applied sciences

MDPI

Article

Transformer-Based Subject-Sensitive Hashing for Integrity Authentication of High-Resolution Remote Sensing (HRRS) Images

Kaimeng Ding [1,2], Shiping Chen [3], Yue Zeng [4,*], Yingying Wang [5,*] and Xinyun Yan [1,2]

1 School of Networks and Tele-Communications Engineering, Jinling Institute of Technology, Nanjing 211169, China
2 Jiangsu AI Transportation Innovations & Applications Engineering Research Center, Nanjing 211169, China
3 CSIRO Data61, Sydney, NSW 1710, Australia
4 School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China
5 School of Intelligent Science and Control Engineering, Jinling Institute of Technology, Nanjing 211169, China
* Correspondence: zengy@jit.edu.cn (Y.Z.); wyy@jit.edu.cn (Y.W.); Tel.: +86-189-1380-6256 (Y.Z.); +86-181-6809-2336 (Y.W.)

Featured Application: The transformer based subject-sensitive hashing algorithm proposed in this paper could be applied to data security of HRRS images to provide integrity authentication services for later use of HRRS images, and to generate watermark information for digital watermarks.

Abstract: The implicit prerequisite for using HRRS images is that the images can be trusted. Otherwise, their value would be greatly reduced. As a new data security technology, subject-sensitive hashing overcomes the shortcomings of existing integrity authentication methods and could realize subject-sensitive authentication of HRRS images. However, shortcomings of the existing algorithm, in terms of robustness, limit its application. For example, the lack of robustness against JPEG compression makes existing algorithms more passive in some applications. To enhance the robustness, we proposed a Transformer-based subject-sensitive hashing algorithm. In this paper, first, we designed a Transformer-based HRRS image feature extraction network by improving Swin-Unet. Next, subject-sensitive features of HRRS images were extracted by this improved Swin-Unet. Then, the hash sequence was generated through a feature coding method that combined mapping mechanisms with principal component analysis (PCA). Our experimental results showed that the robustness of the proposed algorithm was greatly improved in comparison with existing algorithms, especially the robustness against JPEG compression.

Keywords: deep learning; HRRS images; subject-sensitive hashing; transformer; U-net; perceptual hashing; integrity authentication

Citation: 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. https://doi.org/10.3390/app13031815

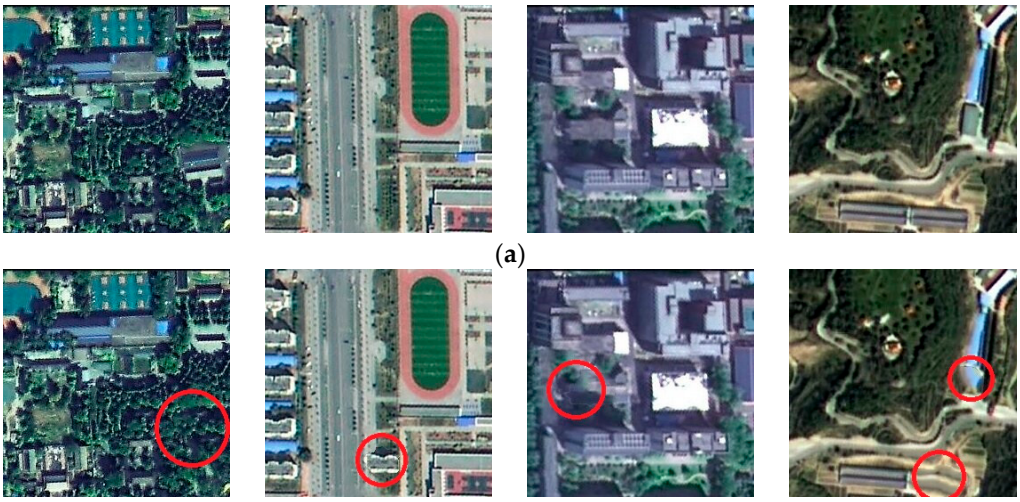
Academic Editors: Saro Lee and Hyung-Sup Jung

Received: 15 December 2022
Revised: 23 January 2023
Accepted: 28 January 2023
Published: 31 January 2023

1. Introduction

High-resolution remote sensing (HRRS) images have come to play an increasingly important role in urban planning, surveying, mapping, and land use. However, the implicit prerequisite for using HRRS images is that the images can be trusted. If tampered HRRS images are used, erroneous analytical conclusions may be drawn and wrong decisions may be made.

Several sets of comparisons (before and after HRRS images were tampered with) are shown in Figure 1. Without integrity authentication technology, it would be difficult for users to determine whether the HRRS image shown in Figure 1b had been tampered with. In such a case, the credibility and value of the HRRS image would be reduced, even losing its value entirely.



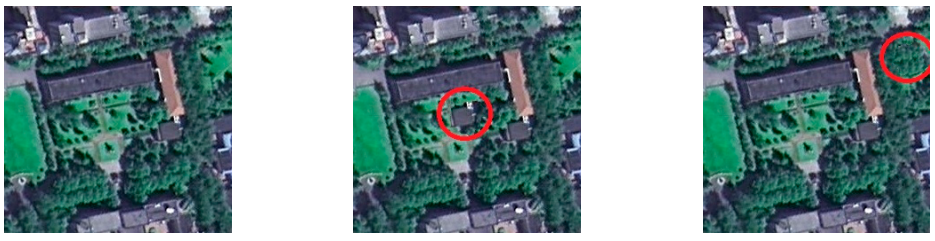
(**a**)

(**b**)

Figure 1. Comparisons before and after HRRS image tampering: (**a**) original images, (**b**) tampered-with HRRS images. From left to right, the alterations of each figure are: a building in the enclosed area replaced with trees, a building added to the enclosed area, a building in the enclosed area defaced, and buildings in the enclosed area deleted. Photoshop is the tool used to tamper with the images.

Tampering with HRRS images, similar to the above, not only reduces the credibility of HRRS images, but also negatively impacts social stability. For example, if a city planning department uses HRRS images that have been tampered with, it may lead to incorrect planning. If a military department uses HRRS images that have been tampered with, it may issue incorrect instructions. Although integrity authentication methods, such as fragile watermark, cryptographic hash, and perceptual hashing solve the above problems to a certain extent, there is still considerable room for improvement. The most prominent problem is that mainstream integrity authentication technologies are oriented to general integrity authentication problems. Thus, they cannot solve the authentication problem in a specific field.

Compared with ordinary images, HRRS images mainly reflect the characteristics of ground objects and often have no clear theme; this makes HRRS images more difficult than ordinary images for the human eye to recognize. In actual tampering of HRRS images, attackers often add or remove specific types of image content instead of modifying images at will. This kind of content-biased behavior is inherently subject-sensitive and is more concealed and harmful than random tampering. Moreover, due to the human visual attention mechanism, alteration of the region of interest to the human eye is more noticeable and more damaging to HRRS images. For example, in the examples shown in Figure 2, although the area of Figure 2b that had been tampered with was even smaller than that of Figure 2c, the tampering in Figure 2b was more destructive to HRRS, especially for users for whom buildings are the main object of use.



(**a**) (**b**) (**c**)

Figure 2. Comparison of subject-related and subject-unrelated tampering (taking buildings as an example of a subject): (**a**) original HRRS image, (**b**) subject-related tampering, (**c**) subject-unrelated tampering.

As a new data security technology derived from perceptual hashing [1–5], subject-sensitive hashing [6] can perform more stringent integrity authentication on image areas of interest to users. However, subject-sensitive hashing methods still have certain deficiencies, especially since subject-sensitive hashing has not been proposed for a long time. The outstanding problem is that the robustness of existing subject-sensitive hashing algorithms require further improvement in order to meet actual authentication needs. The most typical problem is that the robustness against JPEG (Joint Photographic Experts Group) compression is not ideal.

The success of Transformer [7] in multiple research fields has provided new research ideas for solving the existing problems of subject-sensitive hashing. Dosovitskiy et al. [8] applied Transformer to computer vision for the first time; the proposed ViT (Vision Transformer) turned image data into a sequence of tokens through splitting and flattening, achieving excellent performance in the study of image classification. In this paper, Transformer was applied to subject-sensitive hashing, and an improved Swin-Unet network was proposed to improve the robustness of subject-sensitive hashing.

The main contributions of this paper were as follows:

1. To the best of our knowledge, this was the first study to apply Transformer to integrity authentication of HRRS images, and the first research on Transformer-based subject-sensitive hashing;
2. We modified the Swin-Unet structure to make the model more suitable for HRRS subject-sensitive hashing, which helped the algorithm comprehensively outperform existing algorithms, including the original Swin-Unet;
3. We proposed a feature encoding method combining the mapping mechanism and principal component analysis (PCA) for the generation of hash sequences.

2. Related Work

2.1. Perceptual Hashing

Perceptual hashing, also known as perceptual hash algorithms, is a family of algorithms that generate content-based hash sequences, including perceptual image hashing, perceptual audio hashing [9], and perceptual video hashing [10]. Unlike cryptographic hash, which takes binary representation of an image to generate hash sequences [11,12], perceptual hashing takes the content of an image to generate the hash sequence.

Perceptual image hashing has received widespread attention and has been studied in depth. Qin et al. [13] applied singular value decomposition (SVD) and Gaussian low-pass filtering on color image perceptual hashing to improve the robustness of the algorithm. Tang et al. [14] proposed a perceptual hashing algorithm with the histogram of CVA (Color vector angle), which was able to resist rotation with arbitrary angle and reach good discrimination. Hamid et al. [15] proposed a perceptual hash algorithm, based on the difference of Laplacian pyramids, which was able to detect minute-level tampering. Biswas et al. [16] proposed a perceptual hashing for face verification, which was able to protect against AERO (Adversarial Eye Region Occlusion) attack. Wang et al. [17] proposed a perceptual hash method for image tampering detection and localization which used hybrid features to generate hash sequence. Huang et al. [18] proposed a perceptual hash method for copy detection of images.

2.2. Subject-Sensitive Hashing

Subject-sensitive hashing, also known as subject-sensitive hash algorithm, inherently has the robustness and sensitivity to tampering of perceptual hashing [6]. It can meet customized requirements in specific fields for integrity authentication and perform sensitive integrity authentication for specific subjects (such as buildings or roads). Subject-sensitive hashing is not an upgraded version of perceptual hashing, nor can it replace perceptual hashing.

However, it is difficult for traditional image technology to define subject-related features. Subject-sensitive feature extraction is essentially a feature extraction process in

which information of specific feature types is weighted. Deep learning has an excellent capacity for feature expression, and can achieve feature-weighted feature extraction by learning specific samples, reducing the complexity of artificially-designed features [19–21].

Existing subject-sensitive hashing methods mainly use convolutional neural networks (CNNs) to achieve feature extraction of HRRS images. In addition to MUM-net [6], other models, such as U-net [22], Attention U-net [23], M-net [24], and MultiResUNet [25] can also be used to implement subject-sensitive hashing. However, CNNs need to continuously stack convolutional layers to achieve the extraction of image local information to image global information. This causes the model to be bloated, brings about the problem of gradient disappearance, and even leaves the network unable to converge. Moreover, methods based on CNN adopt the method of feature downsampling, which reduces the ability of the algorithm to detect small-scale tampering. In addition, current CNN-based, subject-sensitive hashing algorithms are generally less robust to pixel-level content-preserving operations. The outstanding performance issue is that the robustness of these algorithms against JPEG compression needs to be further optimized. Although the introduction of the attention mechanism enhances the algorithm robustness to a certain extent [26], it still needs further improvement.

With the interdisciplinary development of deep learning, Transformer stands out in computer vision (CV) and provides new possibilities for solutions to problems faced by CNN-based subject-sensitive hashing [27]. Since ViT replaced the convolutional structure with the Transformer structure for vision tasks, Transformer has been successfully applied in the fields of image segmentation [28,29], image classification [30,31], image super-resolution [32,33], image restoration [34], and object detection [35,36].

Transformer-based computer vision methods exploit the self-attention mechanism in Transformer to explore long-range dependencies and learn the attentional interactions of different patch tokens. In VIT, the input image is cropped into fixed-size image patches to transform the image into sequence data that the Transformer structure can process. Then, each image patch is changed into a one-dimensional vector, which is linearly mapped and then added to position encoding. Inspired by ViT, TransUNet [37] adopts a Transformer-based encoder to process image patch sequences and combines the characteristics of U-net. Swin Transformer (Shifted Windows Transformer) [38] uses a hierarchical strategy to restrict the attention computation to a window, aiming to introduce a locality operation, similar to CNN convolution, and significantly reduce the computational cost. Swin-Unet [39] combines the characteristics of Swin Transformer and U-net, and is a pure Transformer-based image segmentation network. Swin-Unet draws on the network structure characteristics of U-net, and sends the tokenized image blocks to U-shaped network architecture composed of pure Transformer through skip connections for local and global feature learning. This method of using small image blocks as the basic processing unit (instead of pixels) brings new possibilities to enhance the robustness of pixel-level operations.

Although Swin-Unet can be directly used for subject-sensitive hash algorithms, just like AAU-net or U-net, subject-sensitive hashing needs to comprehensively consider robustness and tampering sensitivity. Differing from application fields of Transformer such as image classification, subject-sensitive hashing does not necessarily function better with more feature extraction. Nevertheless, too many low-level features are not conducive to the algorithm's robustness. In this article, we built an improved Swin-Unet and proposed a new subject-sensitive hashing based on this network.

3. Method

In this section, we have presented the Transformer-based subject-sensitive hash algorithm. First, the network structure of our improved Swin-Unet was introduced. Then, the feature-encoding, method combining the mapping mechanism with PCA, was explained. Subsequently, the overall flow of the subject-sensitive hash algorithm was discussed. Finally, the integrity authentication process of HRRS images, based on our algorithm, was introduced.

3.1. Architecture of Improved Swin-Unet

As shown in Figure 3, the network framework of the improved Swin-Unet consisted of the following parts: encoder (left part), decoder (right part), bottleneck block (bottom part), and skip connection (middle jumper section). In each part, the Swin Transformer model was the core module.

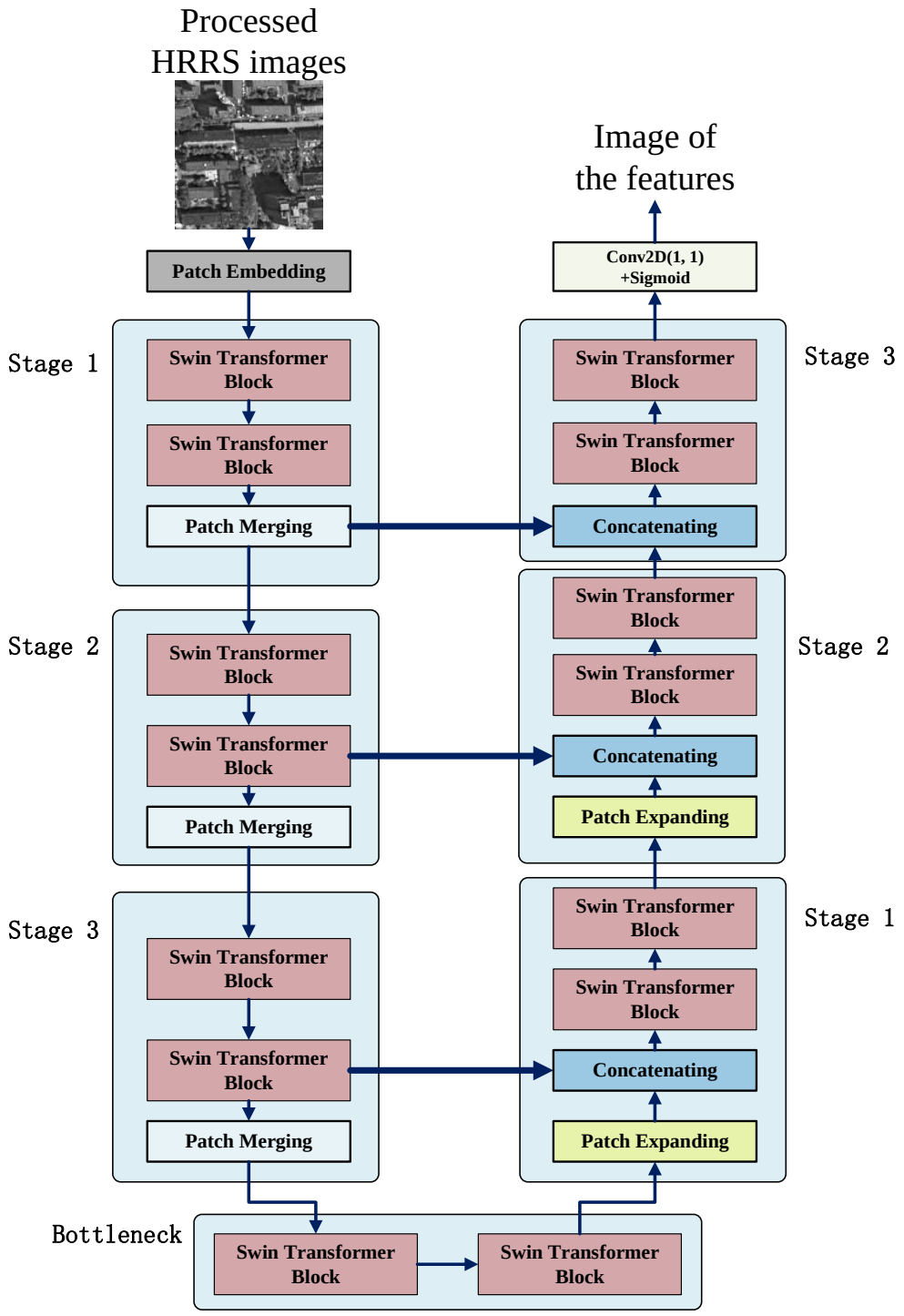


Figure 3. Architecture of improved Swin-Unet.

The encoder split the HRRS image into non-overlapping blocks of size 4 × 4 and converted the input image into sequence embeddings. The encoder consisted of multiple modules to generate the hierarchical representations of HRRS image, and each module contained two Swin Transformers and one Patch Merging layer. The Swin Transformer

module [38] was built on a shifted window and was responsible for learning features. The Patch Merging [39] was responsible for downsampling operation and increasing the feature dimension.

The decoder also consisted of multiple modules. Except for the last module, each module contained two Swin Transformers, one Patch Expanding and one Concatenating. Patch Expanding [39] was used to perform feature dimension and upsampling. Similar to the skip connection of original Swin Unet, Concatenating was responsible for feature fusion, to alleviate the lost spatial information caused by Patch Merging. It should be pointed out that there was no Patch Expanding on the last module.

The most obvious differences between our improved Swin-Unet and the original Swin-Unet were:

(1) In order to improve the algorithm's robustness, the patch expanding layer of the last module of the Swin-Unet decoder was canceled. After all, higher-level Transformers focus on encoding relatively complex high-level semantic information, and overly complex information could weaken algorithm robustness;
(2) The position of the Skip connection between the first module of the encoder and the last module of the decoder was changed to cater to the cancellation of the Patch Expanding layer of the last module of the decoder, which reduced the impact of extraneous features on the tampering sensitivity of the algorithm;
(3) Due to the network structure, the output image size of the improved Swin-Unet was 128 × 128 pixels, while the output image size of the original Swin-Unet was 224 × 224 pixels—the input image size for both models was 224 × 224 pixels. This input-output asymmetry helped to reduce the impact of redundant information and improve the performance of hashing algorithm.

Although the network was structurally different from Swin-Unet, we did not rename it. Instead, we called it the improved Swin-Unet.

3.2. Feature Coding Based on Mapping Mechanism and PCA

The output of the improved Swin-Unet was a single-channel feature image. To enhance the robustness of the algorithm, we proposed a feature-encoding method that combined the mapping mechanism and principal component analysis to further suppress the non-feature pixels of the feature image. The mapping mechanism we designed was based on the sine function, as follows:

$$fm(x,y) = \frac{255.0 \times (1 + \sin(f(x,y) \times \theta \times \frac{\pi}{255} - \frac{\pi}{2}))}{2} \quad (1)$$

Above, $f(x,y)$ represents the numeric value of image pixel, $fm(x,y)$ represents the mapped pixel value, π (radians) stands for 180 degrees, and θ is the adjustment factor. Obviously, once mapped, the value of the pixels whose value is less than the median value will be reduced, which is beneficial for enhancing the robustness and tampering sensitivity of the algorithm. When θ is set to 1, the mapping examples of the pixel value are shown in Table 1.

Table 1. Examples of mapping of different pixel values.

Raw Value of Pixel	Mapped Value of Pixel
60	33.27
64	37.62
110	100.22
128	128.28
160	177.19
255	255

As shown in Table 1, this mapping mechanism, based on sine function, was able to suppress small values while enhancing large values.

The feature matrix processed by the mapping mechanism was then decomposed by PCA. Since the first few columns of the principal components summarized the main features of the matrix, we chose the first column of principal components as subject-sensitive features of the HRRS image. The first column of principal components were binarized into a 0–1 sequence.

3.3. Overview of the Subject-Sensitive Hashing Algorithm

As shown in Figure 4, the flow of our proposed subject-sensitive hash algorithm mainly included: preprocessing of the image, image feature extraction based on the improved Swin-Unet, and feature encoding based on the mapping mechanism and PCA.

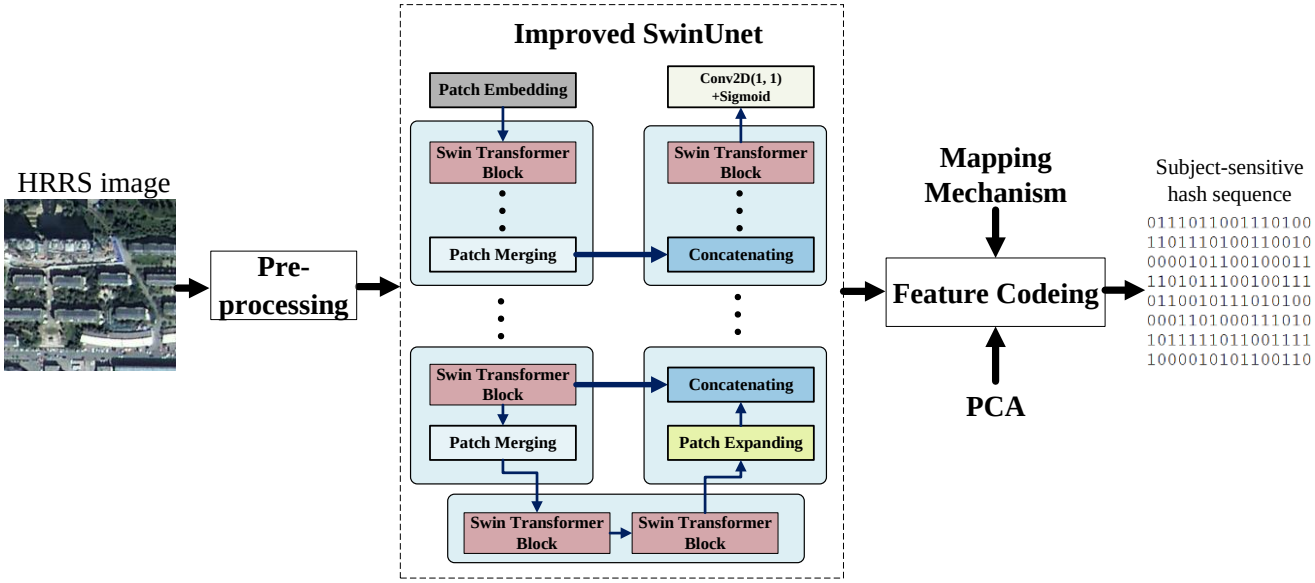


Figure 4. Flow of the proposed subject-sensitive hash algorithm.

(1) Preprocessing was designed to process the HRRS image so that it met the requirements of the input size of the improved Swin-Unet, which was 224 × 224 pixels. If the size of the HRRS image was large (for example, larger than 512 × 512 pixels), it could be divided into non-overlapping grid cells by grid division to ensure the accuracy of integrity authentication. Since grid division-based methods have been discussed repeatedly by existing algorithms [6,26], we chose not to repeat them in this paper.

(2) For feature extraction, the preprocessed HRRS image was input into the trained improved Swin-Unet to obtain the feature map of the corresponding image. The training process of the improved Swin-Unet was discussed in Section IV.

(3) Feature coding: the feature extracted by the improved Swin-Unet was essentially a two-dimensional matrix of pixel gray values. After feature encoding, based on the mapping mechanism and PCA, the obtained one-dimensional 0–1 sequence was encrypted by the AES (Advanced Encryption Standard) algorithm [40,41] to get the hash sequence, denoted as *SH*.

3.4. Integrity Authentication Process

The subject-sensitive hash sequence of an HRRS image needs to be stored together with the corresponding image. When the content integrity of the image needs to be authenticated, the subject-sensitive hash sequence of the image is recalculated and compared with the previous hash sequence to determine whether the HRRS image has been tampered with. Similar to existing subject-sensitive hashing, our algorithm also employed the normalized Hamming distance [42] to compare differences between hash sequences.

We denoted the hash sequences of the original image and the image to be authenticated as *SH* and *SH'*, respectively. Then, the normalized hamming distance was as follows:

$$Dis = \left(\sum_{i=1}^{Length} \left| SH(i) - SH'(i) \right| \right) / Length \quad (2)$$

where *Length* represents the length of the hash sequence. Obviously, the value of normalized hamming distance is a decimal in the range of 0 to 1. The larger the value of *Dis*, the greater the difference between the two hash sequences; conversely, the smaller the value, the smaller the difference between the two hash sequences. If *Dis* is 0, it means the two hash sequences are completely consistent.

4. Experiments

In this part, we conducted experiments to compare our Transformer-based subject-sensitive hash algorithm with other algorithms.

4.1. Implementation Details and Datasets

We leveraged Keras, a high-level neural network framework to implement our improved Swin-Unet. Since Keras uses Python as the programming language, we used Python to implement the proposed subject-sensitive hash algorithm. The hardware platform was: Intel CPU I7-9700K; RTX2080Ti GPU with 11G of memory; 32G RAM. In the training process, batch size was set to 4 due to the size of network model and memory of GPU; regarding the optimization of model weights and biases, we chose the Adam [43] as optimizer. In addition, we trained the model from scratch for 100 epochs. Improved Swin-Unet had a total of 61,947,912 parameters. The size of the model after training was 236M, which was slightly smaller than the original Swin-Unet.

According to the structural similarity of the network model, we compared the following model-based algorithms with our algorithm: original Swin-Unet [39], MUM-Net [6], U-Net [22], Attention U-Net [23], M-Net [24], MultiResU-Net [25], AAU-Net [26], Attention R2U-Net [44], and Attention ResU-Net [45,46]. In order to facilitate comparison with existing algorithms, the flow of each algorithm was consistent with the MUM-Net-based algorithm in [6] and AAU-net-based algorithm in [26]; that is, the process did not contain feature mapping.

Experiments on subject-sensitive hashing algorithms involve multiple types of datasets: datasets for training deep neural networks, datasets for testing algorithm robustness, and datasets for testing algorithm tampering sensitivity. We used the dataset for training AAU-net as the dataset to train the improved Swin-Unet and participating network model, which was convenient when comparing the performance of our algorithm with existing algorithms. This dataset as a variant dataset based on the WHU building dataset [47], with some hand drawn training samples added.

4.2. Evaluation Indicator

The evaluation indicators of subject-sensitive hashing mainly include: robustness, tampering sensitivity, security, computational performance, and digestibility.

(1) Robustness. For a single HRRS image, robustness means that, after the image undergoes an operation, the hash sequence does not change, or the change is lower than a preset threshold. Due to the strong chance of a single or a small number of test data, we measured the robustness using the proportion of HRRS images whose hash sequence variations were lower than the preset threshold T. The calculation method was as follows:

$$R(T) = \frac{Num_R}{Num_T} \quad (3)$$

where Num_T represents the number of instances participating in the test and Num_R represents the number of instances whose hash sequence has not changed or has changed below the threshold T.

(2) Tampering sensitivity. Like cryptographic hash and fragile watermarking, a subject-sensitive hash algorithm has to detect whether the image content has been tampered with, which means that tampering sensitivity is an important evaluation indicator for subject-sensitive hashing. For a single instance of image tampering, the hash sequences before and after the image is tampered with should change by a greater magnitude than the threshold T. Similar to the robustness test, tampering sensitivity also requires more test instances to be more convincing. We used the proportion of instances in which tampering was detected to describe tampering sensitivity, as follows:

$$S(T) = \frac{Num_S}{Num_T} \tag{4}$$

where Num_T represents the number of instances participating in the test and Num_S represents the number of tampering instances that were successfully detected.

(3) Digestibility. The storage space occupied by the hash sequence should be as small as possible—that is, the hash sequence should be as short as possible.

(4) Computational performance. Computational performance requires that the time to calculate and compare hash sequences should be as short as possible. In fact, as the comparison of hash sequences is very efficient due to digestibility, computational performance generally focuses on the efficiency of generating hash sequences.

(5) Security. The security of subject-sensitive hashing means that the content of the image cannot be obtained from the subject-sensitive hash sequence.

4.3. Examples of Integrity Authentication

Before fully comparing our proposed algorithm and existing algorithms, we first conducted a preliminary comparison of the algorithms through a set of integrity authentication instances, as shown in Figure 5. Figure 5a shows the original HRRS image stored in TIFF format, while Figure 5b–h can be divided into 2 groups: instances that do not change the content of the image, and instances that change the content of the image.

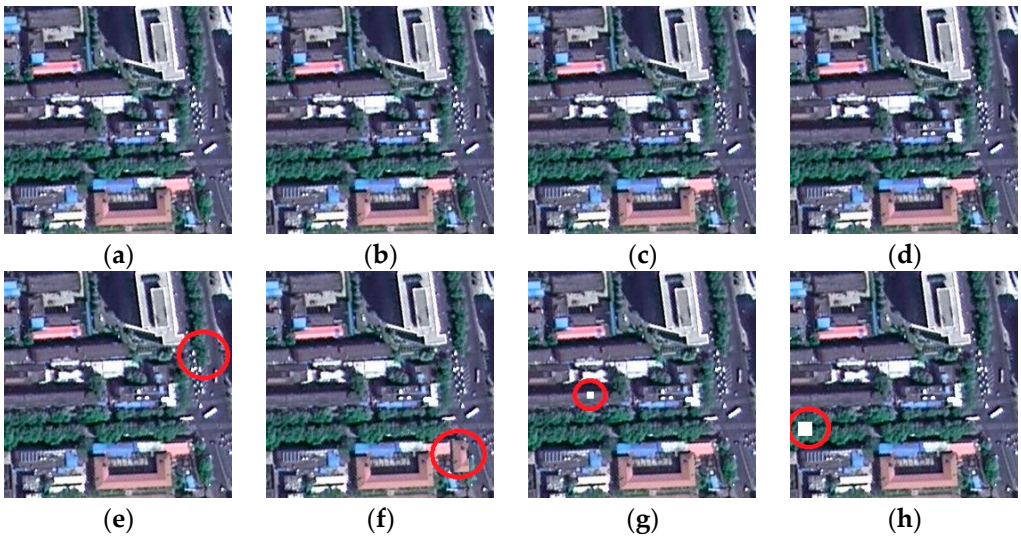


Figure 5. Integrity Authentication Examples for HRRS image: (**a**) original HRRS image; (**b**) formatted image (TIFF format to BMP format); (**c**) the image after watermark embedding (least significant bit algorithm as an example, embedding 128-bit watermark information); (**d**) JPEG-compressed image; (**e**) subject-unrelated tampering; (**f**) subject-related tampering; (**g**) 8 × 8 random tampering; (**h**) 16 × 16 random tampering.

In the first group, Figure 5b–d, the operation of the images was: format conversion (TIFF format to BMP format), digital watermark embedding (using the least significant bit algorithm), and 95% JPEG compression. In the first group, each HRRS image carried the same content as the original image shown in Figure 5a. The human eye could not even distinguish whether these images were different from original; however, they were changed drastically at the binary level.

The second group, Figure 5e–h, contains an image with subject-unrelated tampering, an image with subject-related tampering, an image with an area of 8 × 8 pixels randomly tampered with, and an image with an area of 16 × 16 pixels randomly tampered with, respectively. Obviously, the contents of the second group of HRRS images were tampered with. Their binaries were also changed dramatically.

Normalized hamming distances between hash sequences of Figure 5a and each image of Figure 5b–h are shown in Table 2. Table 3 shows the integrity authentication results, based on Table 2, when the threshold T was 0.02.

Table 2. Normalized hamming distance of each algorithm.

Model Each Algorithm Was Based on	Figure 5b	Figure 5c	Figure 5d	Figure 5e	Figure 5f	Figure 5g	Figure 5h
	Format Conversion	Watermark Embedding	JPEG Compression	Subject-Unrelated Tampering	Subject-Related Tampering	8 × 8 Random Tampering	16 × 16 Random Tampering
MUM-Net	0	0	0.0585	0.1679	0.2500	0.2578	0.2460
MultiResUnet	0	0	0.0234	0.0546	0.0781	0.0859	0.2578
U-net	0	0	0.0312	0.0234	0.0585	0.1835	0.2460
M-net	0	0	0.0625	0.0976	0.0859	0.1210	0.2578
Attention U-Net	0	0	0.0234	0.0078	0.0625	0.2226	0.2617
Attention ResU-Net	0	0	0	0	0	0.0273	0.2734
Attention R2U-Net	0	0	0.0273	0.0664	0.0781	0.2226	0.2382
AAU-Net	0	0.0039	0.0195	0.0898	0.1054	0.2656	0.2578
Swin-Unet	0	0	0	0.0078	0.0390	0.0546	0.1562
Improved Swin-Unet (Our algorithm)	0	0	0.0039	0.0937	0.2656	0.2539	0.2773

Table 3. Integrity authentication results based on Table 2 (T = 0.02).

The Model That Each Algorithm Based on	Figure 5b	Figure 5c	Figure 5d	Figure 5e	Figure 5f	Figure 5g	Figure 5h
	Format Conversion	Watermark Embedding	JPEG Compression	Subject-Unrelated Tampering	Subject-Related Tampering	8 × 8 Random Tampering	16 × 16 Random Tampering
MUM-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
MultiResUnet	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
U-net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
M-net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
Attention U-Net	Not tampered	Not tampered	Tampered	Not tampered	Tampered	Tampered	Tampered
Attention ResU-Net	Not tampered	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered
Attention R2U-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
AAU-Net	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered
Swin-Unet	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered
Improved Swin-Unet (Our algorithm)	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered

Comparing the performance of each algorithm, shown in Table 3, we observed that only our algorithm and the AAU-Net-based algorithm simultaneously kept robustness to the operations, shown in Figure 5b–d, and tampering sensitivity to the malicious tampering, shown in Figure 5e–h. Even compared with AAU-Net-based algorithms, our algorithm performed better for the operations shown in Figure 5b–d, and more sensitively to tampering for the operations shown in Figure 5e–h. Although the Attention ResU-Net-based

algorithm and the Swin-Unet-based algorithm also had good robustness, they performed poorly in tampering sensitivity, failing to fully detect the tampering shown in Figure 5e–h.

In the actual integrity authentication, different thresholds are often set for different scenarios or authentication needs. Tables 4 and 5 show the results of integrity authentication, based on Table 2, with T set to 0.05 and 0.01, respectively.

Table 4. Integrity authentication results based on Table 2 (T = 0.05).

The Model That Each Algorithm Based on	Figure 5b	Figure 5c	Figure 5d	Figure 5e	Figure 5f	Figure 5g	Figure 5h
	Format Conversion	Watermark Embedding	JPEG Compression	Subject Unrelated Tampering	Subject Related Tampering	8 × 8 Random Tampering	16 × 16 Random Tampering
MUM-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
MultiResUnet	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered
U-net	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered
M-net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
Attention U-Net	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered
Attention ResU-Net	Not tampered	Not tampered	Not tampered	Not tampered	Not tampered	Not tampered	Tampered
Attention R2U-Net	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered
AAU-Net	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered
Swin-Unet	Not tampered	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered
Improved Swin-Unet (Our algorithm)	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered

Table 5. Integrity authentication results based on Table 2 (T = 0.01).

The Model That Each Algorithm Based on	Figure 5b	Figure 5c	Figure 5d	Figure 5e	Figure 5f	Figure 5g	Figure 5h
	Format Conversion	Watermark Embedding	JPEG Compression	Subject-Unrelated Tampering	Subject-Related Tampering	8 × 8 Random Tampering	16 × 16 Random Tampering
MUM-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
MultiResUnet	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
U-net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
M-net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
Attention U-Net	Not tampered	Not tampered	Tampered	Not tampered	Tampered	Tampered	Tampered
Attention ResU-Net	Not tampered	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered
Attention R2U-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
AAU-Net	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered	Tampered
Swin-Unet	Not tampered	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered
Improved Swin-Unet (Our algorithm)	Not tampered	Not tampered	Not tampered	Tampered	Tampered	Tampered	Tampered

As demonstrated in Table 4, the MultiResUnet- and Attention R2U-Net-based algorithms also kept robustness to Figure 5b–d and detected the tampering shown in Figure 5e–h when threshold T was 0.05. However, one can also from Table 5 that only our algorithm maintained both robustness and tampering sensitivity when T was reduced to 0.01.

Tables 2–5, our algorithm and AAU-Net-based algorithm performed better than others. Our algorithm was also more robust than AAU-Net-based algorithms. Overall, our improved Swin-Unet-based algorithm achieved the best integrity authentication results, as shown in Figure 5.

4.4. Robustness Testing of the Algorithms

The robustness of the algorithm required significant example testing to give stronger confidence to the test results. We used the dataset $Datasets_{10,000}$ in [26] to test each algorithm's robustness. $Datasets_{10,000}$ contained 10,000 test HRRS images cropped from HRRS images from GF-2 satellite, DOTA [48], WHU dataset [47], and other datasets. Each HRRS image in $Datasets_{10,000}$ was in TIFF format and sized at 256 × 256 pixels.

First, we tested each algorithm's robustness against JPEG compression. We used C++ as a programming language and used OpenCV2.4.13 interface to implement JPEG compression on the HRRS images of $Datasets_{10,000}$, where the level of JPEG compression was 95%. We used the proportion of images that maintained robustness at a specific threshold T, as shown in Equation (3) above, to describe the robustness of the algorithms. Moreover, as the setting of threshold T is often not fixed in actual integrity authentication, depending on the specific application and the strength of integrity authentication, we tested each algorithm's robustness under different thresholds; the results are shown in Table 6.

Table 6. Each algorithm's robustness against JPEG compression under different thresholds.

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	79.57%	90.92%	97.61%	99.88%	100%
MultiResUnet	76.26%	85.95%	94.73%	98.11%	99.59%
U-net	63.28%	75.06%	88.85%	96.74%	99.81%
M-net	70.22%	80.08%	92.23%	97.58%	99.66%
Attention U-Net	85.94%	92.59%	97.14%	98.88%	100%
Attention ResU-Net	86.02%	91.08%	96.07%	99.26%	99.87%
Attention R2U-Net	58.53%	69.22%	82.97%	95.52%	99.26%
AAU-Net	73.57%	81.01%	93.33%	98.76%	99.82%
Swin-Unet	99.23%	99.85%	99.97%	100%	100%
Improved Swin-Unet (our algorithm)	95.88%	99.02%	100%	100%	100%

As shown in Table 6, Transformers have a strong advantage in improving the robustness of subject-sensitive hashing: the Swin-Unet-based algorithm and our improved Swin-Unet-based algorithm both greatly improved the robustness of JPEG compression compared to existing subject-sensitive hash algorithms. With the threshold set to 0.05 or greater, both of these Transformer-based algorithms were 100% robust against JPEG compression. Even at low thresholds (such as 0.02), the robustness against JPEG compression was higher than 95%, reaching a level that existing subject-sensitive hashing algorithms have been unable to achieve.

Next, we tested the algorithm's robustness against digital watermarks. Digital watermarking information can be embedded in one band of HRRS images, or embedded in each band. Since multiband watermark embedding makes greater changes to the data, we focused on testing algorithm's robustness against multiband watermarks, as shown in Table 7.

Table 7. Each algorithm's robustness against watermark embedding.

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	62.09%	74.27%	88.18%	97.32%	99.69%
MultiResUnet	87.48%	92.12%	96.84%	99.06%	99.85%
U-net	61.66%	81.75%	85.59%	95.84%	99.28%
M-net	68.67%	88.49%	89.97%	97.45%	99.59%
Attention U-Net	90.07%	94.16%	97.25%	98.99%	99.87%
Attention ResU-Net	93.58%	96.06%	98.02%	99.77%	99.98%
Attention R2U-Net	48.69%	60.24%	77.45%	91.98%	98.51%
AAU-Net	95.09%	97.07%	98.33%	99.58%	99.85%
Swin-Unet	99.66%	99.95%	100%	100%	100%
Improved Swin-Unet (Our algorithm)	98.07%	99.74%	100%	100%	100%

As can be seen from Table 7, the Swin-Unet-based algorithm performed best against multi-band watermark embeddings, while our improved Swin-Unet-based algorithm was second to it, and superior to the other existing algorithms.

Then, we tested each algorithm's robustness against modification of a small number of pixels. We randomly selected 4 pixels of each HRRS image in $Datasets_{10,000}$ and set them to 0. Figure 6 shows a comparison of a set of images before and after modification. Obviously, the differences between the modified and original images were very small; unless the images were to be enlarged and carefully observed, the differences could not be found.

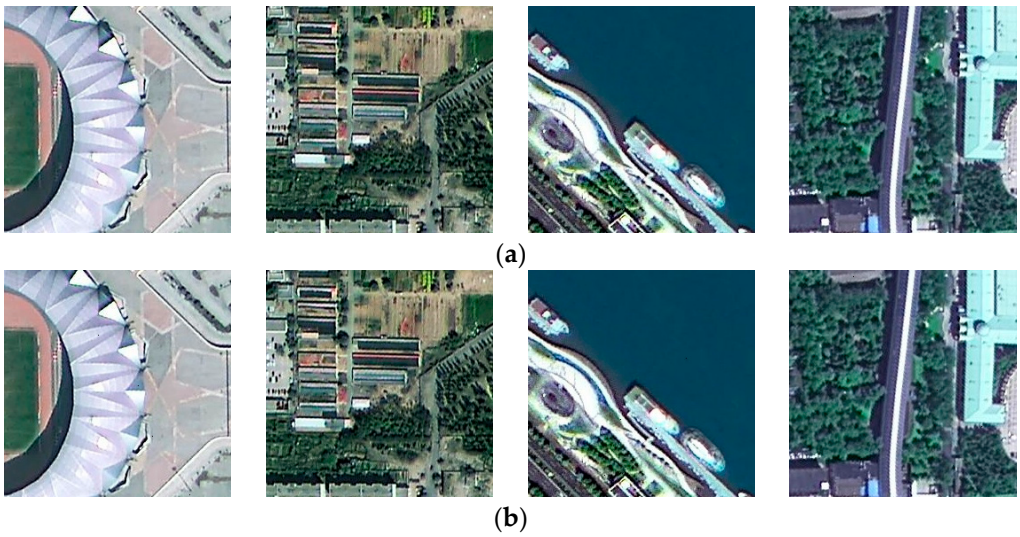


Figure 6. Comparison of images before and after random modification of a small number of pixels (4 pixels randomly selected and set to 0): (**a**) original HRRS images, (**b**) modified image.

The robustness test results for random modification of a small number of pixels are shown in Table 8. The robustness of our algorithm was basically the same as that of the Swin-Unet-based algorithm, and both of them were stronger than existing algorithms.

Table 8. Each algorithm's robustness against modification to small number of pixels (4 pixels).

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	42.05%	52.27%	71.09%	90.38%	98.26%
MultiResUnet	79.69%	88.45%	95.98%	98.68%	99.82%
U-net	51.99%	64.22%	90.61%	93.36%	98.57%
M-net	53.52%	63.39%	78.54%	92.77%	98.64%
Attention U-Net	52.03%	64.24%	80.58%	93.39%	98.56%
Attention ResU-Net	81.49%	86.61%	92.07%	96.43%	98.88%
Attention R2U-Net	19.54%	27.02%	41.27%	64.24%	86.12%
AAU-Net	90.91%	93.65%	97.38%	99.44%	100%
Swin-Unet	94.55%	97.09%	99.36%	100%	100%
Improved Swin-Unet (Our algorithm)	92.82%	97.25%	99.54%	100%	100%

A comprehensive analysis of our results (Tables 6–8) demonstrated that two Transformer-based subject-sensitive hash algorithms, that is, our algorithm and the Swin-Unet-based algorithm, greatly improved in robustness compared with existing subject-sensitive hash algorithms, especially in robustness against JPEG compression. Of course, our improved Swin-Unet-based algorithm as slightly inferior to the Swin-Unet-based algorithm because of the sacrifices that our algorithm had to make to enhance tampering sensitivity. In addition, the robustness of the Attention R2U-Net-based algorithm was the worst and would not be recommended for practical use.

4.5. Tampering Sensitivity Testing of Algorithms

If an algorithm's tampering sensitivity is too weak, the algorithm will have no useful value, even if the robustness of the algorithm is strong. Although tampering sensitivity and robustness are largely a pair of contradictory attributes, a good subject-sensitive hashing algorithm should try to balance both.

First, we tested each algorithm's sensitivity to subtle tampering with random locations. To simulate possible tampering in reality as much as possible, we performed position-random tampering with HRRS images in $Datasets_{10,000}$, with each tampering area measuring 8×8 pixels in size. A set of these tampering areas is shown in Figure 7.

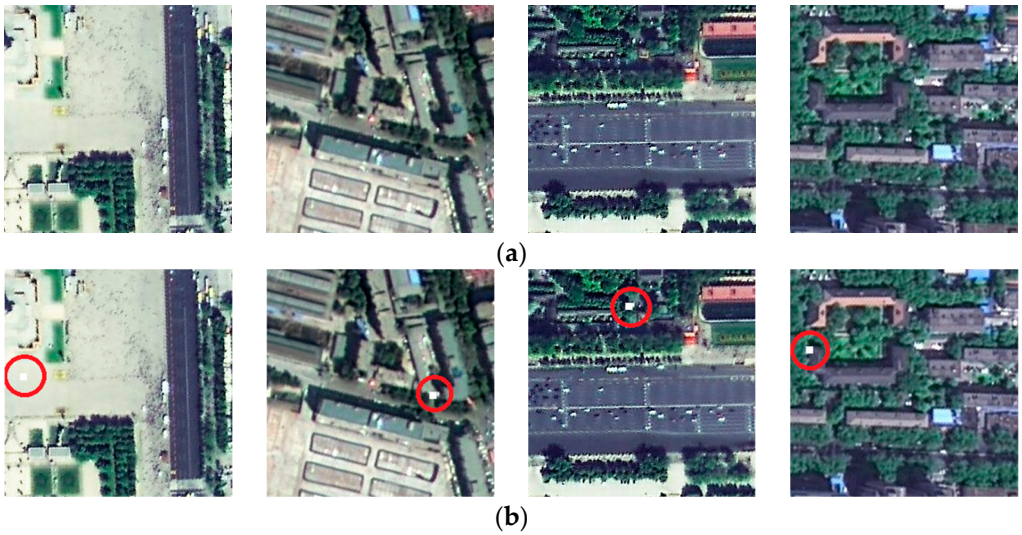


Figure 7. Comparison of images before and after tampering with a random area (8×8 pixels): (**a**) original HRRS images, (**b**) altered images.

The results of tampering sensitivity testing for random-position tampering areas 8×8 pixels in size are shown in Table 9. We used the proportion of instances in which tampering was detected to describe tampering sensitivity, as shown in Equation (4).

Table 9. Tampering sensitivity testing for random 8×8 pixel area tampering.

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	95.02%	90.98%	82.19%	56.24%	14.88%
MultiResUnet	83.95%	78.25%	62.52%	35.01%	11.89%
U-net	96.98%	94.94%	85.63%	61.39%	31.08%
M-net	96.27%	94.31%	89.15%	66.77%	33.16%
Attention U-Net	93.68%	91.29%	84.42%	68.61%	36.24%
Attention ResU-Net	67.94%	58.09%	40.66%	23.15%	7.79%
Attention R2U-Net	98.75%	98.07%	95.36%	80.93%	52.84%
AAU-Net	92.76%	90.92%	83.04%	65.57%	29.52%
Swin-Unet	84.61%	76.42%	56.01%	23.96%	6.92%
Improved Swin-Unet (Our algorithm)	94.29%	90.16%	81.73%	55.42%	19.24%

As shown in Table 9, Attention R2U-Net-based algorithms performed best, in terms of tampering sensitivity. However, the experiments in the previous section showed that its robustness was too poor to be recommended for practical use. The Attention ResU-Net-based algorithm had weaker tampering sensitivity. The tampering sensitivity of our algorithm and the algorithms based on MUM-Net, U-net, M-net, Attention U-Net, and AAU-Net were basically the same, all slightly inferior to the Attention R2U-Net-based

algorithm. Moreover, our algorithm was stronger than the Swin-Unet-based algorithm, in terms of tampering sensitivity.

To further test each algorithm's tampering sensitivity against random-position tampering, we increase the tampering area to 16 × 16 pixels. A set of these tampering areas is shown in Figure 8, in which we selected the same original HRRS images as in Figure 7 to facilitate direct comparison of tampering sensitivity with different granularities. The corresponding tampering sensitivity test results are shown in Table 10.

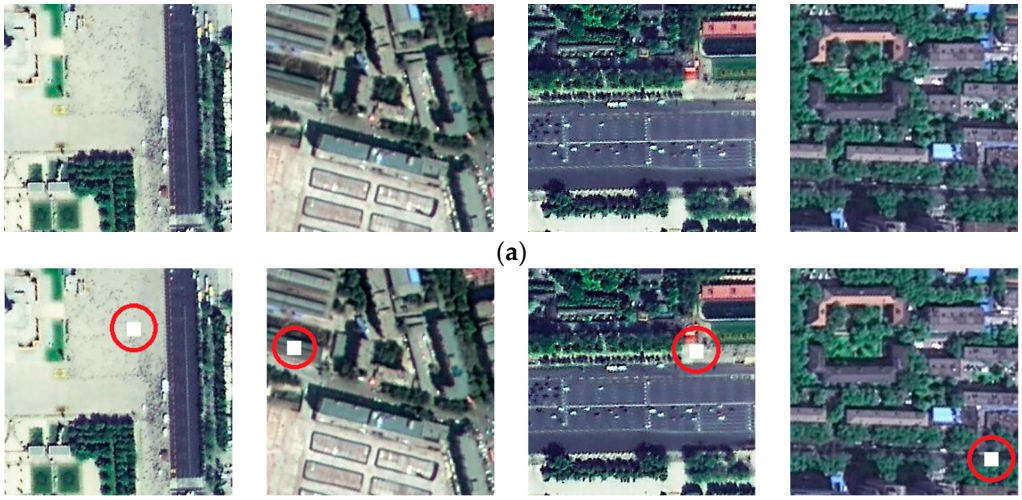


Figure 8. Comparison of images before and after being tampered with random area (16 × 16 pixel area): (**a**) Original HRRS images (The same as Figure 7a), (**b**) altered images.

Table 10. Tampering sensitivity test for random 16 × 16 pixel tampering.

The Model That Each Algorithm Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	98.79%	98.52%	95.25%	78.71%	42.77%
MultiResUnet	99.05%	97.73%	92.43%	79.26%	54.29%
U-net	98.85%	97.98%	96.24%	86.41%	59.17%
M-net	99.16%	99.03%	96.22%	86.98%	60.71%
Attention U-Net	98.86%	97.95%	96.16%	86.39%	59.21%
Attention ResU-Net	83.92%	76.68%	57.52%	34.74%	12.11%
Attention R2U-Net	99.92%	99.90%	99.85%	98.03%	81.54%
AAU-Net	99.54%	99.19%	96.43%	87.01%	64.24%
Swin-Unet	93.23%	89.29%	75.22%	32.61%	7.64%
Improved Swin-Unet (Our algorithm)	99.72%	99.41%	98.35%	88.62%	38.68%

As can be seen from Table 10, our algorithm's tampering sensitivity was second only to the Attention R2U-Net-based algorithm, superior to other existing algorithms, and superior to Swin-Unet-based algorithms.

Subject-sensitive hashing algorithms are more sensitive to tampering related to a particular subject. Since our algorithm chose buildings as the subject, the algorithms should have had a higher tampering sensitivity to building-related tampering. However, subject-sensitive hashing has not been proposed for a long time, and there was no available public dataset for testing subject-related tampering. To test subject-related tampering sensitivity, we distinguished subject-related tampering into adding buildings and deleting buildings and constructed two datasets for testing these two types of tampering. The two datasets each contained 200 sets of tampering instances, named $Data_{addbuildings}$ and $Data_{deletebuildings}$. Figures 9 and 10 show tampering examples of adding buildings and deleting buildings, respectively. The tampering sensitivity tests for adding buildings are shown in Table 11.

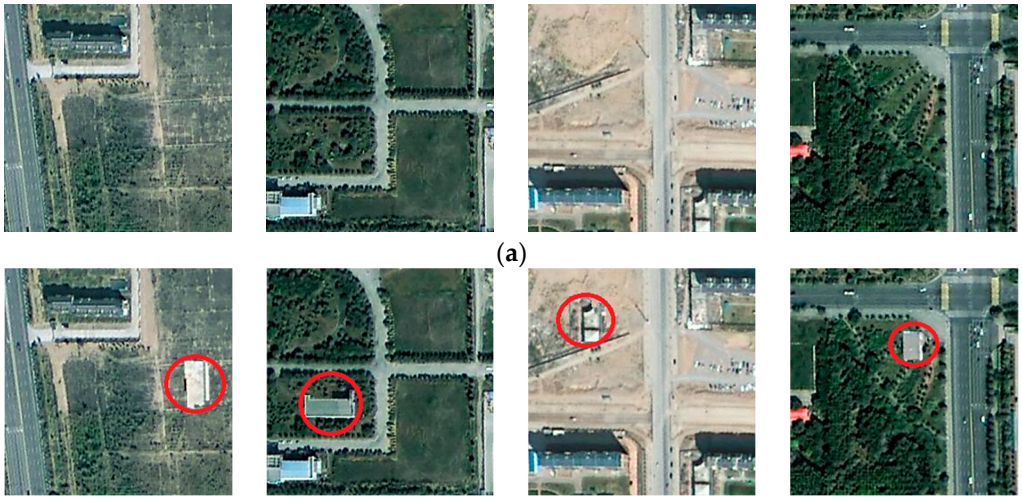


Figure 9. Examples of subject-related tampering (adding buildings): (**a**) original HRRS images, (**b**) altered images.

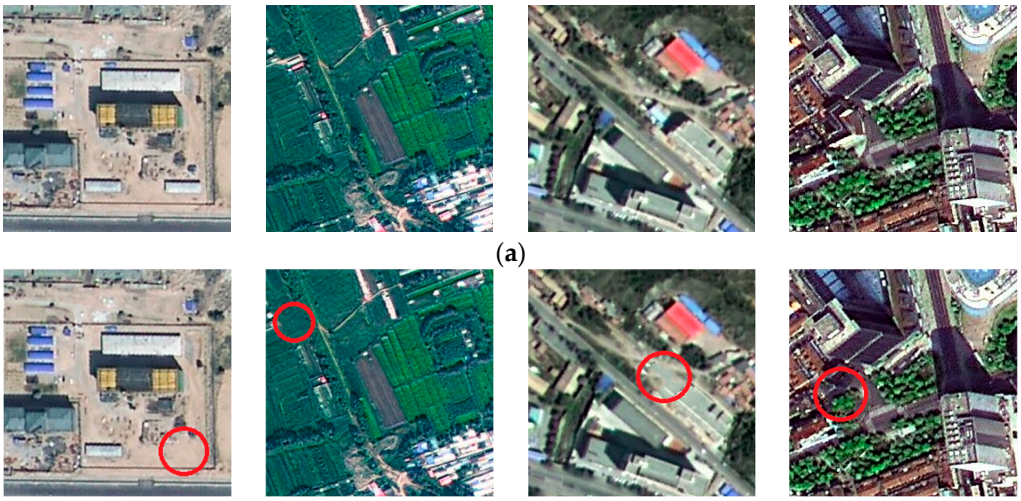


Figure 10. Examples of subject-related tampering (deleting buildings): (**a**) original HRRS images, (**b**) altered images.

Table 11. Tampering sensitivity tests for subject-related tampering (adding buildings).

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	100%	100%	99.0%	98.0%	60.5%
MultiResUnet	96.0%	94.5%	84.0%	56.0%	10.5%
U-net	100%	100%	99.5%	95.0%	65.0%
M-net	100%	100%	99.5%	94.0%	58.5%
Attention U-Net	100%	100%	98.0%	92.5%	50.0%
Attention ResU-Net	75.5%	67.0%	43.0%	19.5%	1.5%
Attention R2U-Net	100%	100%	100%	98.0%	77.0%
AAU-Net	100%	100%	98.0%	89.5%	36.0%
Swin-Unet	97.5%	90.0%	67.5%	20.0%	3.5%
Improved Swin-Unet (Our algorithm)	100%	100%	99.5%	96.0%	41.5%

As can be seen from Table 11, each algorithm demonstrated good sensitivity to tampering by adding buildings. Each algorithm's tampering sensitivity test for subject-related tampering (adding buildings) was ideal, and our algorithm was second only to the Attention R2U-Net-based algorithm at a low threshold (less than or equal to 0.05).

The tampering sensitivity tests for deleting buildings are shown in Table 12. Although the algorithms' sensitivity to tampering by deleting buildings did not significantly vary, our algorithm performed best at this test.

Table 12. Tampering sensitivity test for subject-related tampering (delete buildings).

Model Each Algorithm Was Based on	T = 0.02	T = 0.03	T = 0.05	T = 0.1	T = 0.2
MUM-Net	100%	100%	97.0%	80.5%	37.5%
MultiResUnet	92.0%	86.5%	63.5%	27.0%	8.0%
U-net	99.5%	98.0%	95.0%	76.0%	28.5%
M-net	100%	98.0%	94.5%	76.0%	27.0%
Attention U-Net	100%	97.0%	93.0%	78.5%	36.5%
Attention ResU-Net	90.0%	88.5%	76.0%	38.5%	11.0%
Attention R2U-Net	100%	100%	97.5%	65.0%	36.0%
AAU-Net	96.5%	94.0%	85.0%	56.5%	46.5%
Swin-Unet	95.0%	92.0%	82.5%	32.5%	2.5%
Improved Swin-Unet (Our algorithm)	100%	100%	98.0%	81.0%	43.0%

Comprehensively analyzing and summarizing the data shown Tables 9–12, we concluded that, among the compared algorithms, the Attention R2U-Net-based algorithm's tampering sensitivity was the best. However, its robustness was the worst, as stated in the conclusion of the previous section, meaning that the comprehensive performance of the algorithm was not good. Our algorithm's tampering sensitivity was second only to that of the Attention R2U-Net-based algorithm. Thus, our improved Swin-Unet compensated for the original Swin-Unet's lack of tampering sensitivity.

4.6. Computational Performance

Due to the influence of factors such as computing environment initialization and GPU startup, the computing performance of each algorithm with respect to different data amounts often differed. To test the computational performance of each algorithm under different computational amounts, we selected 300, 1000, and 10,000 HRRS images, respectively, from $Datasets_{10,000}$ to construct three datasets. Table 13 shows the computational performance of each algorithm with respect to these three datasets.

Table 13. Computational performance.

Model Each Algorithm Was Based on	300 Images		1000 Images		10,000 Images	
	Average Time (ms)	Total Time (s)	Average Time (ms)	Total Time (s)	Average Time (ms)	Total Time (s)
MUM-Net	35.13	10.54	24.20	24.20	23.43	234.30
MultiResUnet	44.07	13.22	33.89	33.89	32.37	323.68
U-net	21.20	6.36	13.34	13.34	12.78	127.79
M-net	27.57	8.27	17.79	17.79	15.75	157.50
Attention U-Net	21.47	6.44	15.60	15.60	13.38	133.82
Attention ResU-Net	46.83	14.05	26.60	26.60	22.12	221.24
Attention R2U-Net	30.40	9.12	20.84	20.84	19.24	192.36
AAU-Net	19.70	5.91	14.89	14.89	11.73	117.34
Swin-Unet	59.20	17.76	38.98	38.98	36.08	360.77
Improved Swin-Unet (Our algorithm)	49.73	14.92	33.11	33.11	31.24	312.37

The computational performance of our algorithm was only slightly better than the original Swin-Unet-based algorithm, and was essentially the same as the MultiResUnet-

based algorithm. In fact, Transformer has had a similar problem in other fields, such as Transformer-based image segmentation and image classification; that is, the computing performance of Transformer is lower than CNN.

5. Discussion

Finding methods to improve the robustness of content-preserving operations for HRRS images is one of the main problems faced by subject-sensitive hashing—especially robustness against JPEG compression. Transformer has achieved excellent results in many tasks and provided new research paths toward solving existing problems of subject-sensitive hashing. However, there have been no Transformer-based integrity authentication techniques, such as perceptual hashing and subject-sensitive hashing.

In this paper, we applied Transformer to subject-sensitive hash algorithms for HRRS images and proposed a new subject-sensitive hash algorithm based on our improved Swin-Unet. From the experiments, the following conclusions could be drawn:

1. Robustness

Transformer has demonstrated a significant advantage, in that it can improve the robustness of subject-sensitive hash algorithms. The robustness of the two Transformer-based algorithms in this paper—the Swin-Unet-based algorithm and our improved Swin-Unet-based algorithm—performed better than existing algorithms, especially in robustness against JPEG compression. Compared with the Swin-Unet-based algorithm, our algorithm was slightly less robust; this sacrifice had to be made to increase the algorithm's tampering sensitivity. After all, robustness and tampering sensitivity are essentially a pair of contradictory properties.

2. Tampering sensitivity

Compared to the algorithm based on the original Swin-Unet, our improved Swin-Unet-based algorithm achieved better tampering sensitivity at the expense of a slight decrease in robustness. Experiments showed that the tampering sensitivity of our improved Swin-Unet-based algorithm was second only to the Attention R2U-Net-based algorithm, outperforming other algorithms. The tampering sensitivity of the Attention R2U-Net-based algorithm was the best among the algorithms compared herein, but its robustness was the worst. As such, we could not recommend it for actual application.

3. Security, and Digestibility Analysis

The security of our algorithm was based on several aspects; the first of these was the difficulty of Transformer interpretability [49]. In fact, the interpretability of Transformers, in addition the interpretability of deep neural network models [50,51], has always been a difficult problem for the academic community. However, the difficulty of Transformer interpretability can guarantee the unidirectionality of subject-sensitive hashing; that is to say, Transformer interpretability ensures that valid information from the original HRRS image is hard to get from the hash sequence. The second aspect was the encryption algorithm used in the encoding process. The security of the AES algorithm has been widely recognized; thus, it ensured the security of sensitive hash sequences.

Third, the mapping mechanism used in the coding process made nonlinear modifications to the values of image features, which further increased the difficulty of obtaining the original HRRS image features from the hash sequence and enhanced the security of the algorithm.

Regarding digestibility, since the main difference of each comparison algorithm lay in the deep neural network model and the feature encoding processes were the same, the digestibility values of each algorithm were the same.

4. Computational performance

Compared to CNN, Transformer has higher computational complexity. Other application areas of Transformer have faced the problems of long inference time and training time. In this experiment, the two Transformer-based algorithms, namely our algorithm and

the original Swin-Unet-based algorithm, also had the disadvantage of low computational performance. In fact, the computational performances of these two algorithms were not only inferior to algorithms based on CNN, such as U-net and M-net, they were also inferior to algorithms based on attention mechanisms, such as AAU-net and Attention U-net.

In summary, considering robustness, tampering sensitivity, security, and summarization, our algorithm, based on an improved Swin-Unet, gave the best comprehensive performance among the algorithms in this comparison. In future research, we will strive to focus on solving the problems uncovered in this work—namely, giving the algorithm better robustness and tampering sensitivity while improving computational efficiency.

6. Conclusions

In this paper, we proposed a Transformer-based subject-sensitive hash algorithm for HRRS images. The algorithm extracted the features of HRRS images based on the improved Swin-Unet we constructed, and generated the hash sequence through a feature coding method that combined the mapping mechanism with PCA. Experiments showed that the robustness of our improved Swin-Unet-based algorithm was greatly improved, compared with existing algorithms; for example, the robustness against JPEG compression was significantly improved. Our algorithm addressed the original Swin-Unet's lack of tampering sensitivity, especially to subject-related tampering. The overall performance of our proposed algorithm was better than existing algorithms

However, our algorithm, like other Transformer-based applications, suffered from high model complexity and low computational performance. The model occupied a large amount of storage space and took more time to calculate the hash sequence of an HRRS image than the existing algorithm. In future research, we will focus on improving the computational performance of the algorithm.

Author Contributions: K.D. designed the algorithm under the guidance of Professor S.C.; Y.Z. and X.Y. assisted with the experiments; Y.W. participated in experimental data collation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grants from (a) NSFC (Grant Nos. 42101428, 41801303) and (b) the Research Foundation of Jinling Institute of Technology (Grant Nos. jit-b-201913, jit-fhxm-201905).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: WHU building dataset are available at https://study.rsgis.whu.edu.cn/pages/download/building_dataset.html (accessed on 15 December 2022).

Acknowledgments: Tingting Jiang of Ericsson (China) Company contributed to the compilation of experimental data in this paper, and the authors would like to thank her.

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

References

1. Niu, X.; Jiao, Y. An Overview of Perceptual Hashing. *Acta Electron. Sin.* **2008**, *36*, 1405–1411.
2. Khelifi, F.; Jiang, J. Analysis of the Security of Perceptual Image Hashing Based on Non-Negative Matrix Factorization. *IEEE Signal Process. Lett.* **2010**, *17*, 43–46. [CrossRef]
3. Du, L.; Ho, A.; Cong, R. Perceptual hashing for image authentication: A survey. *Image Commun.* **2020**, *81*, 115713. [CrossRef]
4. Samanta, P.; Jain, S. Analysis of Perceptual Hashing Algorithms in Image Manipulation Detection. *Procedia Comput. Sci.* **2021**, *185*, 203–212. [CrossRef]
5. Li, X.; Qin, C.; Wang, Z.; Qian, Z.; Zhang, X. Unified Performance Evaluation Method for Perceptual Image Hashing. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1404–1419. [CrossRef]
6. 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]
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.

8. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv* **2021**, arXiv:2010.11929.
9. Zhang, Q.; Li, Y.; Hu, Y.; Zhao, X. An Encrypted Speech Retrieval Method Based on Deep Perceptual Hashing and CNN-BiLSTM. *IEEE Access* **2020**, *8*, 148556–148569. [CrossRef]
10. Khelifi, F.; Bouridane, A. Perceptual Video Hashing for Content Identification and Authentication. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 50–67. [CrossRef]
11. Rajeshwaran, K.; Anil Kumar, K. Cellular Automata Based Hashing Algorithm (CABHA) for Strong Cryptographic Hash Function. In Proceedings of the IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, Tamil Nadu, India, 20–22 February 2019; pp. 1–6.
12. Gupta, D.N.; Kumar, R. Sponge based Lightweight Cryptographic Hash Functions for IoT Applications. In Proceedings of the International Conference on Intelligent Technologies (CONIT), Hubbali, Karnataka, India, 25–27 June 2021; pp. 1–5.
13. Qin, C.; Sun, M.; Chang, C. Perceptual hashing for color images based on hybrid extraction of structural features. *Signal Process.* **2018**, *36*, 194–205. [CrossRef]
14. Tang, Z.; Li, X.; Zhang, X.; Zhang, S.; Dai, Y. Image hashing with color vector angle. *Neurocomputing* **2018**, *308*, 147–158. [CrossRef]
15. Hamid, H.; Ahmed, F.; Ahmad, J. Robust Image Hashing Scheme using Laplacian Pyramids. *Comput. Electr. Eng.* **2020**, *84*, 106648. [CrossRef]
16. Biswas, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A new perceptual hashing method for verification and identity classification of occluded faces. *Image Vis. Comput.* **2021**, *113*, 104245. [CrossRef]
17. Wang, X.; Zhang, Q.; Jiang, C.; Xue, J. Perceptual hash-based coarse-to-fine grained image tampering forensics method. *J. Vis. Commun. Image Represent.* **2021**, *8*, 103124. [CrossRef]
18. Huang, Z.; Liu, S. Perceptual Image Hashing With Texture and Invariant Vector Distance for Copy Detection. *IEEE Trans. Multimed.* **2021**, *23*, 1516–1529. [CrossRef]
19. Romero, A.; Gatta, C.; Camps-Valls, G. Unsupervised Deep Feature Extraction for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1349–1362. [CrossRef]
20. Quan, D.; Wei, H.; Wang, S.; Lei, R.; Duan, B.; Li, Y.; Hou, B.; Jiao, L. Self-Distillation Feature Learning Network for Optical and SAR Image Registration. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4706718. [CrossRef]
21. Chen, H.; Qi, Z.; Shi, Z. Remote Sensing Image Change Detection with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5607514. [CrossRef]
22. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
23. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
24. Adiga, V.; Sivaswamy, J. FPD-M-net: Fingerprint Image Denoising and Inpainting Using M-Net Based Convolutional Neural Networks. *arXiv* **2019**, arXiv:1812.10191.
25. Ibtehaz, N.; Rahman, M. MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Net.* **2020**, *121*, 74–87. [CrossRef] [PubMed]
26. 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.* **2022**, *13*, 5109. [CrossRef]
27. Xu, Y.; Xu, W.; Cheung, D.; Tu, Z. Line Segment Detection Using Transformers without Edges. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 21–24 June 2021; pp. 4255–4264.
28. Lin, A.; Chen, B.; Xu, J.; Zhang, Z.; Lu, G.; Zhang, D. DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 4005615. [CrossRef]
29. Wang, W.; Tang, C.; Wang, X.; Zheng, B. A ViT-Based Multiscale Feature Fusion Approach for Remote Sensing Image Segmentation. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4510305. [CrossRef]
30. Song, R.; Feng, Y.; Cheng, W.; Mu, Z.; Wang, X. BS2T: Bottleneck Spatial–Spectral Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5532117. [CrossRef]
31. Xue, X.; Zhang, H.; Fang, B.; Bai, Z.; Li, Y. Grafting Transformer on Automatically Designed Convolutional Neural Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5531116. [CrossRef]
32. Zhang, W.; Zhou, M.; Ji, C.; Sui, X.; Bai, J. Cross-Frame Transformer-Based Spatio-Temporal Video Super-Resolution. *IEEE Trans. Broadcast.* **2022**, *68*, 359–369. [CrossRef]
33. Lei, S.; Shi, Z.; Mo, W. Transformer-Based Multistage Enhancement for Remote Sensing Image Super-Resolution. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5615611. [CrossRef]
34. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 21–24 June 2021; pp. 1833–1844.
35. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2844–2853.

36. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An Efficient Network for Low-Altitude Object Detection Based on Convolution and Transformer. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2507412. [CrossRef]
37. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**. [CrossRef]
38. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision Transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 10012–10022.
39. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *arXiv* **2021**, arXiv:2105.05537.
40. Zhang, X.; Parhi, K. High-speed VLSI architectures for the AES algorithm. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2004**, *12*, 957–967. [CrossRef]
41. Masoumi, M.; Rezayati, M.H. Novel Approach to Protect Advanced Encryption Standard Algorithm Implementation Against Differential Electromagnetic and Power Analysis. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 256–265. [CrossRef]
42. Ding, K.; Zhu, C.; Lu, F. An adaptive grid partition based perceptual hash algorithm for remote sensing image authentication. *Wuhan Daxue Xuebao* **2015**, *40*, 716–720.
43. Kingma, D.P.; Ba, J. ADAM: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
44. Alom, M.Z.; Yakopcic, C.; Hasan, M.; Taha, T.M.; Asari, V.K. Recurrent residual U-Net for medical image segmentation. *J. Med. Imaging.* **2019**, *6*, 014006. [CrossRef]
45. Zhao, S.; Liu, T.; Liu, B.W.; Ruan, K. Attention residual convolution neural network based on U-net (AttentionResU-Net) for retina vessel segmentation. *IOP Conf. Ser. Earth Environ. Sci. IOP Publ.* **2020**, *440*, 032138. [CrossRef]
46. Li, R.; Zheng, S.; Duan, C.; Su, J.; Zhang, C. Multistage Attention ResU-Net for Semantic Segmentation of Fine-Resolution Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 8009205. [CrossRef]
47. Ji, S.P.; Wei, S.Y. Building extraction via convolutional neural networks from an open remote sensing building dataset. *Acta Geod. Cartogr. Sin.* **2019**, *48*, 448–459.
48. Xia, G.; 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.
49. Chefer, H.; Gur, S.; Wolf, L. Transformer Interpretability Beyond Attention Visualization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Online, 21–24 June 2021; pp. 782–791.
50. Wang, S.; Yin, Y.; Wang, D.; Wang, Y.; Jin, Y. Interpretability-Based Multimodal Convolutional Neural Networks for Skin Lesion Diagnosis. *IEEE Trans. Cybern.* **2022**, *52*, 12623–12637. [CrossRef]
51. Chi, S.; Tian, Y.; Wang, F.; Wang, Y.; Chen, M.; Li, J. Deep Semisupervised Multitask Learning Model and Its Interpretability for Survival Analysis. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 3185–3196. [CrossRef] [PubMed]