and Transmission

This paper presents a Perceptual Hash Secure Storage and Transmission Framework (PH-SSTF) based on Hyperledger Fabric and IPFS, as depicted in Figure 4. The abbreviations used below are shown in Table 2. PH-SSTF consists of three phases: initialization, request, and data transmission. The initialization phase corresponds to steps 1 to 2 in Figure 4. The request phase corresponds to steps 3 to 9 in the same figure. Lastly, the data transmission phase aligns with steps 10 to 14 in Figure 4.

3.2.1. Transmission and Storage Procedure Design

The PH-SSTF designed in this paper revolves around the interaction between subjects and the blockchain, and its flow is shown in Table 3.

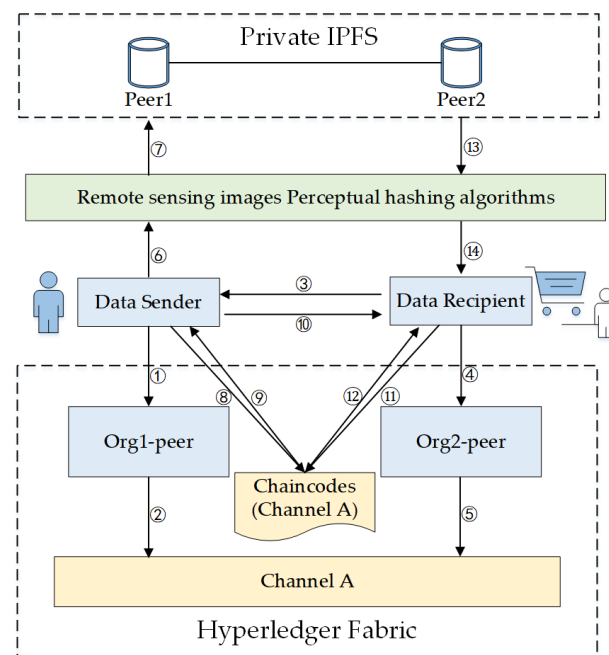


Figure 4. Perceptual Hash Secure Storage and Transmission Framework.

3.2.2. Hash Registration Chaincode Design

In this study, the perceptual hash secure storage design was implemented using the Fabric’s chaincode as a bridge between the client and the Fabric network. Figure 5 illustrates that the hashregistercc chaincode must be installed for each peer node of the representative data sender and receiver after joining Channel A to access the perceptual hash registration or query service. The ledger data of Channel A is divided into two parts: the chaincode invocation record and the current state of the local ledger. The chaincode invocation record is stored in the form of a blockchain, while the current state of the local ledger is stored in the form of a key-value database. This design provides a secure and efficient solution for

perceptual hash storage and transmission in the context of the Fabric network, ensuring data authenticity and integrity for both data senders and receivers.

Table 2. The abbreviations in framework design.

Abbreviation	Description
DS	Data Sender is an organization that has Remote-sensing images
DR	Data Recipients will perform data integrity certification
PH	Initial perceptual hash value
RE	The regional energy values of all bands
hashregistercc	perceptual hash register chaincode
TxID	Hyperledger Fabric transactions number
CID	Location of files in IPFS
swarm.key	Shared keys for private IPFS network
K	Perceptual hash secret key
RESULT	Results returned by the chaincode

Table 3. The flow of PH-SSTF.

Steps	Description
1.	DS starts the Hyperledger Fabric network and establishes a blockchain node peer of Organization 1 (Org1).
2.	Blockchain node peer of Org1 representing DS joins Channel A and deploys the chaincode in Channel A.
3.	DR sends a data transmission request to DS.
4.	DR is identified and establishes the blockchain node of Org2 in the Hyperledger Fabric network.
5.	Blockchain node peer of Org2 representing DR joins Channel A.
6.	DS will need to transmit remote-sensing image data to obtain PH through Section 3.1 Perceptual Hash Algorithm.
7.	DS uploads PH to the private IPFS network to obtain CID.
8.	Execute the chaincode and bind CID and perceptual hash key K and data transmission information to the blockchain for storage.
9.	The chaincode is executed successfully and returns RESULT and TxID.
10.	DS sends the TxID, the swarm.key file of the private IPFS network and the original file to the DR.
11.	After receiving the data and related information, the DR initiates a query by TxID.
12.	DR obtains the data corresponding to the CID of PH and the perceptual hash key K.
13.	DR joins the private IPFS network created by DS through the swarm.key file and obtains the original PH through CID.
14.	DR compares the obtained original PH with the PH obtained by the perceptual hash algorithm and performs integrity authentication.

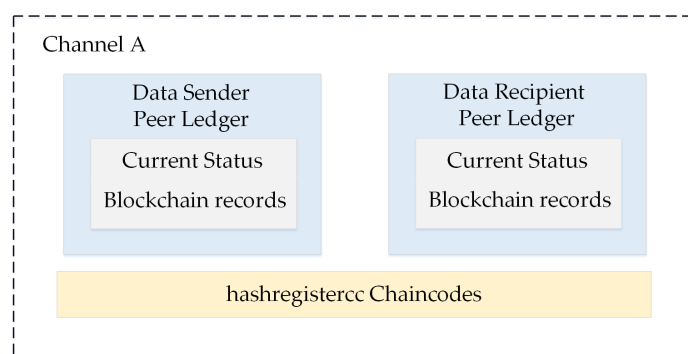


Figure 5. Installation of the chaincode in the channel.

The chaincode hashregistercc used in this system includes a transmission information registration class with the relevant attributes specified in Table 4. Sender information is obtained automatically by the system, and an image transaction timestamp is automatically generated when the hash registration chaincode is invoked.

Table 4. The registration class properties.

Properties	Types	Description
Imagedata	String	Transmitting image information
Sender	String	Sender's information
Receiver	String	Receiver's information
Imagingtime	Int64	Image production time
Transmissiontime	Int64	Image transmission time
PHaddress	String	Perceptual Hash IPFS addresses

The hashregistercc chaincode offers user hash registration and query services, with relevant functions defined in Table 5. To query the ledger information in Fabric using the chaincode, the system must implement the ChaincodeStubInterface interface under the shim package in the chaincode. Transactions are divided into two types, query and invoke, depending on the request type of the chaincode invocation method. For simple queries of ledger information, a query transaction is directly sent to query the local ledger on the peer node. However, if it involves updating the ledger, such as adding, modifying, deleting, etc., an invoked transaction is sent, waiting for endorsement from other nodes before completing the transaction. This ensures the accuracy and integrity of the data on the blockchain.

Table 5. The chaincode functions.

Method Name	Types	Input	Output	Description
init	String	N/A	Boolean	Initializes the chaincode and returns a Boolean value.
invoke	String	N/A	Boolean	Forwarding parameters to the corresponding method.
regist	String	Registration	TxID	Register the hash and return the transaction ID.
query	Int64	TxID	Registration	Query hash and transmission information.

3.2.3. Prototype System Implementation

A prototype system based on PH-SSTF has been designed to ensure the authenticity and traceability of perceptual hashes using blockchain technology. This system requires a Fabric network environment, a private IPFS network, and multiple web services and client development modules. The interface calls to the private IPFS are implemented using the Node.js web framework Express and js-ipfs, with Fabric version 1.4. The major software versions used include Aliyun ECS server for ubuntu18.04, Docker version 19.03.12, Docker-compose version 1.24.1, Go version 1.14.4, and Kubo 0.15.0. The specific construction methods for the PH-SSTF prototype system can be found in Appendix A, located at the end of this paper. The interface for the perceptual hash IPFS storage operation is presented in Figure 6a. Figure 6b shows the interface for perceptual hash registration. Figure 6c illustrates the perceptual hash query operation interface.

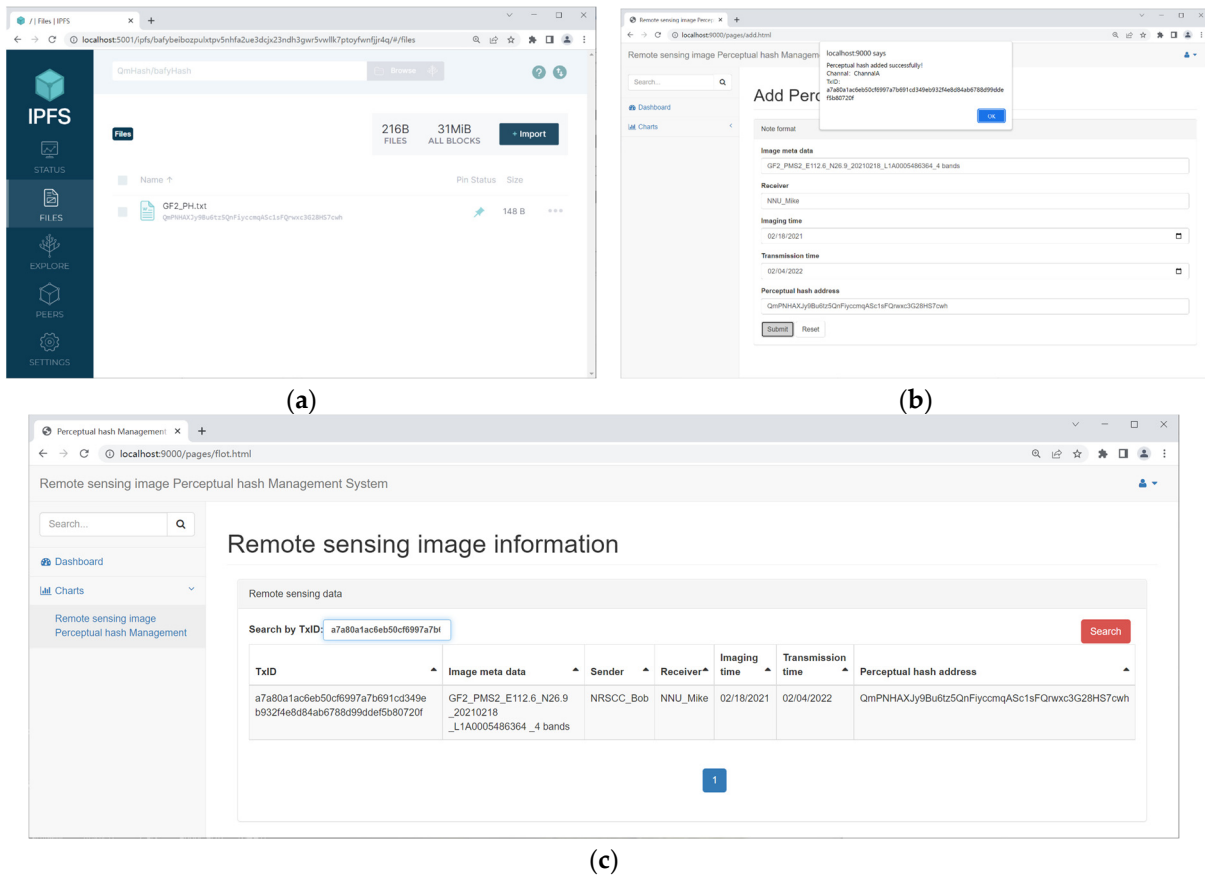


Figure 6. Prototype system interface. (a) hash storage; (b) hash registration; (c) hash query.

3.3. Remote-Sensing Image Authentication and Tampering Localization

3.3.1. Preprocessing Stage

Before integrity authentication, preprocessing is first required for possible tampering at the band level. The fusion weight of the potentially tampered bands needs to be increased by the calculation and comparison of the regional energy (RE). The specific steps are as follows:

1. After receiving the data, the data recipient (DR) calls the CID and perceptual hash key K obtained by chaincode query through TxID. DR then, using the swarm.key file, joins the private IPFS network created by the DS. By utilizing the obtained CID, DR retrieves the original perceptual hash PH (including RE) from the IPFS network.
2. The regional energy (RE) values $\{E^1, E^2, \dots, E^k\}$ for all bands of the received image are calculated by Formula (1), where E^i represents the RE value of the i -th band. Furthermore, by comparing with the regional energy values $\{E^1, E^2, \dots, E^k\}$ for each band of the original image, a new weight α_i of the sub-band coefficients of the i -th band can be assigned, and its expression is given in Formula (8).

$$\alpha_i = \begin{cases} \frac{E^i}{E^1 + \dots + E^i + \dots + E^k}, & E^i = E^i \\ \frac{E^i \times 2}{E^1 + \dots + (E^i \times 2) + \dots + E^k}, & E^i \neq E^i \end{cases} \quad (8)$$

3. The low-frequency sub-band coefficients C^i of each band are multiplied by the new weights α_i to obtain the sub-band fusion result C^i_F . The specific calculation method is shown in Formula (9).

$$C^i_F = \alpha_{1.} * C^1 + \dots + \alpha_{i.} * C^i + \dots + \alpha_{k.} * C^k \quad (9)$$

Any form of tampering will induce alterations in the regional energy of a specific band. Consequently, any band exhibiting inconsistencies compared to the original image is classified as a potentially tampered band. Thus, the multiplication of the original weights in Formula (8) by a factor of 2 is intended to amplify the significance of potentially tampered bands, therefore enhancing the visibility of tampering effects in the band fusion results. Although some inconsistencies may be due to content-preserving operations, this preprocessing does not affect the actual result of integrity authentication.

3.3.2. Integrity Authentication Stage

The main process of the integrity authentication phase can be divided into the following steps.

1. Original PH acquisition phase: DR decrypts the original hash value PH obtained by the IPFS network through perceptual hash key K Get PH_{S1} , $PH_{S1} = (H_{S1}^{1,1}, H_{S1}^{1,2}, H_{S1}^{1,h}, \dots, H_{S1}^{w,h})$.
2. Authentication phase: extract the quantized perceptual hash value PH_{S2} , $PH_{S2} = (H_{S2}^{1,1}, H_{S2}^{1,2}, H_{S2}^{1,h}, \dots, H_{S2}^{w,h})$ from the sub-band fusion result C'_F of the pending authentication image according to the hash value generation step in Section 3.1.2. The distance $D_H^{w,h}$ between the perceptual hash value $H_{S1}^{w,h}$ and $H_{S2}^{w,h}$ corresponding to the same location grid cell $G_{w,h}$ is calculated using the normalized Hamming distance. The specific formula is as follows.

$$D_H^{w,h}(H_{S1}^{w,h}, H_{S2}^{w,h}) = \frac{1}{L} \sum_{\omega} \{ |H_{S1}^{w,h}(\omega) - H_{S2}^{w,h}(\omega)| \} \quad (10)$$

In the formula: $H_{S1}^{w,h}, H_{S2}^{w,h}$ are two hash values of length L ; ω denotes each bit in the hash value.

3. The calculated normalized Hamming distance $D_H^{w,h}$ is compared to a predefined threshold T . If $D_H^{w,h}$ is found to be greater than T , it is concluded that the corresponding grid cell $G_{w,h}$ of the received image has been tampered with and marked as such. This process is repeated for all the grids. If the mean distance $D_H^{w,h}$ of all grid cells of the received image is less than the threshold T , it can be concluded that the received image is either the original image or a trusted image that has undergone content-preserving operations. Regarding the setting of the threshold T , a dedicated experiment will be carried out in Section 4.1.1 of this paper, as it concerns the balance between the robustness and sensitivity of the algorithm.
4. If tampering with a grid is detected, it is considered that the integrity of the transmitted image has been compromised. Additionally, the location of the tampered area can be determined using the grid that was marked during the authentication phase.

4. Experiments and Results

4.1. Perceptual Hash Algorithm Validity Experiment

In this study, a total of 400 original remote-sensing images were used for experiments to form the "original dataset", including 200 Google Earth remote-sensing images from the DOTA high-resolution image database [35], 150 Gaofen-2 (GF2) satellite images from the GID high-resolution image dataset [36], and 50 Sentinel-2B L2A-class remote-sensing image products downloaded from the official Sentinel data website. The image size ranges from 1444×1727 to $10,000 \times 10,000$, with resolutions of 0.5 m for Google Earth images, 1 m for GF2 images, and 10 m, 20 m, and 60 m for Sentinel-2B images. The band information of the experimental data is summarized in Table 6.

Table 6. Introduction of experimental data spectrum.

Parameters	Google Earth	Gaofen-2 (GF2)	Sentinel-2B		
Spectral Number	3	4	13		
Spectral Central wavelength (nm)	B-Blue 490	B01-485	B01-443	B05-705	B09-940
	B-Green 560	B02-555	B02-490	B06-740	B10-1375
	B-Red 665	B03-655	B03-560	B07-783	B11-1610
		B04-830	B04-665	B08-842	B12-2190
			B08A-865		

The hardware platform for the experiment is a quad-core CPU with a 3.8 GHz main frequency and 16 GB memory; the software development platform is MATLAB R2022a, and some functions are implemented based on the NSCT toolbox. The parameter $\mu = 3.6$ for chaotic mapping, and the initial value of encryption key K , i.e., chaotic sequence, is 0.5.

4.1.1. Threshold Determination Experiment

Perceptual hashing algorithms are evaluated based on two performance metrics: robustness and sensitivity. Robustness refers to the ability to produce similar hash values for visually similar images, even after content-preserving operations. Sensitivity refers to the ability to detect tampering operations. These two metrics are often inversely related, meaning that improving one can lead to a degradation in the other. Therefore, a reasonable threshold must be chosen to balance the two and to be able to identify content-preserving operations and content-tampering operations effectively.

Accurately determining the threshold value for testing requires a large amount of data. To achieve this, seven types of content-preserving operations were applied to 400 images from the “original dataset”, resulting in a “similar dataset” comprising 2800 images that retained the visual content of the original images. The operations included including LSB watermark embedding (1 bit-planes), JPEG compression (Quality Factor = 50, 90), Gaussian filtering (standard deviation = 0.5, 5), format conversion to BMP and PNG (Gaofen-2: B03, B02, B01; Sentinel-2B: B04, B03, B02). The original dataset was then divided into 12,648 grid cells, while the similar dataset contained 88,536 grid cells. This resulted in 88,536 pairs of gridded cells with similar content for testing purposes.

To create the “tampered dataset”, a tampering process was applied to 12,648 grid cells from the original dataset. Specifically, an external image block was pasted into a block of each grid cell, with the paste area ranging from 15% to 25% of the original grid cell. Paste blocks were available in three different sizes: 32×32 , 64×64 , and 128×128 .

Formula (10) was utilized to compute the normalized Hamming distance between 88,536 pairs of similar data and 12,648 pairs of tampered data. The “Detection rate”, which represents the probability of correct detection, was defined as presented in Formula (11). The detection rate results under various thresholds T are illustrated in Figure 7. The X-axis corresponds to the authentication threshold T , while the red and blue curves represent the detection rates of similar images and tampered images, respectively. The results indicate that the detection rates of similar and tampered images intersect approximately when $T = 0.05$. This means that the algorithm strikes a balance between robustness and sensitivity when $T = 0.05$ is chosen as the threshold to ensure that the algorithm can perform its task effectively in experiments, both in terms of maintaining robustness to similar images and detecting tampered images. Therefore, in the experiments, a threshold T of 0.05 was adopted to differentiate between the similar and tampered images.

$$\text{Detection rate} = \frac{\text{The number of correct detection}}{\text{The total number of tested images}} \quad (11)$$

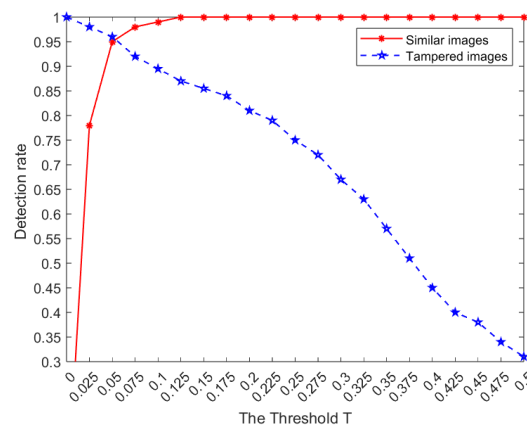


Figure 7. Performance of image authentication with varying threshold T.

4.1.2. Algorithm Robustness and Sensitivity Experiment

The purpose of this section is to assess the robustness of the proposed algorithm to a broader range of unexpected changes caused by content-preserving operations, as well as its sensitivity to detecting tampered data. To investigate the robustness of the algorithm, additional content-preserving operations were applied to the 12,648 grid cells derived from the “original dataset”, as shown in Table 7, using 15 types of content-preserving processing operations to produce a “new similar dataset” of a total of 189,720 grid cells. To evaluate the algorithm’s ability to identify tampered data, an equal number of tampered grid cells were generated for each similar image using the same generation method described in the previous section for the “tampered dataset”.

Table 7. Parameter setting for Content-preserving operations.

Content-Preserving Operation	Variable Parameters	Parameters Setting
JPEG compression	Quality factor	50, 90
Gaussian filtering (4×4)	Standard deviation	0.5, 5
Motion blurring	Motion length	5, 10
Salt and pepper noise	Noise density	30%, 50%
Gaussian noise	Standard deviation	5, 50
Unsharp masking	Gain factor	1, 10
LSB watermark embedding	Bit-planes number	1
Format conversion to BMP	Selected bands	Gaofen-2: B03, B02, B01; Sentinel-2B: B04, B03, B02
Format conversion to PNG	Selected bands	Gaofen-2: B03, B02, B01; Sentinel-2B: B04, B03, B02

The Hamming distances between the perceptual hash of the original image and its corresponding similar image, as well as the tampered image, are calculated. The maximum, minimum, and mean values of the Hamming distances for each operation type in the dataset are shown in Table 8.

Based on the results presented in Table 8, a threshold value of 0.05 proves effective in distinguishing similar images from tampered ones, though a few recognition failures were also observed. To demonstrate the superiority of this algorithm, the evaluation method of the Receiver Operating Characteristic (ROC) curve is primarily adopted for assessing the performance of image perceptual hashing algorithms [37]. The ROC curve plots the False Positive Rate (FPR) on the horizontal axis and the True Positive Rate (TPR) on the vertical axis, with a set of FPR values and TPR values obtained by varying the threshold T. TPR represents the ratio of correctly classified similar images to all similar images. In contrast,

FPR represents the ratio of tampered images misclassified as similar to all tampered images. Formula (12) provides the formal definitions of TPR and FPR.

$$\text{TPR} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \text{FPR} = \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} \quad (12)$$

In this formula, *True Positive* refers to the number of similar samples that are correctly identified as similar. *False Positive* refers to the number of tampered samples that are incorrectly identified as similar. *False Negative* refers to the number of similar samples that are incorrectly identified as tampered. *True Negative* refers to the number of tampered samples that are correctly identified as tampered.

Table 8. Hamming distance under different types of operations.

Processing Operations	Similar			Tampered		
	Max.	Min.	Mean	Max.	Min.	Mean
JPEG compression	0.0495	0.0005	0.0045	0.5547	0.0623	0.3243
Gaussian filtering (4 × 4)	0.0237	0.0014	0.0027	0.5124	0.1134	0.2894
Motion blurring	0.0517	0.0018	0.0124	0.5435	0.1044	0.3827
Salt and pepper noise	0.0356	0.0015	0.0138	0.4989	0.0384	0.2845
Gaussian noise	0.0371	0.0027	0.0110	0.4872	0.0738	0.3143
Unsharp masking	0.0428	0.0016	0.0107	0.6483	0.1845	0.2976
LSB watermark embedding	0.0469	0.0020	0.0265	0.5743	0.0937	0.3285
Format conversion to BMP	0.0124	0.0000	0.0026	0.6463	0.1173	0.4866
Format conversion to PNG	0.0112	0.0000	0.0012	0.6271	0.1321	0.4824

In the context of ROC curve analysis, a larger True Positive Rate (TPR) and a smaller False Positive Rate (FPR) indicate better algorithm performance. Hence, the proximity of the algorithm's curve to the upper-left corner of the ROC graph signifies superior algorithm performance. At times, visual inspection may not suffice to discern which ROC curve corresponds to a better-performing hash algorithm, leading to the calculation of the Area Under Curve (AUC). The AUC is a value between 0 and 1, representing the area enclosed by the ROC curve and the axis. A higher AUC value suggests improved performance of the classification model in accurately distinguishing between positive and negative instances, with a perfect model yielding an AUC of 1, and a random guessing model resulting in an AUC of 0.5. Thus, the application of the ROC curve can effectively indicate the relative superiority or inferiority of different algorithms.

The algorithms in this study are compared with state-of-the-art algorithms recently published in academic journals. These comparison algorithms utilize FAST features [11] and edge features [9,12,38] for the same dataset constructed in this section. The results of the ROC curves plotted by all the algorithms are presented in Figure 8.

By comparison, it can be seen clearly that the AUC value of the proposed algorithm is a bit higher than that of the comparison algorithms. These findings suggest that the algorithm possesses a stronger ability to distinguish between content-preserving and content-tampering operations. Overall, the results demonstrate the algorithm's reliability and accuracy in ensuring data integrity.

4.2. Tampering Localization Experiments

4.2.1. Tampering Localization Ability Test

The proposed algorithm's ability to locate tampering was evaluated on a three-band Google Earth image of size 1024 × 1024, as shown in Figure 9. The algorithm accounts for the vast data volume of remote-sensing images and extracts features for each grid. The tampering accuracy of the algorithm depends on the granularity of the invisible mesh division of the grid. To balance high tampering accuracy and reasonable computation time, a grid cell size of 64 × 64 pixels was used, and the original image was divided

into 16×16 grid cells. Further details on the impact of grid cell size are discussed in Section 5. The original image was subjected to an Erasure attack on the multi-band, a Copy-move attack on the green-band, and a Modify attack on the blue-band. The results of the tampering localization are presented in Table 9.

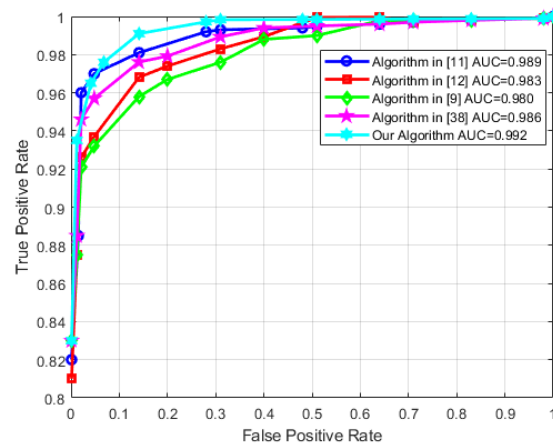


Figure 8. Comparison of ROC among different algorithms.



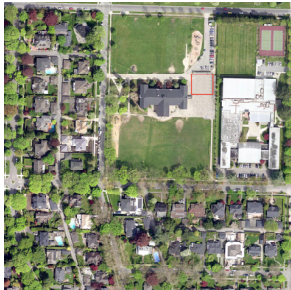




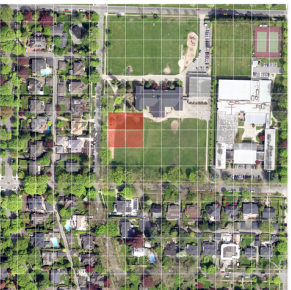



Figure 9. Original image. (a) Multi-band; (b) Blue-band; (c) Green-band; (d) Red-band.

Table 9 displays the average Hamming distances between the tampered and non-tampered regions and the original image. The average Hamming distance of the tampered region exceeds the algorithm's set threshold $T = 0.05$, while the average Hamming distance of the untampered region is below the threshold. The tampering localization results in the table show that the algorithm identified 15 tampered cells with a 100% tampering recognition rate. Notably, the algorithm can identify Copy-move and Modify attacks at the band level, in addition to detecting Erasure attacks across multiple bands.

4.2.2. Comparative Experiments of Different Algorithms for Tampering Detection

This section focuses on evaluating the efficacy of the proposed algorithm for remote-sensing image tampering identification of multiple landscape types. To this end, grid images representing six different landform types in the GID, as illustrated in Figure 10a–f, were selected as the tampering detection experimental objects. The perceptual hashing algorithms that utilized FAST features [11] and edge features [9,12,38] for tamper localization were employed for comparison. Three tampering attack methods described in Section 4.2.1 were used to evaluate the algorithms' effectiveness. The obtained results are presented in Table 10 where the tamper recognition rate is expressed as the percentage of correctly identified tampered grids out of the total number of tampered grids.

Table 9. Tampering with location test results.

Tampering Type	Band Tampering Case	Band Fusion Results	Grid Localization	Average Hamming Distance
Erasure				Untampered area: 0.0032 Tampered area: 0.5456
Copy-move				Untampered area: 0.0034 Tampered area: 0.3856
Modify				Untampered area: 0.0012 Tampered area: 0.4712

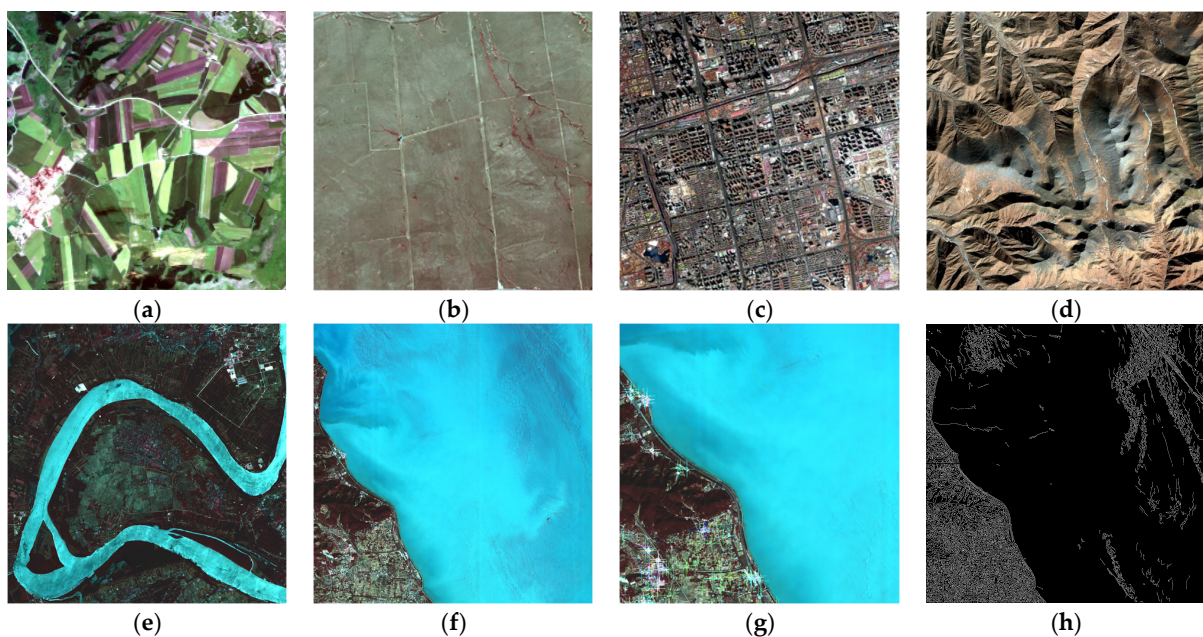


Figure 10. Test images and Feature extraction images. (a) Farmland; (b) Desert; (c) Towns; (d) Mountains; (e) River; (f) Marine; (g) FAST features (partial); (h) Edge features.

Table 10. Comparison of different algorithms for tamper detection.

Algorithm	Farmland	Desert	Towns	Mountains	River	Marine
[9]	96.67%	82.87%	96.70%	96.72%	96.19%	87.02%
[11]	97.22%	88.39%	98.35%	95.62%	95.10%	85.94%
[12]	97.78%	90.61%	96.15%	96.17%	95.65%	86.48%
[38]	98.89%	93.92%	97.80%	97.26%	96.73%	91.89%
This paper	98.89%	94.47%	97.80%	97.81%	97.28%	94.59%

As can be seen from Table 10, the algorithm in this paper has a high tamper recognition capability in recognizing remote-sensing images of various landforms. In particular, it also has a stable tamper recognition capability for images of smooth areas such as deserts and marine where edge or corner point features are less noticeable. Figure 10 shows the FAST feature points and the ‘canny’ edge features for images of marine areas, respectively. Figure 10g shows that the corner point features (cross marks) are predominantly present in coastal areas compared to gently sloping oceanic waters. Figure 10h also shows that edge features are predominantly present in complex coastal areas. Therefore, the comparison algorithm is poor at identifying Copy-move and Modify attack tampering for the interior of gently sloping regions with sparse features such as marine and desert. The algorithm proposed in this paper is based on the statistical features of image texture, which has more stability and broader applicability than the perceptual hashing algorithm based on FAST features and edge features.

4.3. Perceptual Hash Storage and Query Performance Experiments

4.3.1. Perceptual Hash Storage Efficiency Experiment

In the PH-SSTF, the perceptual hash file is packaged into data blocks and stored in a private IPFS network, with the hash addresses stored on the Hyperledger Fabric. In contrast, existing methods directly store data on the public IPFS network. This experiment focuses on comparing the storage performance between the traditional public IPFS network and the PH-SSTF using a private IPFS network for files of varying sizes. Five groups of perceptual hash data with sizes of 1 MB, 5 MB, 10 MB, 20 MB, and 50 MB were selected, resulting in a total of 50 test cases per group.

The results, as shown in Figure 11, indicate that the average latency in the private IPFS network constructed by PH-SSTF is 0.14 s, 0.31 s, 0.98 s, 11.55 s, and 26.15 s, respectively, for the different file sizes. Conversely, the average latency in the public IPFS network is 0.26 s, 0.83 s, 1.65 s, 11.71 s, and 27.65 s, respectively. In terms of storage speed, the average storage rate in the private IPFS network is 9.09, 19.23, 20.52, 24.73, and 29.94 MB/s, respectively, for the different file sizes. In comparison, the average storage rate in the public IPFS network is 8.35, 17.69, 11.15, 5.88, and 4.73 MB/s, respectively. The experimental results demonstrate that the storage rates in the private IPFS network are consistently higher than those in the public IPFS network. For data sizes up to 5 MB, there is minimal discrepancy in speed between the private and public IPFS networks. However, as the file size increases (≥ 10 MB), the disparity in speed becomes more pronounced. Notably, the private IPFS network achieves a speed of more than five times faster than the public IPFS network for a 50 MB file. In the public IPFS network, data transmission primarily occurs through a peer-to-peer mechanism, with data distributed and stored across multiple nodes. This distributed characteristic implies that the transmission speed of data can be limited by the slowest or bottleneck nodes within the network, especially when handling larger files. Additionally, as file size increases, the need to transfer a greater number of data blocks can lead to longer response times by nodes, therefore slowing down the overall data transmission speed. Furthermore, factors such as network congestion, node loads, and network traffic also influence data transmission speed in the public IPFS network. During periods of heightened network traffic or congestion, the competition for limited network resources by numerous data streams can further contribute to a decline in transmission speed. Therefore, the observed decrease in speed with increasing file size in the public IPFS

network is the result of the combined effects of multiple factors. Hence, the data storage performance of the private IPFS network generally outperforms that of the public IPFS network.

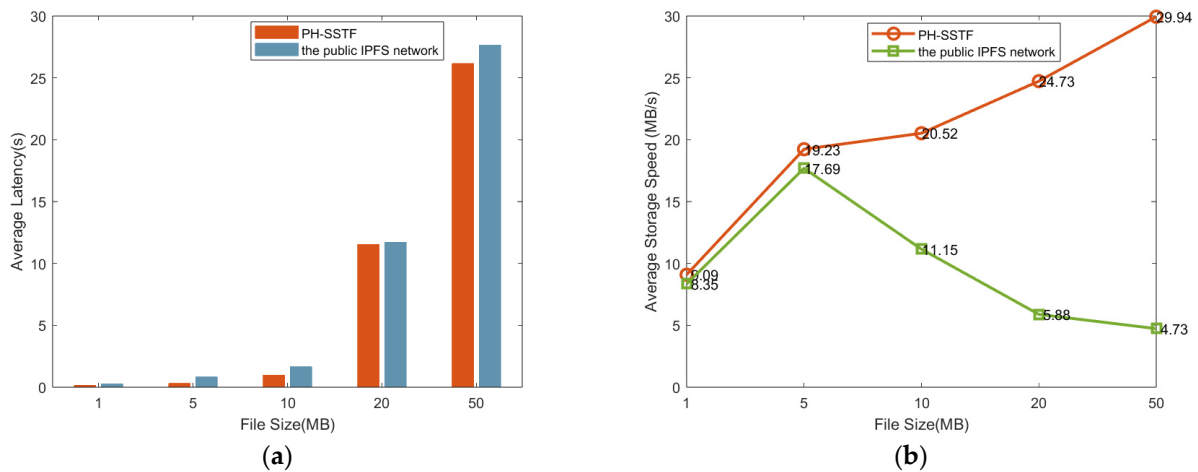


Figure 11. Performance comparison of perceptual hash storage efficiency. (a) Average latency; (b) Average Storage Speed.

4.3.2. Perceptual Hash Query Efficiency Experiment

The perceptual hash query process does not involve IPFS network interaction. Thus, only Fabric network data query experiments were conducted. Hyperledger Fabric provides two methods for querying data on the blockchain: chaincode queries and CouchDB queries. Chaincode queries involve invoking the query function of the chaincode to retrieve data stored on the blockchain, ensuring high availability and persistence. The query results are based on the latest blockchain state. On the other hand, CouchDB serves as a separate state database where data are stored independently, enabling individual backup and recovery. By utilizing CouchDB queries, data can be directly retrieved from the CouchDB database without depending on the blockchain. The query performance of these two methods is compared in Figure 12, using the results of 50 queries as an example. The x-axis labeled ‘i’ represents the index of each query, while the y-axis labeled ‘t’ represents the corresponding response time. The response time is subject to dynamic variations depending on the network speed at the time of the query execution. In this study, the chaincode query approach demonstrates shorter response times compared to the CouchDB query method, meeting the requirements for network latency.

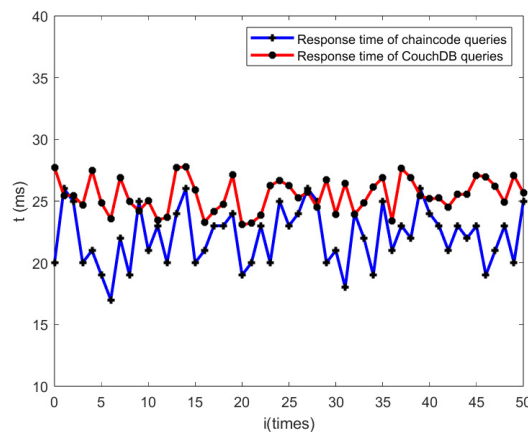


Figure 12. Performance comparison of perceptual hash query efficiency.

Chaincode queries are executed within the chaincode runtime environment and do not necessitate additional network communication. As a result, chaincode queries are

generally faster than CouchDB queries, particularly for simple key-value queries. In contrast, CouchDB queries require network access to the CouchDB database, and their performance can be affected by network latency and throughput. If there are higher demands for the performance and availability of blockchain data, chaincode queries may be more appropriate.

5. Discussion

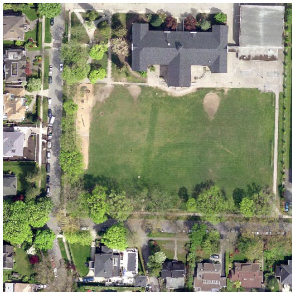
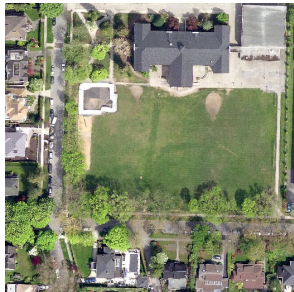








5.1. Tamper Sensitivity Analysis

5.1.1. Sensitivity Analysis for Grid Cell Size

The size $w \times h$ of the grid division can be considered to be the protection level of the algorithm. The size of the grid cells will directly affect the granularity and accuracy of tampering identification. Since the experimental image in Figure 9 is 1024×1024 size for the square, for the convenience of computer processing, this experiment takes 512×512 , 256×256 , 128×128 , 64×64 , and 32×32 the five grid cells for grid cell size sensitivity analysis.

The results presented in Table 11 indicate that the calculation time decreases as the grid size decreases. Upon multiplying the single-grid computation time with the total number of grids, the total time consumption for integrity authentication was estimated to be 121.16, 135.2, 170.24, 286.72, and 573.44 s for grid sizes of 512×512 , 256×256 , 128×128 , 64×64 , and 32×32 , respectively.

Table 11. Comparison of tampering sensitivity for different grid sizes.

Grid Size	Sample of Original Images	Sample Tampered Images	Single-Grid Calculation Time (s)	Hamming Distance between Similar Grids
512×512			30.29	0.0411
256×256			8.45	0.1491
128×128			2.66	0.3492
64×64			1.12	0.4349
32×32			0.56	0.3667

In the absence of improvement in tamper identification, the time consumed for selecting a 32×32 size grid is almost twice as long as the 64×64 size. When the grid size is too small, there is an overall problem of excessive time consumption. When the integrity authentication threshold T is set to 0.05, the tampered grid is divided into a 512×512 size grid with missed judgments, so the grid size should not be too large. The tampering recognition is stronger when the grid size is less than 256×256 .

Based on the analysis of computation time and tampering localization accuracy, a 128×128 grid size was selected for authentication, which represents a balance between



















algorithmic complexity and recognition capability. If a higher level of tampering localization accuracy is required, a grid size of 64×64 is recommended.

5.1.2. Sensitivity Analysis for Different Levels of Tampering

The preceding section's results indicate that improved tamper localization capability can be achieved within a reasonable computational elapsed time for a grid size of 64×64 . However, further experiments and discussions are required to determine the tamper recognition sensitivity within the grid. In this section, we adopt a 64×64 grid as the basic unit and conduct various degrees of tampering, ranging from 10% to 90% of the grid, to determine the minimum degree of tampering detectable by the algorithm.

When using a grid of 64×64 as the basic unit and comparing it with the threshold $T = 0.05$ set in this paper, the results presented in Table 12 reveal that there are instances of tampering that cannot be recognized when the degree of tampering is less than or equal to 10%. At this level, the tampered area is approximately 20×20 in size, and the threat posed by such isolated tampering is limited for conventional remote-sensing images. However, if computational conditions permit, finer grid division, such as 32×32 , yields accurate recognition of this degree of tampering, as demonstrated in the last row of Table 11.





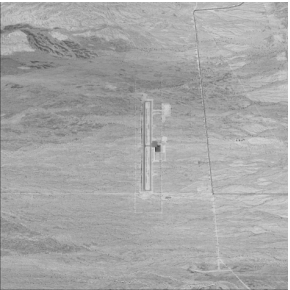
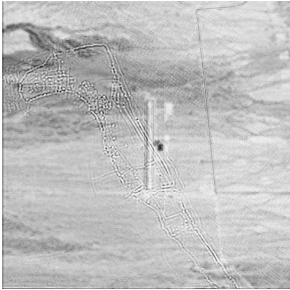
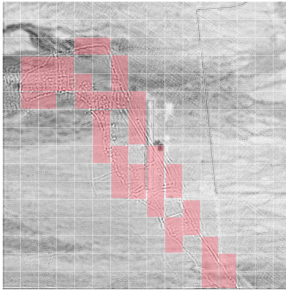

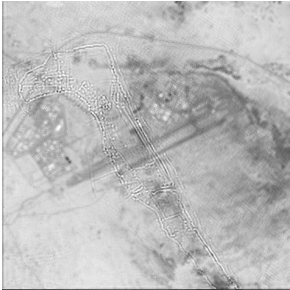
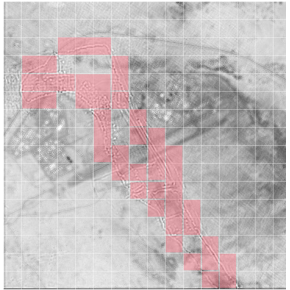
Table 12. Comparison of tampering sensitivity for different levels of tampering.

Level of Tampering	Sample of Original Images	Sample of Tampered Images	Hamming Distance
10%			0.0364
20%			0.0567
30%			0.1150
40%			0.1664
50%			0.2445
60%			0.3457
70%			0.3862
80%			0.4547
90%			0.5349

5.1.3. Sensitivity Analysis of High-Frequency Sub-Band Tampering

The algorithm proposed in this study is based on fusing the low-frequency sub-bands obtained through the application of NSCT to the original image. To assess the tampering effect of solely manipulating the high-frequency sub-bands, experimental validation becomes imperative. Tampering methods for high-frequency sub-bands can be categorized into two main types. The first category involves local tampering, such as erasing or modifying the high-frequency sub-bands themselves. The second type is global tampering, which entails directly replacing the original high-frequency sub-bands with high-frequency sub-bands from other images. Table 13 presents a comparison of the tampering effects of these two methods, where the high-frequency sub-bands of the third and fourth images are replaced with the high-frequency sub-bands of the second image.

Table 13. Comparison of tampering sensitivity of high-frequency sub-band tampering.

Tampering Methods	Sample of Original Images	Sample Tampered Images	Grid Localization	Average Hamming Distance
Erasure			none	0.0237
Modify			none	0.0145
Swap				0.3678
Swap				0.4147

The results presented in Table 13 demonstrate that local tampering with the high-frequency sub-bands alone leads to a reconstructed image that is perceptually similar to the original image. The Hamming distances between the reconstructed image and the original image perceptual hash are all below the threshold T . In the context of this paper, such tampering can be considered a content-preserving operation, rendering it invalid for practical applications. On the other hand, global tampering, like the direct replacement of high-frequency sub-bands, causes significant changes in the texture of the reconstructed image, making it valid tampering. The algorithm in this paper successfully identifies valid tampering, as evident from the table results. Due to the inherent randomness of tampering in the frequency domain, coupled with the diverse texture characteristics of the original image base map, inconsistencies arise in the tampering localization results.

5.2. Original Perceptual Hash File Storage Reliability Analysis

The original perceptual hash files are stored in a distributed manner using the private IPFS network to address the issue of a single point of failure, which can lead to data loss and inaccessibility due to node failures. IPFS breaks down these files into fragments, with each node storing fragments rather than complete data, based on unique file addresses generated using content-based hashing. Even if a node is compromised, data accuracy and integrity are not compromised.

However, the potential for a node to engage in a conspiracy attack and tamper with the stored perceptual hash files exists. The probability of such an attack is denoted as P_{fault} and can be expressed using Formula (13) [39].

$$P_{fault} = \sum_{i=0}^{k-1} P^k (1-P)^{n-k} C_n^k \quad (13)$$

In the formula, n is the total number of nodes, k is the number of malicious nodes present, and P is the probability of each node being online.

According to Formula (11), the probability of a successful attack by a malicious node in a peer-to-peer network can be calculated based on the percentage of such nodes in the network. The probability values are presented in Table 14, where ' f ' represents the percentage of malicious nodes out of the total number of nodes in the network.

Table 14. Probability of malicious nodes successfully tampering with data.

$f\%$	P_{fault}
5	9.53×10^{-7}
10	2.00×10^{-5}
15	2.01×10^{-4}
20	1.29×10^{-3}
25	5.90×10^{-3}
30	2.06×10^{-2}

In Table 14, the following base conditions are set: the total number of nodes n is 20, the probability of each node being online is 0.5, and the percentage of k to the total number of nodes is less than 1/3.

Table 14 shows that an increase in the percentage of malicious nodes in the network raises the probability of a successful attack. For instance, when 30% of nodes are malicious, the probability of a successful attack is only approximately 2%, indicating a relatively low success rate. Moreover, this method employs a perceptual hashing algorithm that uses logical transitions to encrypt the hash values, ensuring the original hash values are securely protected.

5.3. Comparison with Existing Method

5.3.1. Comparison of Integrity Authentication Algorithms

This section presents a comparison between the proposed image integrity authentication algorithm and several existing algorithms that address the same issue. These include digital signature algorithms [40], semi-fragile watermarking algorithms [5,8], and perceptual hashing algorithms [11,38]. The comparison and discussion are conducted in four dimensions: non-destructive to the original data, robustness to content-preserving operations, consideration of multi-band properties, and tamper localization ability. Table 15 presents the comparison results.

The results in the table show that the digital signature-based integrity authentication method lacks robustness and tamper locating capability. Although the semi-fragile watermarking method improves on this, it is somewhat destructive to the original data. However, the method described in this paper refrains from making any alterations to the

original image; as a result, it does not have any bearing on the accuracy of subsequent applications, such as ground object extraction, in practical remote-sensing research [41]. Compared with existing perceptual hashing algorithms, the proposed algorithm takes into account the multi-band characteristics of remote-sensing images and thus can identify band-level tampering attacks. Moreover, the overall efficiency of the algorithm is higher than the perceptual hashing algorithm that requires multiple repetitions of single-band processing because the band fusion process has been performed.

Table 15. Comparison of different algorithms for integrity authentication.

Algorithm	Digital Signature [40]	Watermarking [5]	Watermarking [8]	FAST Based [11]	Canny Based [38]	Proposed Algorithm
Non-destructive to the original data	No	No	No	Yes	Yes	Yes
Robust to Content-preserving operations	No	Yes	Yes	Yes	Yes	Yes
Consider multi-band characteristics	No	No	Yes	No	No	Yes
Tamper Localization	No	Yes	Yes	Yes	Yes	Yes

5.3.2. Comparison of Waveband Fusion Methods

In this section, the fusion method proposed in this paper will be objectively compared with existing band fusion methods. Three comparative methods are employed, encompassing classical Principal Component Analysis (PCA), HSV conversion, and Wavelet Transform. To meet the requirements of high efficiency and sensitivity to tampering, it is necessary to quantitatively calculate the algorithm execution time and the peak signal-to-noise ratio (PSNR) value between the fused result and the original tampered band. PSNR is defined as shown in Formula (14).

$$PSNR = 10 \times \log_{10} \left[\frac{255^2}{\frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [X_{i,j} - X'_{i,j}]^2} \right] \quad (14)$$

In the formula, $X_{i,j}$ and $X'_{i,j}$ represent the pixel values at position (i, j) of the fused image and the original tampered image, respectively. M and N represent the number of rows and columns of the images. PSNR can be used to measure the performance of the fusion result in detecting tampering. A higher PSNR value indicates a higher sensitivity of the fusion to tampering. The image data from Section 4.2.1 was selected as the unified experimental data, and the specific experimental results are shown in Table 16.

From the results shown in the table, it can be seen that the fusion method proposed in this paper has the shortest execution time. In terms of tampering at the band level, the fusion method proposed in this paper also has the highest PSNR value. The method proposed in this paper includes a preprocessing step of checking the energy variation before detection, which increases the fusion weight of potential tampered bands. Therefore, the fusion method exhibits higher sensitivity to tampering and is more suitable for integrity authentication scenarios.

5.3.3. Comparison of Previous Schemes and the Proposed Scheme

The security of traditional perceptual hashing algorithms relies on the chaotic transformation or encryption methods to convert image information into a meaningless encrypted string, which can be securely transmitted using a secret key. Implementation of these methods may use the logistic transform employed in this study's algorithm or encryption algorithms such as RC4 or AES to encrypt the hash value [9,15]. However, traditional methods do not guarantee the authenticity of the original perceptual hash. Therefore, the

assistance of blockchain technology is needed to achieve further refinement. Currently, the research area of remote-sensing image integrity authentication by combining perceptual hashing with blockchain has only been initially explored by Ding et al. [42]. Nevertheless, that study focused on improving the tamper sensitivity of the perceptual hashing algorithm itself. Only a blockchain structure was proposed for perceptual hash storage, with no innovative design or practical deployment for specific storage and transmission scenarios. Therefore, in this section, the proposed PH-SSTF framework is compared horizontally with the Hyperledger Fabric and IPFS integration schemes used for other Objectives. Table 17 presents the comparison results, which can be analyzed to identify the contributions and significance of the proposed approach in this paper.

Table 16. Comparison of waveband fusion methods.









Methods	Tampering Sample	Fusion Results	Fusion Time (s)	PSNR (dB)
PCA			0.53452	11.35
HSV conversion			0.83972	33.20
Wavelet Transform			0.92643	29.85
Ours			0.32347	51.62

Table 17. Functionality comparison of previous schemes and the proposed scheme.

Authors	Objective	1	2	3	4
Mukne et al. [31]	Land acquisition and ownership record management	Yes	Yes	No	No
Nyalety et al. [32]	Proposed a BlockIPFS to create a clear audit trail	Yes	Yes	No	No
Li et al. [33]	Safe storage and sharing of medical record data	Yes	Yes	No	No
Mani et al. [34]	Patient-centric healthcare data management	Yes	Yes	No	No
PH-SSTF	Secure transmission and storage of perceptual hash	Better	Better	Yes	Yes

Notes: 1: Data security and privacy protection, 2: High-speed data storage and retrieval, 3: Elastic storage and capacity expansion, 4: Offline data storage and offline transaction.

According to the comparison results in the table, it can be concluded that the PH-SSTF proposed in this paper has the following optimization points.

1. Stronger Data Security and Privacy Protection

Although existing combination schemes consider both data security and privacy protection, a private IPFS network enhances these aspects by restricting access to a specific group or organization. Only nodes with the shared swarm.key can access and share content, preventing unauthorized access and data leakage.

2. Higher Speed Data Storage and Retrieval

Utilizing IPFS as the underlying storage layer for Hyperledger Fabric enables distributed data storage and retrieval. The combination of a private IPFS network and Hyperledger Fabric enhances data storage and retrieval performance. Private IPFS networks provide a direct and efficient way to access specific data, reducing latency and increasing retrieval speed.

3. Support Elastic Storage and Capacity Expansion

Private IPFS networks offer elastic storage capabilities, allowing flexible allocation and expansion of storage capacity based on data requirements. Additional storage nodes can be added, or existing nodes' capacity can be increased to handle expanding data scales. Hyperledger Fabric's flexible architecture seamlessly integrates expanded storage capacity within the blockchain network.

4. Support Offline Data Storage and Offline Transactions

Private IPFS networks enable local data storage on nodes, even without network connectivity. Integration with Hyperledger Fabric allows for offline transaction support, where transactions can be created and signed offline and then propagated once connectivity is restored. This enhances system robustness and resilience.

Based on the above analysis, it can be concluded that by innovatively combining Hyperledger Fabric and a private IPFS network, a higher level of data security and privacy protection, faster data storage and retrieval, and elastic storage and capacity expansion can be achieved. This provides more possibilities and flexibility for blockchain applications and is more suitable for various sensitive data storage and transmission scenarios.

6. Conclusions

In summary, this paper presents a comprehensive approach to enhance the integrity authentication of remote-sensing image data. The key contributions of this paper revolve around the introduction of the Perceptual Hash Secure Storage and Transmission Framework (PH-SSTF). This framework excels at securely storing and transmitting original perceptual hash values by integrating a private InterPlanetary File System (IPFS) network and Hyperledger Fabric. It effectively bridges a critical gap in existing research, establishing a secure infrastructure for remote-sensing image authentication.

The practical implementation and scalability testing of PH-SSTF have been effectively demonstrated, underscoring its real-world viability and scalability. Beyond fortifying data security and privacy, this approach streamlines data storage and retrieval processes while offering flexible capacity scaling options.

Moreover, this paper introduces an advanced perceptual hashing algorithm tailored specifically for remote-sensing images. This algorithm leverages the capabilities of NSCT to meticulously capture and represent multi-band features, ensuring exceptional sensitivity to tampering. Extensive experimentation across diverse landscapes and scenarios has unequivocally confirmed the algorithm's exceptional robustness and wide-ranging applicability.

Looking ahead, extending this method to other domains with critical data integrity and privacy requirements, such as medical imaging and environmental monitoring, will open exciting interdisciplinary research prospects. Furthermore, future research directions may encompass the exploration of unsupervised learning methods to develop even more potent remote-sensing image perceptual hashing algorithms based on deep hashing learning. Additionally, the integration of artificial intelligence and blockchain techniques for automated tampering localization and real-time anomaly detection holds the potential to

enhance data security further. This work will lay a solid foundation for ensuring trust and authenticity in an increasingly data-driven world.

Author Contributions: All authors made a valuable contribution to this paper. D.X. and N.R. conceived, researched, completed the experiment, and wrote the paper; C.Z. contributed research framing, ideas, context, and wordsmithing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China, Grant 41971338 and 42071362; in part by the National Key Research and Development Program of China, Grant 2022YFC3803600.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors thank the anonymous referees for their constructive comments, which improved the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This appendix provides a comprehensive overview of the construction methodology for the prototype system of PH-SSTF.

For the IPFS part, this study uses the Kubo and IPFS-Cluster client to build a local private IPFS network in the private cluster mode, which is configured by setting the environment variable `LIBP2P_FORCE_PNET` and restricting access to nodes with the same `swarm.key` file. The operation is also integrated with the web interface. The interface for the perceptual hash IPFS storage operation is presented in Figure 6a. Specifically, the data sender uploads the transmitted data to the private IPFS network via the web interface by selecting the file with the perceptual hash value PH that contains the original perceptual hash image. The server receives the file from the client via the Post method, calls the toolkit API to upload the file to IPFS, and then returns the file hash value returned by IPFS to the client. When downloading, the client receives the file hash through the network, calls the toolkit API to download the file according to the hash value, and then sends the file back to the client.

The perceptual hash secure storage prototype system startup process corresponds to the PH-SSTF initialization phase and the request phase, first completing the Fabric network environment deployment, using the configuration file method to create organization nodes on local servers and Aliyun ECS servers, and generating certificates and data files as well as system and channel founding blocks through the Fabric's modules. Then complete the main steps of the perceptual hash secure storage prototype system startup, including ① The sender (DS) compiles the system project and generates executable prototype system files; ② DS starts the Hyperledger Fabric network on which the prototype system runs; ③ initializes the Fabric SDK Go; ④ creates the Channel A and establishes the block of Organization 1 (Org1) blockchain node peer0, after the receiver (DR) identity is verified, DS adds the blockchain node peer1 representing DR to the specified channel; ⑤ completes the installation and instantiation of the perceptual hash storage chaincode; ⑥ creates the channel client; ⑦ starts the web service. After the service is started, the DS and DR can register and query the perceptual hash through the browser.

Figure 6b shows the interface for perceptual hash registration. The sender (DS) can input the perceptual hash IPFS address and relevant transmission information directly into the perceptual hash storage interface and submit the registration request. After successful registration, DS sends the TxID returned by the contract and the private IPFS network `swarm.key` shared vital file along with the original file to the receiver (DR). Figure 6c illustrates the perceptual hash query operation interface. Corresponding to the PH-SSTF integrity authentication phase, DR can query the data by entering the transaction number (TxID) in the query interface upon receiving it. The query result enables the receiver to

obtain the original actual perceptual hash value of the remote-sensing image for integrity authentication or the sender to trace the data receiver for the maintenance of rights and responsibilities in the case of data tampering.

References

1. Avtar, R.; Kouser, A.; Kumar, A.; Singh, D.; Misra, P.; Gupta, A.; Yunus, A.P.; Kumar, P.; Johnson, B.A.; Dasgupta, R.; et al. Remote Sensing for International Peace and Security: Its Role and Implications. *Remote Sens.* **2021**, *13*, 439. [[CrossRef](#)]
2. Chen, W.; Han, B.; Yang, Z.; Gao, X. MSSDet: Multi-Scale Ship-Detection Framework in Optical Remote-Sensing Images and New Benchmark. *Remote Sens.* **2022**, *14*, 5460. [[CrossRef](#)]
3. Dempster, A. GNSS Data as Court Evidence: Lessons from Remote Sensing. In Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018), Miami, FL, USA, 28 September 2018; pp. 1427–1433. [[CrossRef](#)]
4. Nurhaida, I.; Ramayanti, D.; Riesaputra, R. Digital Signature & Encryption Implementation for Increasing Authentication, Integrity, Security and Data Non-Repudiation. *Int. Res. J. Comput. Sci.* **2017**, *4*, 4–14.
5. Hou, X.; Yang, H.; Min, L. An Efficient Semi-Fragile Watermarking Scheme for Tamper Localization and Recovery. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *322*, 052055. [[CrossRef](#)]
6. Serra-Ruiz, J.; Megías, D. Watermarking Scheme for Tampering Detection in Remote Sensing Images Using Variable Size Tiling and DWT. In *Satellite Data Compression, Communications, and Processing VI*; SPIE: San Diego, CA, USA, 2010; Volume 7810, pp. 72–82. [[CrossRef](#)]
7. Serra-Ruiz, J.; Megías, D. A Novel Semi-Fragile Forensic Watermarking Scheme for Remote Sensing Images. *Int. J. Remote Sens.* **2011**, *32*, 5583–5606. [[CrossRef](#)]
8. Serra-Ruiz, J.; Qureshi, A.; Megías, D. Entropy-Based Semi-Fragile Watermarking of Remote Sensing Images in the Wavelet Domain. *Entropy* **2019**, *21*, 847. [[CrossRef](#)]
9. Ding, K.; Meng, F.; Liu, Y.; Xu, N.; Chen, W. Perceptual Hashing Based Forensics Scheme for the Integrity Authentication of High Resolution Remote Sensing Image. *Information* **2018**, *9*, 229. [[CrossRef](#)]
10. Weng, L.; Preneel, B. A Secure Perceptual Hash Algorithm for Image Content Authentication. In *Communications and Multimedia Security*; De Decker, B., Lapon, J., Naessens, V., Uhl, A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; pp. 108–121. [[CrossRef](#)]
11. Zhang, X.; Yan, H.; Zhang, L.; Wang, H. High-Resolution Remote Sensing Image Integrity Authentication Method Considering Both Global and Local Features. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 254. [[CrossRef](#)]
12. Ding, K.; Chen, S.; Meng, F. A Novel Perceptual Hash Algorithm for Multispectral Image Authentication. *Algorithms* **2018**, *11*, 6. [[CrossRef](#)]
13. Ding, K.; Chen, S.; Wang, Y.; Liu, Y.; Zeng, Y.; Tian, J. AAU-Net: Attention-Based Asymmetric U-Net for Subject-Sensitive Hashing of Remote Sensing Images. *Remote Sens.* **2021**, *13*, 5109. [[CrossRef](#)]
14. Ding, K.; Yang, Z.; Wang, Y.; Liu, Y. An Improved Perceptual Hash Algorithm Based on U-Net for the Authentication of High-Resolution Remote Sensing Image. *Appl. Sci.* **2019**, *9*, 2972. [[CrossRef](#)]
15. Ding, K.; Liu, Y.; Xu, Q.; Lu, F. A Subject-Sensitive Perceptual Hash Based on MUM-Net for the Integrity Authentication of High Resolution Remote Sensing Images. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 485. [[CrossRef](#)]
16. Ding, K.; Zeng, Y.; Wang, Y.; Lv, D.; Yan, X. AGIM-Net Based Subject-Sensitive Hashing Algorithm for Integrity Authentication of HRRS Images. *Geocarto Int.* **2023**, *38*, 2168071. [[CrossRef](#)]
17. Ding, K.; Chen, S.; Zeng, Y.; Wang, Y.; Yan, X. Transformer-Based Subject-Sensitive Hashing for Integrity Authentication of High-Resolution Remote Sensing (HRRS) Images. *Appl. Sci.* **2023**, *13*, 1815. [[CrossRef](#)]
18. Lee, S.; Seok, H.-W.; Lee, K.; In, H.P. B-GPS: Blockchain-Based Global Positioning System for Improved Data Integrity and Reliability. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 186. [[CrossRef](#)]
19. Krichen, M.; Ammi, M.; Mihoub, A.; Almutiq, M. Blockchain for Modern Applications: A Survey. *Sensors* **2022**, *22*, 5274. [[CrossRef](#)] [[PubMed](#)]
20. Qingqing, H.; Yuan, J.; Jian, Y. Improved Fusion Method for Infrared and Visible Remote Sensing Imagery Using NSCT. In Proceedings of the 2011 6th IEEE Conference on Industrial Electronics and Applications, Beijing, China, 21–23 June 2011; pp. 1012–1015. [[CrossRef](#)]
21. Chen, P.; Zhang, Y.; Jia, Z.; Yang, J.; Kasabov, N. Remote Sensing Image Change Detection Based on NSCT-HMT Model and Its Application. *Sensors* **2017**, *17*, 1295. [[CrossRef](#)] [[PubMed](#)]
22. Da Cunha, A.L.; Zhou, J.; Do, M.N. The Nonsubsampled Contourlet Transform: Theory, Design, and Applications. *IEEE Trans. Image Process.* **2006**, *15*, 3089–3101. [[CrossRef](#)]
23. Du, C.; Gao, S. Remote Sensing Image Fusion Based on Nonlinear IHS and Fast Nonsubsampled Contourlet Transform. *J. Indian Soc. Remote Sens.* **2018**, *46*, 2023–2032. [[CrossRef](#)]
24. Wang, S.; Shen, Y. Multi-Modal Image Fusion Based on Saliency Guided in NSCT Domain. *IET Image Process.* **2020**, *14*, 3188–3201. [[CrossRef](#)]
25. Dai, W.; Tan, L.; Yang, A. Fusion Algorithm of Infrared and Visible Images Based on Local Energy Using NSCT. In Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012; pp. 4579–4582. [[CrossRef](#)]

26. Zheng, Z.; Xie, S.; Dai, H.-N.; Chen, X.; Wang, H. Blockchain Challenges and Opportunities: A Survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [[CrossRef](#)]
27. Yang, R.; Wakefield, R.; Lyu, S.; Jayasuriya, S.; Han, F.; Yi, X.; Yang, X.; Amarasinghe, G.; Chen, S. Public and Private Blockchain in Construction Business Process and Information Integration. *Autom. Constr.* **2020**, *118*, 103276. [[CrossRef](#)]
28. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the Thirteenth EuroSys Conference, EuroSys'18, Porto, Portugal, 23–26 April 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–15. [[CrossRef](#)]
29. Ma, C.; Kong, X.; Lan, Q.; Zhou, Z. The Privacy Protection Mechanism of Hyperledger Fabric and Its Application in Supply Chain Finance. *Cybersecurity* **2019**, *2*, 5. [[CrossRef](#)]
30. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2014**, arXiv:1407.3561. [[CrossRef](#)]
31. Mukne, H.; Pai, P.; Raut, S.; Ambawade, D. Land Record Management Using Hyperledger Fabric and IPFS. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–8. [[CrossRef](#)]
32. Nyalety, E.; Parizi, R.M.; Zhang, Q.; Choo, K.-K.R. BlockIPFS—Blockchain-Enabled Interplanetary File System for Forensic and Trusted Data Traceability. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 18–25. [[CrossRef](#)]
33. Li, L.; Yue, Z.; Wu, G. Electronic Medical Record Sharing System Based on Hyperledger Fabric and InterPlanetary File System. In Proceedings of the 2021 5th International Conference on Compute and Data Analysis; ICCDA 2021, Sanya, China, 2–4 February 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 149–154. [[CrossRef](#)]
34. Mani, V.; Manickam, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Hyperledger Healthchain: Patient-Centric IPFS-Based Storage of Health Records. *Electronics* **2021**, *10*, 3003. [[CrossRef](#)]
35. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.
36. Tong, X.-Y.; Xia, G.-S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-Cover Classification with High-Resolution Remote Sensing Images Using Transferable Deep Models. *Remote Sens. Environ.* **2020**, *237*, 111322. [[CrossRef](#)]
37. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]
38. Liu, M.; Gao, H.; Xia, X.; Gui, S.; Gao, T. Perceptual Image Hashing Based on Canny Operator and Tensor for Copy-Move Forgery Detection. *Comput. J.* **2022**, bxac186. [[CrossRef](#)]
39. Yu, B.; Li, X.; Zhao, H. Virtual Block Group: A Scalable Blockchain Model with Partial Node Storage and Distributed Hash Table. *Comput. J.* **2020**, *63*, 1524–1536. [[CrossRef](#)]
40. Wahid, M.; Ahmad, N.; Zafar, M.H.; Khan, S. On Combining MD5 for Image Authentication Using LSB Substitution in Selected Pixels. In Proceedings of the 2018 International Conference on Engineering and Emerging Technologies (ICEET), Lahore, Pakistan, 22–23 February 2018; pp. 1–6. [[CrossRef](#)]
41. Guo, M.; Liu, H.; Xu, Y.; Huang, Y. Building Extraction Based on U-Net with an Attention Block and Multiple Losses. *Remote Sens.* **2020**, *12*, 1400. [[CrossRef](#)]
42. Ding, K.; Chen, S.; Yu, J.; Liu, Y.; Zhu, J. A New Subject-Sensitive Hashing Algorithm Based on MultiRes-RCF for Blockchains of HRRS Images. *Algorithms* **2022**, *15*, 213. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.