SDTU-Net: Stepwise-Drop and Transformer-Based U-Net for Subject-Sensitive Hashing of HRRS Images

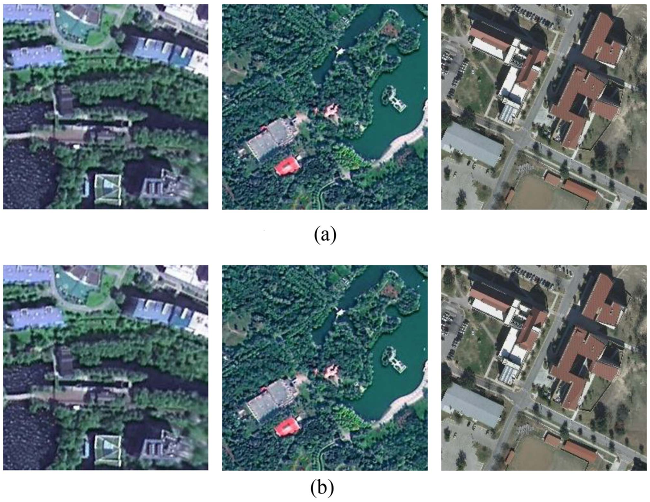
Kaimeng Ding, Shiping Chen, *Senior Member, IEEE*, Yue Zeng, Yanan Liu, Bei Xu, and Yingying Wang

***Abstract*—As a new integrity authentication technology, subject-sensitive hashing has the ability to achieve subject-sensitive authentication for high-resolution remote sensing (HRRS) images and can provide a security guarantee for their subsequent use. However, existing research on subject-sensitive hashing focuses on improving the structure of the deep neural network of the algorithm to improve the algorithm's performance, which makes it necessary to reconstruct the training dataset or modify the network structure in the face of different integrity authentication requirements. In this article, we delve into the impact of dropout on subject-sensitive hashing and propose a stepwise-drop mechanism to address the robustness and tampering-sensitivity requirements of subject-sensitive hashing. On this basis, a network named stepwise-drop and transformer-based U-net (SDTU-net) is proposed for subject-sensitive hashing of HRRS images. SDTU-net can use our proposed stepwise-drop mechanism to determine the drop rate of different network layers, which makes it possible to adjust the algorithm performance without changing network structure and training data. Experiments show that our SDTU-net based subject-sensitive hashing has better overall performance compared with existing algorithms, especially at medium and low thresholds. Our approach solves the problem that the existing algorithms cannot balance robustness and tamper sensitivity at low thresholds.**

***Index Terms*—Dropout, geodata security, high-resolution remote sensing (HRRS) images, integrity authentication, subject-sensitive hashing, transformer, U-net.**

I. Introduction

HIGH-RESOLUTION remote sensing (HRRS) images with rich information, high precision, fast information transmission, and all-weather work [1] are widely used in the assessment of the environment [2], disaster monitoring [3], and other applications [4]. However, the security of the HRRS image must be guaranteed before it can be used. If an HRRS image is maliciously tampered with, the information carried by the HRRS image will be distorted, and the analysis or prediction results based on the HRRS image cannot be trusted. Additionally, if the tampered content is primarily used by the user, user's conclusions based on this tampered HRRS image are wrong, which will have a serious impact on decision-making. This means that HRRS images can only be used well when their content integrity is guaranteed, especially the subject-sensitive content that users focus on.


(a)

(b)

Fig. 1. Comparison of content after HRRS image format conversion. (a) Original high-resolution remote sensing images (TIF format). (b) Format converted images (BMP format).

Mainstream data security technologies, such as fragile watermarking [5], cryptographic algorithms [6], and block chain [7], are still insufficient in the integrity authentication of HRRS images. The most prominent performance is that mainstream security technologies implement authentication at the binary level. This makes it impossible for them to determine whether the carrier of the data has changed or the content of the data has been tampered with. A set of operations that do not alter the content of the HRRS image are shown in Fig. 1, original images in TIF format are shown in

Manuscript received 17 September 2023; revised 11 December 2023; accepted 17 January 2024. Date of publication 22 January 2024; date of current version 1 February 2024. This work was supported in part by the NSFC under Grant 42101428 and Grant 41801303 and in part by Natural Science Fund for Colleges and Universities in Jiangsu Province under Grant 20KJD170002. *(Corresponding author: Yingying Wang.)*

Kaimeng Ding is with the Jinling Institute of Technology, Nanjing 211169, China, and also with the Jiangsu Artificial Intelligence Transportation Innovations & Applications Engineering Research Center, Nanjing 211169, China (e-mail: dkm@jit.edu).

Shiping Chen is with CSIRO Data61, Sydney, NSW 1710, Australia (e-mail: shiping.chen@data61.csiro.au).

Yue Zeng, Yanan Liu, Bei Xu, and Yingying Wang are with the Jinling Institute of Technology, Nanjing 211169, China (e-mail: zengy@jit.edu.cn; yanan.liu@jit.edu.cn; xubei@jit.edu.cn; wyy@jit.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3356660

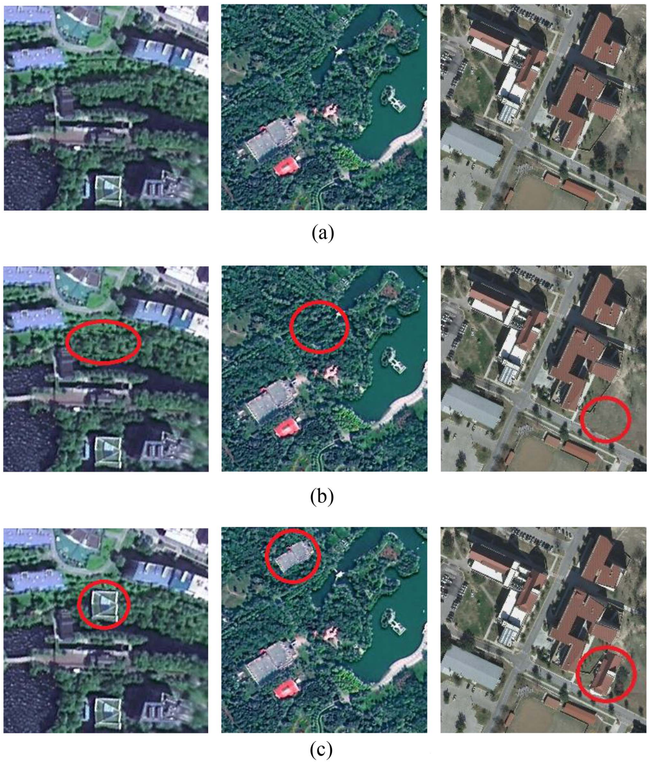


Fig. 2. Examples of subject-unrelated tampering and subject-related tampering. (a) Original high-resolution remote sensing images. (b) Tampered images (subject-unrelated tampering). (c) Tampered images (subject-related tampering).

Fig. 1(a), and the images converted to BMP format are shown in Fig. 1(b).

In Fig. 1, the remote sensing information carried by the images has not changed after converting the images to BMP format, it is the carrier of information that has been changed. In this case, mainstream security technologies, such as cryptography and blockchain, believe that the data have changed, which is inappropriate.

Although perceptual hashing [8], [9], [10] can realize content-based integrity authentication for HRRS images, it cannot carry out stricter integrity authentication for the content that users are interested in, and does not have subject sensitivity. A set of tampered examples of HRRS images are shown in Fig. 2, where Fig. 2(a) is the original HRRS images and Fig. 2(b) and (c) shows tampered images. If the tampered areas are not pointed out, it is difficult to determine whether and where these images have been maliciously tampered. Moreover, the tampering shown in Fig. 1(c) is more harmful and makes it easier for users to make incorrect analysis results and decisions, especially for users who take building information as their primary object. Here, the tampering content in Fig. 2(b) and (c) is subject-unrelated and subject-related tampering, respectively.

Subject-sensitive hashing, derived from perceptual hashing, overcomes the above-mentioned shortcomings [11], [12] and can realize subject-sensitive authentication. It can perform stricter authentication on the main ground object information used by users. However, there are still some shortcomings in existing subject-sensitive hashing, which are mentioned as follows.

1) Under the premise of maintaining tampering sensitivity, the robustness of subject-sensitive hashing is not ideal, especially for JPEG compression.
2) As existing subject-sensitive hashing relies too much on the deep learning models and training sample set, subject-sensitive hashing lacks the means to adjust the robustness and tampering sensitivity of the algorithm in the case of determining the training dataset and network structure.
3) Existing deep learning network models for subject-sensitive feature extraction adopt the same dropout rate to extract the shallow and deep features of the network equally, which is detrimental to extracting robust features.

Based on the study of the influence of dropout on subject-sensitive hashing, we propose a stepwise-drop and transformer-based U-net (SDTU-net) to implement subject-sensitive hashing for the authentication of HRRS images. The main contributions can be summarized in the following points.

1) We take a closer look at the impact of the dropout mechanism on the performance of subject-sensitive hashing and propose a stepwise-drop mechanism for neural networks to improve the robustness of subject-sensitive hashing
2) A new deep neural network named SDTU-net is proposed based on a stepwise-drop mechanism, which is more suitable for extracting subject-sensitive features.
3) SDTU-net based subject-sensitive hash algorithm is proposed, which can adjust the robustness of the algorithm without changing the training dataset and network structure.

This article is then organized as follows. The theory of subject-sensitive hashing is briefly discussed in Section II. Section III discusses our proposed stepwise-drop and STDU-net based algorithm in detail. Section IV demonstrates the experimental setup and experimental results. A discussion is presented in Section V. Section VI presents future works and conclusions of this article.

II. Related Work

A. Perceptual Hashing and Subject-Sensitive Hashing

Perceptual hashing can map images with the same content into the same digest as a string. It is also known as image hashing or perceptual hash algorithm. Compared with the cryptographic hash (such as MD5 and SHA1), digital signature, and blockchain, perceptual hashing can map images with the same content into the same hash sequences. It can be used to realize image authentication [12], image retrieval [13], and image copy detection [14].

Some scholars have studied perceptual hashing for remote sensing images. Li et al. [15] proposed a hashing network based on deep learning for remote sensing image retrieval, which is not suitable for the integrity authentication of remote sensing images. Ding et al. [16] proposed an improved U-net for the perceptual hash algorithm of HRRS images, improving the algorithm's robustness. Zhang et al. [17] proposed a perceptual hash algorithm combining local and global features of images, which can locate tampering.

With the improvement of the resolution of HRRS image, perceptual hashing is gradually unable to cope with some problems of HRRS image authentication, the more prominent problem is that perceptual hashing cannot distinguish the key content in HRRS images. For example, if building information is mainly used by the users, the building information in HRRS images should be more strictly authenticated, and the authentication method should pay more attention to whether building information in HRRS images has changed. This type of integrity authentication, a specific type of feature, that is focused on is known as subject-sensitive authentication. When objects and backgrounds are complex, the insufficient feature difference leads to the fact that perceptual hashing cannot maintain sufficient sensitivity to a specific subject.

Subject-sensitive hashing, derived from perceptual hashing, is also named subject-sensitive hash algorithm. Subject sensitivity is the main difference between perceptual hashing and subject-sensitive hashing [11]. Subject sensitivity makes subject-sensitive hashing meet the integrity authentication needs of users in different domains that are sensitive to a particular subject and can achieve subject-related integrity authentication with as few perceptual features as possible, avoiding too large hash sequence. Although it is not easy to implement subject-sensitive hashing with traditional methods, the rise of deep learning provides a feasible way to implement subject-sensitive hashing.

Deep learning has strong feature extraction capabilities [18] and has been deeply studied in the field of remote sensing to overcome problems that are difficult to solve by traditional methods. In [19], a cloud detection method based on deep semisupervised learning and active learning is proposed to achieve state-of-the-art segmentation with a small number of labels. In [20], a multistage self-guided separation network is presented for the classification of remote sensing images, solving the problem of unbalanced change of background and target between interclass samples. Zhang et al. [21] proposed a structural optimization transmission network for land-cover classification of HSI and LiDAR data to enhance the complementary ability of multiple sources in collaborative classification tasks.

The deep learning model for feature extraction is the key to a subject-sensitive hash algorithm. In [11], MUM-Net is used for subject-sensitive feature extraction. In [12], AAU-Net is used to implement critical feature extraction tasks. In fact, other deep learning networks, such as U-net [22], M-net [23], Attention U-net [24], MultiResUNet [25], and Attention ResU-Net [26], can be used for subject-sensitive feature extraction.

B. Dropout

The difficulty of deep learning based subject-sensitive hash algorithm lies in how to balance the robustness of the algorithm with tampering sensitivity [12], and dropout technology can improve this difficulty to some extent.

Dropout [27] is a generalization technique for deep neural networks. Dropout randomly selects a subset of the inputs from each training iteration to prevent the trained network from overfitting. Several variants of dropout have been proposed for various learning tasks. Ko et al. [28] proposed controlled dropout to facilitate the reduction of training time and memory usage. Inoue et al. [29] presented multisample dropout, which is an enhanced dropout technique. Different from the original dropout randomly discards the neurons, multisample dropout creates multiple dropout samples. Liu et al. [30] developed β-dropout that unifies discrete dropout with continuous dropout, which can derive approximate Gaussian dropout and Bernoulli dropout. To solve the problem that the transformer is prone to overfitting with an insufficient amount of training data, Li et al. [31] developed DropKey to improve the dropout technique in ViT (Vision Transformer). DropKey implicitly assigns an adaptive operator to each attention block by randomly dropping part of the key to constrain the attention distribution. Lu et al. [32] proposed a regularization algorithm named MultiDrop, which drops some random tasks in optimization.

Different from other applications of deep learning such as object detection and instance segmentation, subject-sensitive hashing is not the richer the extracted features, the better: if too much information is extracted, the subject-sensitive hash algorithm will have to process too many subject-unrelated features when generating subject-sensitive hash sequences, which is not conducive to algorithm's robustness; if image feature (especially subject-related feature) extraction is insufficient, tampering sensitivity of the algorithm will be reduced.

Inspired by DropKey [31], we propose a new drop mechanism named stepwise drop to overcome the above-mentioned problems. Stepwise drop enables deep neural networks to adopt different drop rates at different network layers to reduce the impact of shallow features on the algorithm's robustness while extracting more deep features to enhance tampering sensitivity.

C. Transformers

Transformer has been successfully applied to application research of remote sensing images, providing a new idea to solve the problems of insufficient robustness faced by subject-sensitive hashing. Chen et al. [33] employed a pure multiscale transformer for captioning of remote sensing images, which can effectively generate specific types of captions. Zhang et al. [34] built a dual stream network (DTHNet) based on a transformer for shadow extraction of remote sensing images. Drawing on the principle and structure of TransUNet [35], Wu et al. [36] proposed a multilevel TransUNet (MTU-net) for multilevel feature extraction of remote sensing images. He et al. [37] proposed Swin Transformer embedding U-net (ST-Unet) for remote sensing image semantic segmentation.

Taking advantage of the transformer in feature extraction, combined with our proposed stepwise-drop mechanism, we design a novel stepwise-drop and transformer-based U-net (SDTU-net) to achieve subject-sensitive feature extraction

III. Method

In this section, we introduce the proposed stepwise-drop mechanism and SDTU-net based subject-sensitive hash algorithm in detail. First, our proposed stepwise-drop mechanism is introduced. The network structure of SDTU-net is described subsequently. Finally, the SDTU-net based subject-sensitive hash algorithm is introduced.

A. Stepwise Drop

Stepwise drop is a mechanism for determining the drop rate of each layer in an encoder–decoder network, making different layers of the network adopt different dropout rates. The stepwise-drop mechanism is stated as follows.

1) For the encoder stage, set two initial values TopKey and MinusKey, which, respectively, represent the initial drop rate at the beginning of the encoder stage and the decrement value. For the decoder stage, two initial values, BotKey and PlusKey, are set to represent the initial value of the drop rate of the decoder stage and the value of each increase.
2) Let EnNum represents the block number of the encoder, then

$$\text{droprate}_{\text{Encoder}}^{\text{EnNum}} = \text{TopKey} - (\text{EnNum}-1) \times \text{MinusKey}. \tag{1}$$

The drop rate for each block in the encoder stage can be expressed as follows:

$$\text{encoder\_drop}^{\text{EnNum}} = \begin{cases} \text{droprate}_{\text{Encoder}}^{\text{EnNum}} \\ 0, \text{ if } \text{droprate}_{\text{Encoder}}^{\text{EnNum}} < 0 \end{cases}. \tag{2}$$

3) Let DeNum represents the block number of the decoder, then

$$\text{droprate}_{\text{Decoder}}^{\text{DeNum}} = \text{BotKey} - (\text{DeNum} - 1) \times \text{PlusKey}. \tag{3}$$

The drop rate for each block of the decoder stage can be expressed as follows:

$$\text{decoder\_drop}_{\text{Decoder}}^{\text{DeNum}} = \begin{cases} \text{droprate}_{\text{Decoder}}^{\text{DeNum}} \\ 0, \text{ if } \text{droprate}_{\text{Decoder}}^{\text{DeNum}} < 0 \end{cases}. \tag{4}$$

It can be seen from (1)–(4) that the output drop rate of stepwise drop decreases gradually in the encoder stage, whereas the drop rate increases gradually in the decoder stage, and the minimum drop rate of both the encoder and decoder is 0.

In fact, DropKey [31] also sets a different drop rate for each network layer, but it is quite different from our stepwise-drop mechanisim.

1) Different from DropKey, which sets Key as a drop object, stepwise-drop sets the drop rate of each module by setting the initial drop rate, and value of each increase and decrease to adjust the algorithm's robustness and tampering sensitivity.
2) The drop rate of DropKey is decreasing layer by layer, whereas the drop rate of our stepwise drop is increasing in the decoder stage and decreases in the encoder stage.
3) Our stepwise drop produces a series of discrete drop rates, mainly for the convolutional neural network (CNN), whereas DropKey is for transformers.

B. Architecture of SDTU-Net

The detailed structure of our proposed SDTU-net is shown in Fig. 3. The overall structure of SDTU-net is similar to that of U-net [22] and TransUnet [35]. SDTU-net takes the preprocessed HRRS image as input and consists of an encoder part, a transformer block, and a decoder part.

1) Encoder: The encoder part of SDTU-net is divided into four modules, each module is similar to the original U-net's encoder, consisting of convolutional layers, a pooling layer (the last module does not contain a pooling layer), batch normalization (BN) layers, and dropout layers. The main difference from U-net's encoder is that the drop rate of the encoder of STDU-net is determined by stepwise drop mechanism, whereas the drop rate of U-net's encoder is constant.

2) Decoder: The decoder part of SDTU-net is divided into three modules, each consisting of two convolutional layers, one upsampling layer, one concatenation layer, and one dropout layer. As with STDU-net's encoder, the decoder's drop rate is also determined by a stepwise drop mechanism.

3) Transformer Block: SDTU-net's transformer block differs from TransUnet in that it has only 4 transformer layers instead of 12. This is mainly because the transformer is computationally intensive, and too many transformer layers do not significantly improve the performance of the subject-sensitive hash algorithm. A more detailed analysis will be carried out in Section V-C.

Stepwise drop enables SDTU-net to extract HRRS image features with stronger subject sensitivity. Shallow networks of deep neural networks extract low-dimensional visual features, whereas deep networks are more suitable for extracting crude but complex essential information. If a shallow network extracts too many features, the algorithm's robustness would be affected. If there is no stepwise drop, the structure of SDTU-net is similar to TransUnet [35] (except for the number of transformer layers). TransUnet has good feature extraction capabilities, but it is not suitable for direct use in implementing the subject-sensitive hash algorithm due to its poor robustness, which will be demonstrated by the experiments in this article.

C. Overview of the SDTU-Net Based Subject-Sensitive Hash Algorithm

As shown in Fig. 3, the main steps of the SDTU-net based subject-sensitive hash algorithm include preprocessing of the HRRS image, SDTU-net-based feature extraction, feature compression, and encoding.

In preprocessing, the HRRS image is resized to 256 × 256 pixels to meet the input requirements of SDTU-net. In the feature compression stage, principal component analysis is used to decompose the feature matrix extracted by SDTU-net, and the first column of principal components after matrix decomposition is extracted as the subject-sensitive feature, which is binary encoded by the low-bit priority principle to generate a 128-b hash sequence.

The hash sequence of the original HRRS image is transmitted and stored with the corresponding HRRS image. If an HRRS image needs to be verified whether it has been tampered with, the

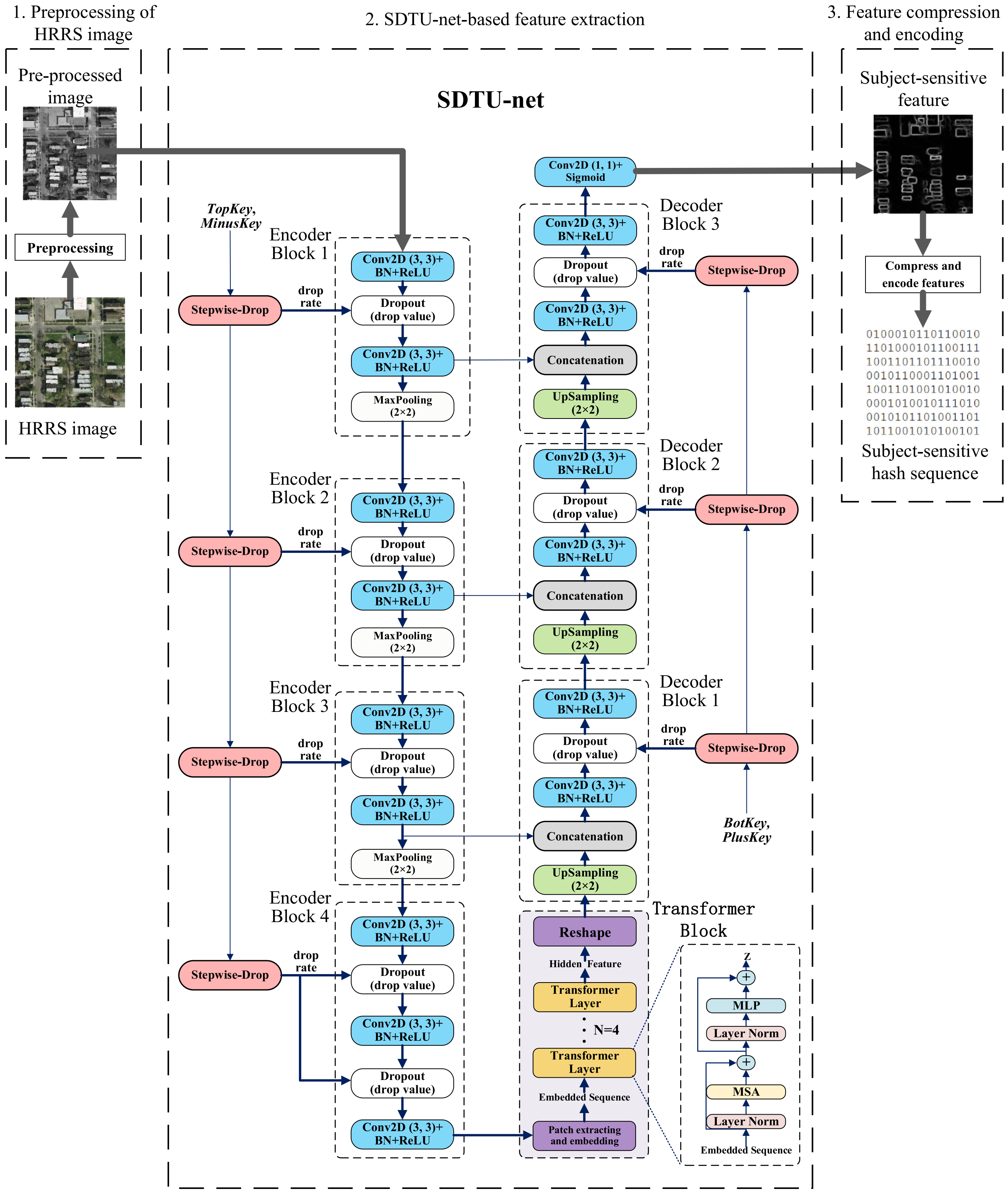


Fig. 3. Subject-sensitive hash algorithm based on SDTU-net.

hash sequence of this HRRS image is generated and compared with the hash sequence of the original image. Our algorithm also uses normalized Hamming distance (*N-Dis*) to compare two hash sequences: if *N-Dis* is greater than the pre-set threshold *T*, this HRRS image has been tampered with.

N-Dis between two hash sequences is shown as follows:

$$N - Dis = \left(\sum_{i=1}^{128} \left| SH\left(i\right) - SH'\left(i\right) \right| \right) /128 \qquad (5)$$

where *SH* and *SH*' represent hash sequences of the HRRS image to be authenticated and the original HRRS image, respectively.

IV. Experiments and Analysis

In this section, details of model training and experimental data are detailed first. Then, a set of integrity authentication examples is shown to initially compare each algorithm. Next, we focus on testing the tampering sensitivity and robustness of each comparison algorithm.

A. Datasets and Implementation Details

1) Datasets: The experimental data in this article includes two categories: datasets used to train deep neural networks, and datasets used to test the subject-sensitive hash algorithm. SDTU-net is trained using the same training dataset in [11] and [12], which is a dataset based on WHU building dataset [38] and contains 3166 pairs of training samples. Datasets testing the performance of subject-sensitive hashing algorithms will be presented in Section IV-C.

2) Training Settings: Our SDTU-net is implemented based on Keras 2.9.0 (with TensorFlow as the backend). In the training process of SDTU-net, the number of epochs is set to 100, and batch size is 8; the Adam optimizer is used with an initial learning rate of 0.0001. Since the proportion of pixels occupied by edge features of the object is generally low in the training sample, we use the binary focal loss as the loss function and set the parameters alpha and gamma to 0.25 and 2, respectively. Gelu is used as the activation function of multilayer perceptron in the transformer block, Sigmoid is used as the activation function of the final output layer of SDTU-net, and the rest of the network layers of SDTU-net all use Relu as activation function. Through multiple experiments, we set (Topkey, MinusKey) and (BotKey, PlusKey) to (0.4, 0.1) and (0.1, 0.05). The impact of the initial value of stepwise drop on algorithm performance will be tested and analyzed in Section V-B.

To compare the performance of SDTU-net, we take MUM-Net [11], U-net [22], M-net [23], Attention U-Net [24], MultiResUnet [25], Attention ResU-Net [26], TransUnet [35], and Swin-Unet [39] as comparison models. Among them, TransUnet contains 12 transformer layers; each module of Swin-Unet contains 2 transformer blocks, and all dropouts are set to 0. The above-mentioned models are all built by the Keras framework. The training process of these models uses the same dataset as SDTU-net, and the epochs and batch size are also the same as SDTU-net.

To maintain algorithm compatibility, we take Python as a programming language to implement each model-based subject-sensitive hash algorithm. All experiments were performed on a workstation with an RTX4090 GPU, Intel I7-13700K CPU, and 32G DDR4 RAM.

B. Examples of Integrity Authentication

In this section, we use a set of instances to make a preliminary comparison of each algorithm, and the HRRS images used for integrity authentication are shown in Fig. 4. The original HRRS image is shown in Fig. 4(a), stored in TIF format. Fig. 4(b) shows an example of image format conversion (TIF format to BMP format). Fig. 4(c) is the image after an invisible watermark is embedded (128-b watermark information is embedded in a single band using the least significant bit algorithm). Fig. 4(d) shows the results of 95% JPEG compression. Fig. 4(e)–(g) shows the results of Fig. 4(a) being tampered with subject-unrelated tampering and subject-related tampering, respectively. Fig. 4(h) is the image of being tampered with a random 24 × 24-pixel size area in Fig. 4(a).

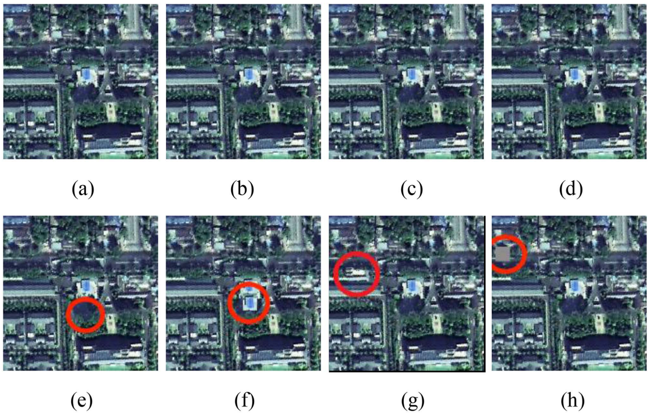


Fig. 4. HRRS images for integrity authentication. (a) Original HRRS image. (b) Format converted image (TIFF to BMP format). (c) Invisible watermark embedding. (d) JPEG compressed image. (e) Subject-unrelated tampered image. (f) and (g) Subject-related tampered image. (h) Randomly tampered image (24 × 24 pixel).

Obviously, the content of the formatted [see Fig. 4(b)] image, watermark embedded image [see Fig. 4(c)], and JPEG compressed image [see Fig. 4(d)] do not differ from that of the original image [see Fig. 4(a)]. However, since the original HRRS image is very different from the HRRS images shown in Fig. 4(b)–(d) at the binary level, cryptography-related techniques (such as MD5, DSA, and blockchains) treat the images shown in Fig. 4(a)–(d) as different data. It is not appropriate to consider these four images as different data, after all, users are concerned about the content carried by HRRS images, not the carrier itself.

The contents of HRRS images shown in Fig. 4(e)–(h) are significantly different from Fig. 4(a), that is, they have been tampered with. Among them, the tampered content in Fig. 4(e) is that the woods have been altered, not related to the building, that is, the tampering is subject-unrelated tampering. The tampering in Fig. 4(f)–(g) is subject-related tampering. Fig. 4(h) shows positional random tampering.

Each comparison algorithm was used to generate hash sequences of the HRRS image shown in Fig. 4(a)–(h). Then, *N-Dis* between the hash sequence of the original image shown in Fig. 4(a) and the hash sequences of the images shown in Fig. 4(b)–(h) is calculated, as shown in Table I.

The authentication results based on Table I are shown in Tables II and III, in which thresholds *T* is set to 0.02 and 0.05, respectively.

TABLE I
N-Dis of Each Algorithm Based on Different Models

	Fig. 4 (b)	Fig. 4 (c)	Fig. 4 (d)	Fig. 4 (e)	Fig. 4 (f)	Fig. 4 (g)	Fig. 4 (h)
Image manipulation	Format conversion	Invisible watermark embedding	JPEG compression	Subject-unrelated tampered image	Subject-related tampered image	Subject-related tampered image	Random tampered image
MUM-Net based algorithm	0	0	0.0117	0.0156	0.0976	0.1250	0.0859
U-net based algorithm	0	0	0.0312	0.0820	0.1835	0.1289	0.0351
M-net based algorithm	0	0	0.0156	0.0039	0.1250	0.1211	0.1210
MultiResUnet based algorithm	0	0	0.0117	0.0117	0.0429	0.0898	0.0390
Attention U-Net based algorithm	0	0	0.0039	0.0234	0.1093	0.2343	0.0507
Attention ResU-Net based algorithm	0	0	0.0078	0.0	0.0390	0.1484	0.0078
Swin-Unet based algorithm	0	0	0.0117	0.0156	0.0938	0.1875	0.0585
TransUnet based algorithm	0	0	0.0156	0.0429	0.1250	0.2109	0.1328
Our algorithm	0	0	0.0039	0.0429	0.1523	0.2578	0.1328

TABLE II
Integrity Authentication Result Based on Table I (Threshold *T* is 0.01)

	Fig. 4(b)	Fig. 4(c)	Fig. 4(d)	Fig. 4(e)	Fig. 4(f)	Fig. 4(g)	Fig. 4(h)
Image manipulation	Format conversion	Invisible watermark embedding	JPEG compression	Subject-unrelated tampered image	Subject-related tampered image	Subject-related tampered image	Random tampered image
MUM-Net based algorithm	Pass	Pass	Tampered	Tampered	Tampered	Tampered	Tampered
U-net based algorithm	Pass	Pass	Tampered	Tampered	Tampered	Tampered	Tampered
M-net based algorithm	Pass	Pass	Tampered	Pass	Tampered	Tampered	Tampered
MultiResUnet based algorithm	Pass	Pass	Tampered	Tampered	Tampered	Tampered	Tampered
Attention U-Net based algorithm	Pass	Pass	Pass	Tampered	Tampered	Tampered	Tampered
Attention ResU-Net based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Pass
Swin-Unet based algorithm	Pass	Pass	Tampered	Tampered	Tampered	Tampered	Tampered
TransUnet based algorithm	Pass	Pass	Tampered	Tampered	Tampered	Tampered	Tampered
Our algorithm	Pass	Pass	Pass	Tampered	Tampered	Tampered	Tampered

As seen from Table II, Attention U-Net, Attention ResU-Net, and our SDTU-net based algorithms maintain robustness to JPEG when threshold T is 0.01, and other algorithms fail to keep robustness to JPEG (95%) compression. However, Attention ResU-Net based algorithm exhibits poor tampering sensitivity, it failed to detect the tampering in Fig. 4(e) and (h). Algorithms based on Swin-Unet and TransUnet, two transformer-based models, fail to maintain robustness to JPEG compression.

From Table III, it can be seen that each algorithm keeps robustness to format conversion and JPEG compression when threshold T increases to 0.05, whereas M-net, MultiResUnet, and Attention ResU-Net based algorithms have reduced tampering sensitivity, and they failed to detect all of the malicious tampering shown in Fig. 4(f)–(h).

An ideal subject-sensitive hash algorithm should have the ability to distinguish subject-related tampering from subject-unrelated tampering. Combined with the results of subject-related tampering detection, algorithms based on MUM-Net, MultiResUnet, Attention U-Net, Swin-Unet, TransUnet, and our SDTU-net can achieve subject-sensitive authentication by setting different thresholds.

TABLE III
INTEGRITY AUTHENTICATION RESULT BASED ON TABLE I (THRESHOLD T IS 0.05)

	Fig. 4(b)	Fig. 4(c)	Fig. 4(d)	Fig. 4(e)	Fig. 4(f)	Fig. 4(g)	Fig. 4(h)
Image manipulation	Format conversion	Invisible watermark embedding	JPEG compression	Subject-unrelated tampered image	Subject-related tampered image	Subject-related tampered image	Random tampered image
MUM-Net based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered
U-net based algorithm	Pass	Pass	Pass	Tampered	Tampered	Tampered	Pass
M-net based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered
MultiResUnet based algorithm	Pass	Pass	Pass	Pass	Pass	Tampered	Tampered
Attention U-Net based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered
Attention ResU-Net based algorithm	Pass	Pass	Pass	Pass	Pass	Tampered	Pass
Swin-Unet based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered
TransUnet based algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered
Our algorithm	Pass	Pass	Pass	Pass	Tampered	Tampered	Tampered

Based on Tables II and III, we can see that our algorithm and Attention U-Net based algorithm performed better in this set of integrity authentication examples.

C. Robustness Testing of the Algorithms

To compare the robustness of each algorithm, we removed some unsatisfactory HRRS images in dataset Datasets$_{10000}$ (containing 10000 images) of [11] and added some HRRS images from DOTA and GF-2 satellite to dataset Datasets$_{10000}$ to construct a new dataset containing 11000 images, named *Datasets*$_{11000}$. Each image in Datasets$_{11000}$ is 256 × 256 pixels in size and stored in TIF format.

First, each algorithm's robustness to JPEG compression is tested. HRRS image in Datasets$_{11000}$ is compressed (95% JPEG compression) based on OpenCV's interface. Here, the proportion of HRRS images with normalized Hamming distances higher than threshold T is used to describe algorithm's robustness: the lower the proportion, the better the robustness of the algorithm at the corresponding threshold T. The results of the robustness test to JPEG (95%) compression are shown in Table IV. From Table IV, we can see that our SDTU-net based algorithm is the best of these algorithms, especially at a lower threshold such as 0.02. At medium thresholds, such as 0.05, our algorithm is slightly more robust than MUM-net based algorithms and significantly better than other algorithms. At higher thresholds, such as 0.1, each algorithm's robustness to JPEG (95%) compression is ideal, but a higher threshold will result in a decrease in algorithm's tampering sensitivity, which means that higher thresholds are often not used in actual integrity authentication. In short, our SDTU-net based algorithm's robustness to JPEG (95%) compression is not only a great improvement over subject-sensitive hashing based on CNN networks such as M-net and MUM-net, but also better than algorithms based on attention mechanism models such as Attention ResU-Net, Attention U-net, and TransUnet.

TABLE IV
COMPARISON OF ROBUSTNESS TO JPEG (95%) COMPRESSION

	T=0.1	T=0.05	T=0.03	T=0.02	T=0.01
MUM-Net based algorithm	0.2%	1.6%	6.9%	15.3%	43.9%
U-net based algorithm	2.0%	9.2%	20.2%	29.5%	55.0%
M-net based algorithm	2.2%	8.3%	19.5%	31.3%	59.5%
MultiResUnet based algorithm	2.1%	5.1%	11.3%	17.3%	41.4%
Attention U-Net based algorithm	0.8%	3.4%	8.4%	14.5%	37.6%
Attention ResU-Net based algorithm	0.7%	3.5%	8.6%	12.3%	24.5%
Swin-Unet based algorithm	1.4%	6.4%	13.2%	19.0%	41.4%
TransUnet based algorithm	1.5%	6.8%	16.4%	24.1%	51.9%
Our algorithm	0.0%	0.6%	2.2%	4.5%	21.9%

Next, each algorithm's robustness to digital watermark embedding is tested. For each image in Datasets$_{11000}$, we use least significant bit (LSB) algorithm to embed 24 b of information in each of its bands, not just one band. The test results are shown in Table V.

TABLE V
ROBUSTNESS TEST COMPARISON OF WATERMARKING EMBEDDING

	T=0.1	T=0.05	T=0.03	T=0.02	T=0.01
MUM-Net based algorithm	1.9%	10.4%	20.1%	28.7%	54.7%
U-net based algorithm	3.1%	10.%	21.8%	30.2%	53.5%
M-net based algorithm	1.2%	7.9%	20.8%	32.7%	57.5%
MultiResUnet based algorithm	0.3%	1.9%	5.7%	10.6%	30.3%
Attention U-Net based algorithm	1.3%	2.5%	4.9%	6.4%	21.5%
Attention ResU-Net based algorithm	0.5%	1.5%	3.0%	4.7%	11.1%
Swin-Unet based algorithm	0.0%	0.2%	0.8%	2.5%	12.8%
TransUnet based algorithm	1.0%	3.2%	8.5%	14.2%	39.9%
Our algorithm	0.2%	0.7%	2.9%	7.6%	27.8%

From Table V, it can be found that our algorithm's robustness to multiband watermark is inferior to Attention ResUNet and Swin-Unet based algorithms. However, as can be seen from Sections IV-D and IV-E, Attention ResUNet and Swin-Unet based algorithms have poor tampering sensitivity and computational performance, which makes the overall performance of the two algorithms unsatisfactory.

D. Tampering Sensitivity Testing of Algorithms

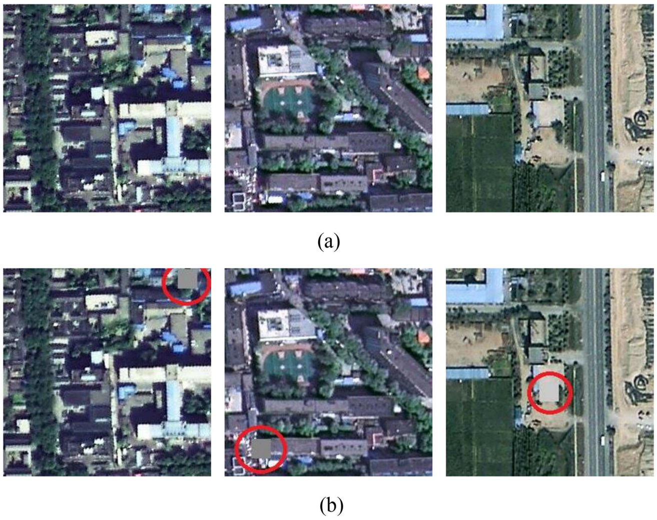
In this section, dataset Datasets$_{11000}$ is still used to test each algorithm's tampering sensitivity.

First, each algorithm's tampering sensitivity to rectangular area tampering with random locations is tested: Each image in Datasets$_{11000}$ is randomly selected for an area of 24 × 24 pixels, and each pixel in the region is set to a random value to simulate image tampering in reality. A set of HRRS images before and after tampering is shown in Fig. 5.

Here, we also use the proportion of tampered images with normalized Hamming distance (*N-Dis*) above the threshold T to describe the algorithm's tampering sensitivity: the higher the proportion, the better the tampering sensitivity. The results are shown in Table VI.

From Table VI, it can be found that our SDTU-net-based algorithm's tampering sensitivity is similar to MUM-net and TransUnet based algorithms, stronger than MultiResUnet and Attention ResU-Net based algorithms, and slightly weaker than that of M-net and U-net based algorithm at a higher threshold (such as $T = 0.1$).

Next, each algorithm's tampering sensitivity to random pixel tampering is tested: for each image in *Datasets*$_{11000}$, 64 pixels are randomly selected in each image to set to 255. The results are shown in Table VII.



(a)

(b)

Fig. 5. Examples of tampering with 24 × 24 pixels area. (a) Original images. (b) Tampered images.

TABLE VI
TAMPERING SENSITIVITY TEST FOR 24 × 24 PIXELS TAMPERING

	T=0.1	T=0.05	T=0.03	T=0.02	T=0.01
MUM-Net based algorithm	70.6%	94.5%	98.6%	99.4%	99.9%
U-net based algorithm	82.6%	95.9%	98.7%	99.1%	99.7%
M-net based algorithm	81.7%	97.1%	98.7%	99.1%	99.6%
MultiResUnet based algorithm	39.2%	71.6%	86.1%	90.7%	95.5%
Attention U-Net based algorithm	63.5%	85.3%	93.9%	96.0%	98.5%
Attention ResU-Net based algorithm	16.9%	33.4%	48.2%	57.1%	75.5%
Swin-Unet based algorithm	65.6%	91.4%	96.5%	97.8%	99.4%
TransUnet based algorithm	68.2%	93.0%	97.8%	98.4%	99.2%
Our algorithm	66.4%	93.5%	98.1%	99.1%	99.5%

From Table VII, it can be seen that our algorithm is optimal except that attention U-Net based algorithm has better tamper sensitivity at a high threshold ($T = 0.1$). Attention ResUNet and Swin-Unet based algorithms have poor tampering sensitivity to random pixel tampering.

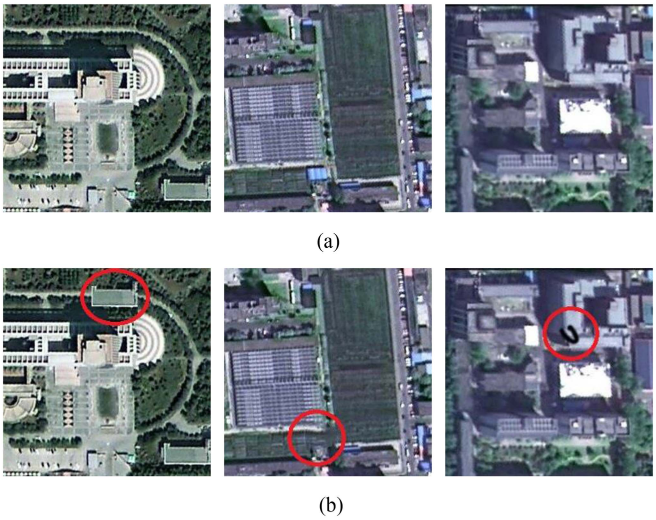
Subject sensitivity is the main difference between perceptual hashing and subject-sensitive hashing, which means subject-sensitive hashing is more effective for detecting subject-related tampering. Since no dataset for testing subject-related tampering is now exposed, we build a dataset containing 400 tampering instances to test the algorithm's subject sensitivity. This dataset takes buildings as subject and contains 200 tampering instances for adding buildings and 200 tampering instances for deleting

TABLE VII
TAMPERING SENSITIVITY TEST FOR MODIFICATION OF 64 RANDOM PIXELS

	T=0.1	T=0.05	T=0.03	T=0.02	T=0.01
MUM-Net based algorithm	97.5%	99.9%	100%	100%	100%
U-net based algorithm	98.2%	99.9%	100%	100%	100%
M-net based algorithm	87.2%	96.9%	98.3%	98.9%	99.4%
MultiResUnet based algorithm	51.1%	82.4%	90.4%	93.3%	96.0%
Attention U-Net based algorithm	99.5%	99.9%	100%	100%	100%
Attention ResU-Net based algorithm	27.5%	70.6%	87.8%	92.3%	96.5%
Swin-Unet based algorithm	19.6%	57.9%	81.8%	89.2%	97.5%
TransUnet based algorithm	95.8%	99.9%	100%	100%	100%
Our algorithm	95.1%	100%	100%	100%	100%

TABLE VIII
TAMPERING SENSITIVITY TEST FOR SUBJECT-RELATED TAMPERING

	T=0.1	T=0.05	T=0.03	T=0.02	T=0.01
MUM-Net based algorithm	90.25 %	98.50 %	100%	100%	100%
U-net based algorithm	81.00 %	97.50 %	99.25	100%	100%
M-net based algorithm	91.25 %	100%	100%	100%	100%
MultiResUnet based algorithm	55.50 %	85.25 %	95.75 %	96.00 %	99.25 %
Attention U-Net based algorithm	82.75 %	98.00 %	99.50 %	100%	100%
Attention ResU-Net based algorithm	18.00 %	42.00 %	62.75 %	72.50 %	84.50 %
Swin-Unet based algorithm	81.00 %	97.50 %	100%	100%	100%
TransUnet based algorithm	99.00 %	100%	100%	100%	100%
Our algorithm	92.25 %	100%	100%	100%	100%



(a)

(b)

Fig. 6. Instances of Subject-related tampering (taking buildings as subjects). (a) Original images. (b) Tampered images.

buildings. Three tampering instances of this dataset are shown in Fig. 6. The tampering methods of these three instances are adding buildings, deleting buildings, and smearing the contents of buildings. Test results are shown in Table VIII.

As seen from Table VIII, our SDTU-net based algorithm has good sensitivity to subject-related tampering and is only slightly inferior to TransUnet based algorithm at a higher threshold ($T = 0.1$), and overall better than other algorithms.

E. Comparison of Computing Performance

Since deep learning-based subject-sensitive hashing needs to load the trained model first every time it starts computation, the average time for the algorithm to calculate a hash sequence of different numbers of HRRS images may be different. To avoid the chance caused by a single amount of computation, we built four test datasets based on Datasets$_{11000}$, each containing 10, 200, 2000, and 10000 HRRS images.

We evaluate the computational performance of the algorithm from three perspectives: total time, average time and frames per second (FPS). The test results are shown in Table IX.

From Table IX, the following conclusions can be drawn.

1) Each algorithm exhibits different computational performance under different datasets. The average computation time when there is less test data (such as ten images) is greater than that when the data volume is large. Our SDTU-net-based algorithm achieves stable computational speed at several 2000 HRRS images per batch.
2) The computational performance of algorithms based on traditional CNNs is generally better than that of transformer-based algorithms. After all, the transformer itself has a high level of computational complexity.
3) The computational performance of our SDTU-net-based algorithm is at a medium level among these algorithms. It is better than that of Swin-Unet, TransUnet, Attention ResU-Net, and MultiResUnet, but not as good as the algorithms based on MUM-net, U-net, M-net, and Attention U-net algorithms. This means that it is necessary to improve the computational performance of the algorithm in our next research.

V. DISCUSSION

A. Comprehensive Evaluation of Algorithm Performance

An ideal subject-sensitive hash algorithm (also known as subject-sensitive hashing) should have both high robustness and high tampering sensitivity. However, robustness and tampering sensitivity are often contradictory, and some algorithms' tampering sensitivity is reduced when robustness is improved: for

TABLE IX
TESTS OF COMPUTE PERFORMANCE

	10 images			200 images			2000 images			10000 images		
	Total time (s)	Average time (ms)	FPS	Total time (s)	Average time (ms)	FPS	Total time (s)	Average time (ms)	FPS	Total time (s)	Average time (ms)	FPS
MUM-Net based algorithm	1.8	180	5.56	4.3	21.5	46.51	30.5	15.3	65.57	148.7	14.9	67.25
U-net based algorithm	1.1	110	9.09	2.6	13.0	76.92	26.3	13.1	76.05	133.6	13.4	74.85
M-net based algorithm	1.2	120	8.33	3.1	15.5	64.52	23.8	11.9	84.03	117.4	11.7	85.18
MultiResUnet based algorithm	4.1	410	2.44	7.5	37.5	26.67	48.0	24.0	41.67	231.9	23.2	43.12
Attention U-Net based algorithm	1.3	130	7.69	3.5	17.5	57.14	27.4	13.7	72.99	144.6	14.5	69.16
Attention ResU-Net based algorithm	3.2	320	3.13	6.5	32.5	30.77	44.4	22.2	45.05	226.5	22.7	44.15
Swin-Unet based algorithm	5.3	530	1.89	18.2	76.0	10.99	143.5	71.8	13.94	740.5	74.1	13.50
TransUnet based algorithm	4.7	470	2.13	11.4	57.0	17.54	76.4	38.2	26.18	371.0	37.1	26.95
Our algorithm	2.2	220	4.55	5.7	28.5	30.09	42.8	21.4	46.73	204.8	20.5	48.83

example, Attention ResUNet based algorithm performs well in robustness, but it does not perform well in tamper sensitivity.

In actual integrity authentication, low thresholds are often used to ensure that malicious tampering is detected. In this way, how to make the algorithm have good robustness while keeping high tamper sensitivity has become a key issue. According to the experimental results shown in Tables IV–IX, it can be found that our SDTU-net based algorithm has the best comprehensive performance.

1) Robustness: In general, the robustness of our algorithm is better than that of existing algorithm, especially the robustness of JPEG (95%) compression. Although Attention ResUNet based subject-sensitive hash algorithm is more robust to LSB digital watermark at low thresholds, the difference is not obvious, and the robustness at high thresholds is still not as good as our algorithm.

2) Tampering Sensitivity: From the experimental results shown in Tables VI–VIII, the tampering sensitivity of our algorithm is at the same level as that of existing algorithms, and there is no obvious gap.

3) Computing Performance: The computational performance of each algorithm participating in the comparison varies greatly, whereas our SDTU-net-based algorithm is at the medium level. Although our SDTU-net is better than other transformer-based models such as Swin-Unet and TransUnet, it is significantly inferior to traditional CNN network models such as U-net and M-net.

4) Security: Security of a subject-sensitive hash algorithm mainly refers to unidirectionality and uninterpretability of deep neural networks used by subject-sensitive hashing satisfies this well. As each algorithm used a deep neural network to extract the feature of HRRS images, the security of each algorithm is at the same level.

TABLE X
ROBUSTNESS TO JPEG COMPRESSION UNDER DIFFERENT MULTIDROP
($T = 0.02$)

Decoder Drop Encoder Drop	*BotKey=0.1* *PlusKey=0.05*	*BotKey=0.2* *PlusKey=0.0*	*BotKey=0.2* *PlusKey=0.05*
TopKey=0.2 *MinusKey=0.0*	6.4%	15.0%	8.0%
TopKey=0.3 *MinusKey=0.1*	11.6%	15.4%	21.9%
TopKey=0.4 *MinusKey=0.05*	11.9%	13.1%	15.5%
TopKey=0.4 *MinusKey=0.1*	4.5%	31.4%	9.8%

B. *Impact of Stepwise Drop on Algorithm's Performance*

In this section, the impact of stepwise drop on algorithm performance is tested. While keeping the network structure as shown in Fig. 3 unchanged, we set (Topkey, MinusKey) and (BotKey, PlusKey) to different values, and constructed 12 sets of stepwise-drop settings. In fact, this means that 12 different models have been built. For example, $Decrease = 0.0$ means that the drop rate in the encoder stage is a constant value, and Increase $= 0.0$ means that the drop rate in the decoder stage is a constant value.

After training these 12 models using the same training dataset, the algorithm's robustness to JPEG compression and tampering sensitivity to random tampering in 24 × 24 pixel region under the threshold $T = 0.02$ were tested, and the results are shown in Tables X and XI.

TABLE XI
TAMPERING SENSITIVITY TO 24 × 24 PIXELS TAMPERING UNDER DIFFERENT MULTIDROP ($T = 0.02$)

Decoder Drop Encoder Drop	*BotKey =0.1* *PlusKey =0.05*	*BotKey =0.2* *PlusKey =0.0*	*BotKey =0.2* *PlusKey =0.05*
Topkey=0.2 *MinusKey =0.0*	98.1%	96.5%	97.9%
Topkey=0.3 *MinusKey =0.1*	97.3%	96.8%	95.2%
Topkey=0.4 *MinusKey =0.05*	96.2%	95.3%	95.9%
Topkey=0.4 *MinusKey =0.1*	99.1%	95.4%	90.2%

As can be seen from Table X, when (Topkey, Decrease) and (Bottomkey, Increase) are set to (0.4, 0.1) and (0.1, 0.05) respectively, the algorithm's robustness is the best, and this set of values is exactly what our SDTU-net based algorithm uses.

Similarly, as can be seen from Table XI, when (Topkey, Decrease) and (Bottomkey, Increase) are set to (0.4, 0.1) and (0.1, 0.05), tampering sensitivity is also the best, which means that this set of stepwise drop can make the algorithm have good tamper sensitivity and robustness at the same time.

Combining Tables X and XI, it can be found that different drop rates at different layers of deep neural networks have a large impact on the performance of a subject-sensitive hash algorithm, and our stepwise-drop mechanism is effective for improving the subject-sensitive hash algorithm. The setting of the stepwise drop is a whole: coordinating the different drop rates of the encoder and the decoder can make the model optimal, and adjusting the drop rate of the encoder stage or decoder stage alone cannot make the algorithm have a good comprehensive performance. The initial value of the stepwise drop that we set in Section IV can optimize the overall performance of our algorithm.

C. Effect of the Number of Transformer Layers

In addition to stepwise drop, another difference between our SDTU-net and TransUnet is that there are4 transformer layers in STDU-net and 12 transformer layers in TransUnet. This is not only because transformer is computationally intensive, but more importantly, too many transformer layers have limited improvement on the tampering sensitivity of an algorithm, and will reduce the robustness. For the subject-sensitive hash algorithm, overextracted features will affect the algorithm's robustness, whereas insufficient extracted features will reduce the algorithm's tampering sensitivity.

To test the effect of different number of transformer layers on the performance of subject-sensitive hashing, we build networks with 2, 4, 8, and 12 transformer layers while keeping network structure and stepwise-drop parameters unchanged. The robustness to JPEG compression and tampering sensitivity to random tampering in 24 × 24 pixel region of the algorithms under different transformer layers is tested. Here, (Topkey, MinusKey) and (Botkey, PlusKey) are set to (0.4, 0.1) and (0.1, 0.05), consistent with Section IV.

TABLE XII
ROBUSTNESS TO JPEG (95%) UNDER DIFFERENT TRANSFORMER LAYERS ($T = 0.02$)

Number of transformer layers	*T*=0.1	*T*=0.05	*T*=0.03	*T*=0.02	*T*=0.01
12	0.8%	4.3%	11.9%	21.1%	44.0%
8	0.1%	0.6%	4.1%	9.3%	29.8%
4	0.0%	0.6%	2.2%	4.5%	21.9%
2	0.1%	0.8%	3.0%	8.5%	29.6%

TABLE XIII
TAMPERING SENSITIVITY TO 24 × 24 PIXELS TAMPERING UNDER DIFFERENT TRANSFORMER LAYERS ($T = 0.02$)

Number of transformer layers	*T*=0.1	*T*=0.05	*T*=0.03	*T*=0.02	*T*=0.01
12	70.2%	84.9%	90.1%	91.9%	95.8%
8	52.9%	84.7%	94.5%	96.5%	98.4%
4	66.4%	93.5%	98.1%	99.1%	99.5%
2	55.8%	81.3%	88.6%	91.6%	96.0%

As can be seen from Table XII, an algorithm's robustness to JPEG compression is the best when the number of transformer layers is 4, and increasing transformer layers does not make the algorithm more robust.

From Table XIII, it can be found that the model with 4 transformer layers performs better at a low and medium threshold, although it has a weaker tamper sensitivity at a high threshold than the model with 12 transformer layers. Moreover, the model with four transformer layers has better tamper sensitivity than models with two and eight transformer layers.

Combined with Tables XII and XIII, it can be seen that the algorithm's overall performance is optimal when there are 4 transformer layers, and increasing the number of transformer layers does not have a significant improvement on the algorithm's performance.

VI. CONCLUSION

In this article, a new deep neural network model named SDTU-net for the subject-sensitive hash algorithm of HRRS images is proposed. The drop rate of different network layers of SDTU-net is determined according to our proposed stepwise-drop mechanism. Based on the experiments and discussions, the following conclusions can be drawn.

1) Dropout has a great impact on the tampering sensitivity and robustness of subject-sensitive hashing.

2) Our proposed stepwise drop can significantly adjust the subject-sensitive hash algorithm's performance by setting different drop rates for different layers of deep neural networks.
3) The SDTU-net based algorithm has good comprehensive performance, especially at medium and low thresholds, which solves the problem that existing algorithms cannot balance robustness and tamper sensitivity at low thresholds.

However, due to the high computational complexity of the transformer, the computational efficiency of our SDTU-net is inferior to that of pure convolutional neural networks such as M-net. On the other hand, existing subject-sensitive hash algorithms are mostly for a single subject and do not have the sensitivity to multisubject. Therefore, our future research include improving the computational efficiency of algorithms and exploring multisubject-sensitive hashing.

References

[1] Y. Yang et al., "AR2Det: An accurate and real-time rotational one-stage ship detector in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5605414.

[2] W. Qiao, L. Shen, J. Wang, X. Yang, and Z. Li, "A weakly supervised semantic segmentation approach for damaged building extraction from postearthquake high-resolution remote-sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 6002705.

[3] G. Yin, A. Li, W. Zhao, H. Jin, J. Bian, and S. Wu, "Modeling canopy reflectance over sloping terrain based on path length correction," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4597–4609, Aug. 2017.

[4] Y. Zhao, P. Chen, Z. Chen, Y. Bai, Z. Zhao, and X. Yang, "A triple-stream network with cross-stage feature fusion for high-resolution image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5600417.

[5] L. Jiang, H. Zheng, and C. Zhao, "A fragile watermarking in ciphertext domain based on multi-permutation superposition coding for remote sensing image," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 5664–5667.

[6] R. Maruthi, P. Anusha, and K. Bhuvaneswari, "An effective payload management (2,3) scheme for visual secret sharing over remote sensing images," in *Proc. 1st Int. Conf. Elect., Electron., Inf. Commun. Technol.*, 2022, pp. 1–6.

[7] X. Ouyang, Y. Xu, Y. Mao, Y. Liu, Z. Wang, and Y. Yan, "Blockchain-assisted verifiable and secure remote sensing image retrieval in cloud environment," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1378–1389, 2023.

[8] Z. Tang, X. Li, X. Zhang, S. Zhang, and Y. Dai, "Image hashing with color vector angle," *Neurocomputing*, vol. 308, pp. 147–158, 2018.

[9] L. Du, A. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Image Commun.*, vol. 81, 2020, Art. no. 115713.

[10] C. Qin, M. Sun, and C. Chang, "Perceptual hashing for color images based on hybrid extraction of structural features," *Signal Process.*, vol. 36, pp. 194–205, Jan. 2018.

[11] K. Ding, Y. Liu, Q. Xu, and F. Lu, "A subject-sensitive perceptual hash based on MUM-net for the integrity authentication of high resolution remote sensing images," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 8, 2020, Art. no. 485.

[12] K. Ding, S. Chen, Y. Wang, Y. Liu, Y. Zeng, and J. Tian, "AAU-net: Attention-based asymmetric U-net for subject-sensitive hashing of remote sensing images," *Remote Sens.*, vol. 13, no. 24, 2022, Art. no. 5109.

[13] C. Liu, J. Ma, X. Tang, F. Liu, X. Zhang, and L. Jiao, "Deep hash learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3420–3443, Apr. 2021.

[14] Z. Huang, Z. Tang, X. Zhang, L. Ruan, and X. Zhang, "Perceptual image hashing with locality preserving projection for copy detection," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 463–477, Jan./Feb. 2023.

[15] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-scale remote sensing image retrieval by deep hashing neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 950–965, Feb. 2018.

[16] K. Ding, Z. Yang, Y. Wang, and Y. Liu, "An improved perceptual hash algorithm based on U-Net for the authentication of high-resolution remote sensing image," *Appl. Sci.*, vol. 9, no. 15, Jul. 2019, Art. no. 2972.

[17] X. Zhang, H. Yan, L. Zhang, and H. Wang, "High-resolution remote sensing image integrity authentication method considering both global and local features," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 4, Apr. 2020, Art. no. 254.

[18] S. Dong, Y. Zhuang, Z. Yang, L. Pang, H. Chen, and T. Long, "Land cover classification from VHR optical remote sensing images by feature ensemble deep learning network," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1396–1400, Aug. 2020.

[19] X. Yao, Q. Guo, and A. Li, "Cloud detection in optical remote sensing images with deep semi-supervised and active learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 6006805.

[20] J. Wang, W. Li, M. Zhang, R. Tao, and J. Chanussot, "Remote-sensing scene classification via multistage self-guided separation network," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5615312, doi: 10.1109/TGRS.2023.3295797.

[21] M. Zhang, W. Li, Y. Zhang, R. Tao, and Q. Du, "Hyperspectral and LiDAR data classification based on structural optimization transmission," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3153–3164, May 2023, doi: 10.1109/TCYB.2022.3169773.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, vol. 9351, pp. 234–241.

[23] V. Adiga and J. Sivaswamy, "FPD-M-net: Fingerprint image denoising and inpainting using M-net based convolutional neural networks," in *Inpainting and Denoising Challenges. (The Springer Ser. on Challenges in Mach. Learn.)*, S. Escalera, S. Ayache, J. Wan, M. Madadi, U. Güçlü, and X. Baró, Eds. Cham, Switzerland: Springer, 2019.

[24] O. Oktay et al., "Attention U-Net: Learning where to look for the pancreas," in *Proc. 1st Conf. Med. Imag. Deep Learn.*, 2018, pp. 1–10.

[25] N. Ibtehaz and M. Rahman, "MultiResUNet: Rethinking the U-net architecture for multimodal biomedical image segmentation," *Neural Netw.*, vol. 121, pp. 74–87, 2020.

[26] R. Li, S. Zheng, C. Duan, J. Su, and C. Zhang, "Multistage attention ResU-net for semantic segmentation of fine-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 8009205.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[28] B.-S. Ko, H.-G. Kim, K.-J. Oh, and H.-J. Choi, "Controlled dropout: A different approach to using dropout on deep neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput.*, 2017, pp. 358–362.

[29] H. Inoue, "Multi-sample dropout for accelerated training and better generalization," 2019, *arXiv:1905.09788*.

[30] L. Liu, Y. Luo, X. Shen, M. Sun, and B. Li, "β-dropout: A unified dropout," *IEEE Access*, vol. 7, pp. 36140–36153, 2019.

[31] B. Li et al., "DropKey," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22700–22709.

[32] Z. Lu, C. Xu, and B. Du, "MultiDrop: A local rademacher complexity-based regularization for multitask models," in *Proc. Int. Joint Conf. Neural Netw.*, 2023, pp. 1–6.

[33] Z. Chen, J. Wang, A. Ma, and Y. Zhong, "TypeFormer: Multiscale transformer with type controller for remote sensing image caption," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6514005.

[34] S. Zhang, Y. Cao, and B. Sui, "DTHNet: Dual-stream network based on transformer and high-resolution representation for shadow extraction from remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 8000905.

[35] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[36] T. Wu et al., "MTU-Net: Multilevel TransUNet for space-based infrared tiny ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5601015.

[37] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin transformer embedding UNet for remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4408715.

[38] S. P. Ji and S. Y. Wei, "Building extraction via convolutional neural networks from an open remote sensing building dataset," *Acta Geodaetica Cartographica Sinica*, vol. 48, pp. 448–459, 2019.

[39] H. Cao, "Swin-UNet: UNet-like pure transformer for medical image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 205–218.

Kaimeng Ding received the Ph.D. degree in geographic information systems from Nanjing Normal University, Nanjing, China, in 2015.

He is currently an Associate Professor with Jinling Institute of Technology, Nanjing, China. His research interests include subject-sensitive hashing, perceptual hashing, security technology for geodata, and the application of deep learning.

Shiping Chen (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales, Sydney, Australia, in 2001.

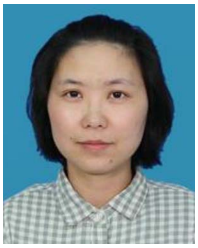
He is currently a Principal Research Scientist with the Common wealth Scientific and Industrial Research Organisation, Data61, Sydney, NSW, Australia. His research interests include blockchain, service-oriented collaborations, and cybersecurity. He is also actively involved in service/cloud computing research communities through journal editorships, publications, and conference, including EDOC, IEEE ICWS/SCC/CLOUD, and ICSOC.

Yue Zeng received the Ph.D. degree in computer science and technology from Xidian University, Xi'an, China, in 2011.

He is currently a Professor with Jinling Institute of Technology, Nanjing, China. He is a Senior Member of CCF. His research interests include intelligent information processing and data security.

Yanan Liu received the Ph.D. degree in computer science and technology from Nanjing University of Aeronautics and Astronaut, Nanjing, China, in 2013.

She is currently an Associate Professor with Jinling Institute of Technology, Nanjing, China. Her research interests include applications of cryptography and security protocols.



Bei Xu received the Ph.D. degree in meteorology from Nanjing University of Information Science & Technology, Nanjing, China, in 2019.

She is currently a Lecturer with Jinling Institute of Technology, Nanjing, China. Her interests include land surface process application and short-term climate prediction.

Yingying Wang received the Ph.D. degree in geographic information systems from Nanjing Normal University, Nanjing, China, in 2018.

She is currently a Lecturer with Jinling Institute of Technology, Nanjing, China. Her research interests include data security of geodata and applications of geographic information systems.