Expert Systems With Applications 238 (2024) 122105

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Deep global semantic structure-preserving hashing via corrective triplet loss for remote sensing image retrieval

Hongyan Zhou [a], Qibing Qin [b,*], Jinkui Hou [b,*], Jiangyan Dai [b], Lei Huang [c], Wenfeng Zhang [d]

[a] School of Computer Science, Qufu Normal University, Rizhao, China
[b] School of Computer Engineering, Weifang University, Weifang, China
[c] College of Information Science and Engineering, Ocean University of China, Qingdao, China
[d] College of Computer and Information Science, Chongqing Normal University, Chongqing, China

ARTICLE INFO

Keywords:
Remote sensing image retrieval
Deep hashing
Global contextual information
Corrective triplet loss
Swin transformer
Asymmetric learning

ABSTRACT

With the explosive increase of remote sensing data, how to search for remote sensing data quickly and accurately in a vast dataset is an incredibly critical matter for research subjects. The deep hashing method has become the dominant method for remote sensing image retrieval because of its low-cost storage and high-speed retrieval. However, for the reason of the limitation of fixed convolutional kernels, deep hashing frameworks based on Convolutional Neural Networks (CNNs) fail to obtain the global semantic information well, which leads to the generation of suboptimal solutions. Furthermore, existing hashing methods commonly employ the random sampling strategy or hardest sample mining to build training batches, resulting in bad local minima. To remedy these problems, a novel Deep Global Semantic Structure-preserving Hashing framework via corrective triplet loss (DGSSH) is proposed for remote sensing image retrieval to learn a discriminative and stable embedding space, achieving intra-class confusion and inter-class diversity. Specifically speaking, the feature extraction module based on Swim Transformer architecture is developed to capture global semantic information and multiscale features from remote sensing images. Based on a distribution matching constraint, the corrective triplet loss for deep hashing schemes is designed to reduce the distribution shift caused by the random selection or hardest sample mining. Meanwhile, to reduce the time overhead of the model, the asymmetric learning strategy is employed to perform effective compact representation learning. Numerous experiments have been carried out on three publicly available benchmarks, which indicates that the proposed DGSSH framework could achieve optimal performance for remote sensing image retrieval applications. The source code of our DGSSH framework is hosted at https://github.com/QinLab-WFU/DGSSH.git.

1. Introduction

Since the beginning of the 21st century, the amount and quality of remote sensing data have been explosively increasing as Earth observation technology is evolving at a rapid pace. Millions of remote sensing data, which are acquired by satellite sensors, are deposited in several large datasets (Tong et al., 2019). At the moment, remote sensing image data serves an essential role in disaster prevention and control (Song, Park and Park, 2022), ecological environment monitoring (Tan et al., 2022) and other domains, which has captured the interest of many scholars and social groups. Hence, how to efficiently and accurately search the desired target data has become an important subject and urgent task.

Being the most typical method for massive image search, hash learning is the research emphasis of many scholars (Li, Zhang, Huang, Zhu and Ma, 2017; Zhu, Lu, Cheng, Li, & Zhang, 2020; Zhu et al., 2023). The purpose of hash learning is to map the raw data into the discrete Hamming space via hash projection, and the similarity between images is calculated with a binary code XOR operation. This approach is able to reduce storage space and promote retrieval efficiency while also preserving the original similarity between images. Analyzed from the perspective of whether the training data are used or not, the current hashing methods primarily comprise data-independent hashing and data-dependent hashing. Data-independent learning, like Locality Sensitive Hashing (LSH) (Kulis & Grauman, 2009), Spectral Hashing (SH) (Weiss, Torralba, & Fergus, 2008), and Iterative Quantization (ITQ) (Gong, Lazebnik, Gordo, & Perronnin, 2012), do not exploit training samples to train the hashing model, hence they require longer binary codes to keep the discrimination of learned hash codes. Instead,

[*] Corresponding authors.
E-mail addresses: zhouhongyan199607@163.com (H. Zhou), qinbing@wfu.edu.cn (Q. Qin), JKHOU@163.com (J. Hou), daijy@wfu.edu.cn (J. Dai), huangl@ouc.edu.cn (L. Huang), itzhangwf@cqnu.edu.cn (W. Zhang).

https://doi.org/10.1016/j.eswa.2023.122105
Received 9 June 2023; Received in revised form 28 August 2023; Accepted 9 October 2023
Available online 12 October 2023
0957-4174/

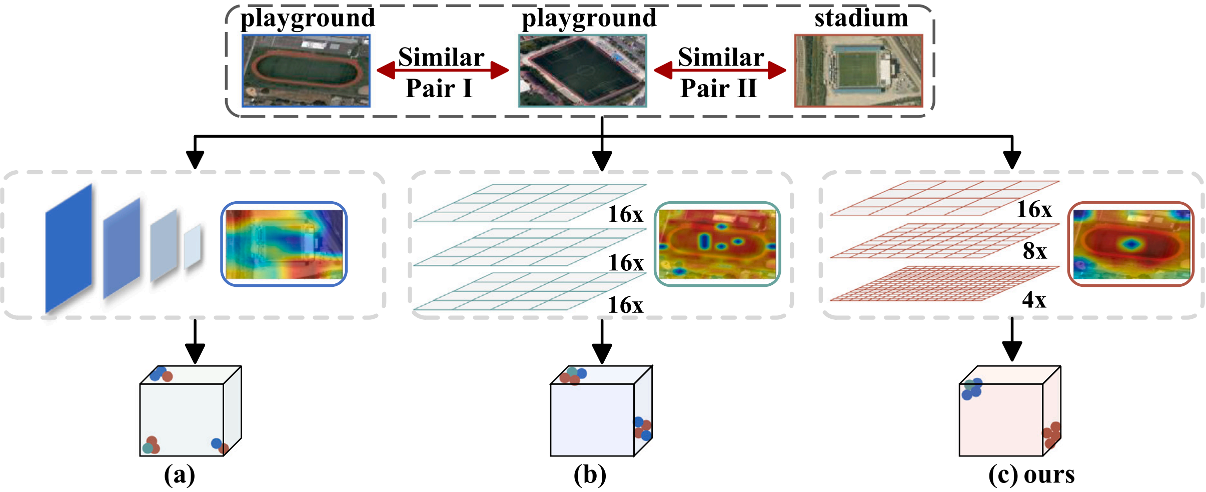

Fig. 1. Description of the motivation for the DGSSH method. (a) Traditional CNNs-based remote sensing image retrieval methods commonly exploit the downsampling operation to capture multiscale features, but the limitations of convolution kernels could make it hard to fully extract global semantic information. (b) Due to the feature map of fixed size, Vision Transformer (ViT) overlooks the multiscale characteristics of remote sensing images. (c) By introducing the layer-by-layer downsampling structure to improve the range of perceptual field, on the basis of Swin Transformer architecture, our proposed module of extracting features can obtain feature maps of different sizes and generate the global contextual descriptors, which is beneficial to give better discriminative power to the final binary code.

for data-dependent hashing, the training sample label information is utilized to explore the hash projection, which can realize high retrieval performance with shorter hash codes. As a result, data-dependent learning has been of considerable concern to many researchers. Available data-dependent hashing learning is roughly classified as supervised learning (Alizadeh, Helfroush, & Müller, 2023; Lu et al., 2019; Qin et al., 2022), unsupervised learning (Li, Zhang, Tao, & Zhu, 2016; Qin et al., 2021; Qin, Huang, Wei, Xie, & Zhang, 2020), and semi-supervised learning (Tang, Liu, Ma et al., 2019; Tang, Liu, Zhang et al., 2019). Compared to unsupervised and semi-supervised learning, supervised learning makes use of supervised signals, i.e., label information, to train the hashing framework, which could realize satisfactory performance.

In the last several years, deep learning has gained dramatic success in computer vision tasks (Li, Zhang, Pei, & Gan, 2022; Tan, Pang, & Le, 2020; Zhang, Huang, Wei, & Nie, 2021) with the increasing maturity of its development. Deep hashing-based framework has aroused the attention of an overwhelming number of researchers, and it gradually becomes the principal technique to retrieve remote sensing images (Chen, Zou, Shao, Sun, & Qin, 2018). The researchers employ CNNs to capture deep characteristics from complicated remote sensing images and leverage hash learning to get the binary codes for fast retrieval. Simultaneously, Vision Transformer (ViT) framework (Dosovitskiy et al., 2020) is proposed for image classification tasks in 2020 and is also extended to image retrieval (Dubey, Singh, & Chu, 2022; El-Nouby, Neverova, Laptev, & Jégou, 2021), achieving excellent performance.

Although making great progress, current deep hashing for remote sensing image retrieval still encounters some underexplored limitations: (1) Remote sensing images generally contain a large number of complex backgrounds unrelated to the content and include scale variations target. Available CNNs-based remote sensing image hashing (Song, Gao et al., 2022) exploits the fixed-size convolutional kernels to extract high-level semantic characteristics. Theoretically, the range of the perceptual field should overlap the whole images, but in practice, its range is much smaller than the theoretical ones, which can make it difficult to fully explore the global semantic features of the input samples (Liu et al., 2021; Peng, Qian, Wang, Liu, & Dong, 2023). Besides, based on the fixed-size and low-resolution feature map, Vision Transformer (ViT) tends to overlook the multiscale features of the remote sensing image. Meanwhile, ViT isolates the information communication between different windows (Dosovitskiy et al., 2020; Liu et al., 2021). Fig. 1 visually represents the motivation of the proposed DGSSH framework. (2) Current deep remote sensing image hashing commonly employs random sampling or hardest sample mining (Shan, Liu, Gou, Zhou, & Wang, 2020; Sumbul, Ravanbakhsh, & Demir, 2022) to construct the training batches and ignores the distribution shift of selected samples, which ultimately leads to bad local minima (Yu, Liu, Gong, Ding, & Tao, 2018). For the foregoing reasons, the remote sensing image hashing capability has yet to be further improved.

To overcome the aforementioned dilemma, we put forward a novel deep hashing framework to strengthen the quality of the generating hash codes by exploiting global semantic information and multiscale features of remote sensing images, called Deep Global Semantic Structure-preserving Hashing (DGSSH). It is well known that our work is one of the pioneers to perform remote sensing image retrieval by introducing the Swin Transformer framework. Specifically, our proposed DGSSH framework principally contains feature representation module and hash learning module, which is portrayed in Fig. 2. For the feature representation module, the Swin Transformer mechanism is employed to obtain more discriminative fine-grained descriptors of remote sensing images by learning global contextual information and multiscale features. For hash learning, the corrective triplet loss and asymmetric constraint are jointly optimized to generate discriminated hash codes of remote sensing images.

Overall, the major contributions of this work can be summarized below:

(1) Feature extraction module based on Swin Transformer architecture is first introduced into remote sensing image retrieval to extract global contextual characteristics through modeling relationships among different blocks of input samples. Meanwhile, by exploring the layer-by-layer downsampling structure, the designed framework could capture multiscale feature descriptors of raw remote sensing samples.

(2) By introducing distribution matching restriction, the novel corrective triplet loss for deep hashing is developed to rectify the distribution shift of the selected samples, which could make selected triplets and all possible triplets share the similar distribution.

(3) To boost the training efficiency, the asymmetric strategy is introduced into the remote sensing space to explore the similarity preservation and learn discriminative discrete codes of query and database samples, respectively.

(4) Abundant experiments were executed on three open-available datasets to justify our listed contributions and show the DGSSH method is superior to other approaches in real-world implementation of remote sensing image retrieval.

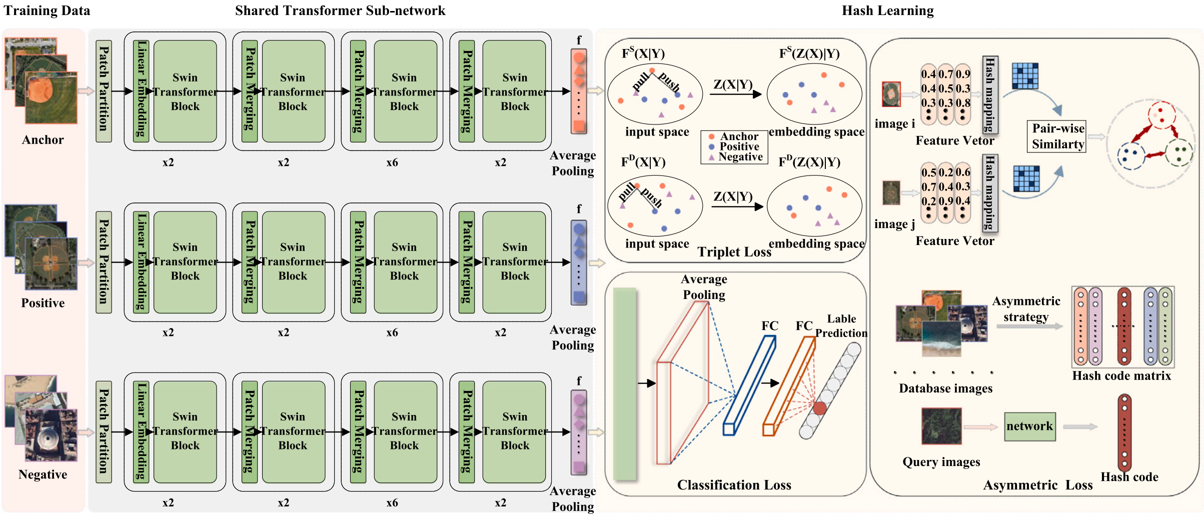

Fig. 2. An overview of the proposed Deep Global Semantic Structure-preserving Hashing Framework. The DGSSH framework mainly includes two parts: feature representation and hash learning. To better represent complex remote sensing semantic contents, the Swin Transformer framework is first developed to explore multiscale characteristics and global semantic information from the input remote sensing samples. The corrective triplet loss is brought in to reduce the distribution shift caused by the triplet sampling method, which constrains the data distribution of all triplet samples and selected triplet samples in a mini-batch to share the similar distribution. Besides, asymmetric loss and classification loss are also exploited jointly to optimize the model parameters for generating discriminative compact hash codes.

The remaining section of this paper is arranged in the following way. Section 2 describes several works relevant to the paper. In Section 3, the detailed work of the DGSSH is described. Section 4 demonstrates the robustness of the proposed DGSSH architecture through comprehensive experiments. Section 5 summarizes our conclusions and perspectives for the future.

2. Related work

2.1. Hash learning

The object of hash learning maps continuous and high-dimensional feature vectors into discrete Hamming space with significant discrimination. Because of its merits of taking less storage and conducting fast searches, hashing methods have absorbed the interest of many researchers at home and abroad. Especially the evolution of Deep Neural Networks is growing more and more sophisticated in the computer vision field, some researchers integrate hash learning and neural networks into a unified deep hashing network to achieve an encouraging retrieval system. Specifically, since hand-crafted visual features are unable to well express the deep semantic information of raw images, Xia et al. first introduce convolutional neural networks into a deep hashing learning framework, meanwhile, by designing the two-stage optimization strategy, Convolutional Neural Network Hashing (CNNH) is proposed for massive image search (Xia, Pan, Lai, Liu, & Yan, 2014). However, image representation cannot be fed back to the updating of discrete codes, the above framework does not fall into one end-to-end manner, leading to suboptimal binary codes. To this end, by developing the pairwise ranking loss, the Deep Pairwise-Supervised Hashing framework (DPSH) (Li, Wang, & Kang, 2015) is proposed for simultaneous feature representation and hash learning. To realize efficient and accurate retrieval of massive remote sensing images, the deep hash framework is induced into the remote sensing field and achieve outstanding property. To boost feature-searching efficiency, by employing neural networks to learn both feature extraction and hash mapping, a novel Deep Hashing Neural Networks architecture (DHNNs) is proposed for searching remote sensing images (Li, Zhang et al., 2017). To explore the semantic characteristics of remote sensing samples, the Deep Semantic Hashing (DSH) architecture is devised to provide a better representation of deep semantic characteristics (Chen et al., 2018). To alleviate overfitting problems, by replacing raw-pixel samples with intermediate representations of RS images, a novel Metric-Learning-based Deep Hashing Network, which is developed to achieve encouraging efficiency (Roy, Sangineto, Demir, & Sebe, 2020). Existing deep hashing-based RS image retrieval architectures usually employ the random sampling mechanism to build training batches, overlooking the sample distribution. To this end, by designing a probability sampling module to select training data, Shan et al. (2020) propose one novel deep hashing framework to explore more informative samples, achieving an excellent representation. By integrating accurate classification and fast retrieval into a unified framework, the Deep Hashing Convolutional Neural Network framework (DHCNN) (Song, Li, & Benediktsson, 2020) is introduced to the task of searching remote sensing images. As opposed to supervised learning, unsupervised learning frameworks have attracted increasing attention in the world of remote sensing, such as IRMFRCAMF (Li et al., 2016), DCAE (Tang, Zhang, Liu, & Jiao, 2018), pLSH (Fernandez-Beltran, Demir, Pla, & Plaza, 2020), and DUCH (Mikriukov, Ravanbakhsh, & Demir, 2022). Although the unsupervised learning framework makes no use of labeled signals within the process of training and seems to be more suitable for practical applications, they have poor retrieval accuracy compared to supervised hashing.

2.2. Transformer framework

Transformer architecture was put forward by Vaswani et al. (2017) for Natural Language Processing in 2017, which achieves decent performance just by using attention mechanisms and fully connected layers. Since then, numerous researchers have also demonstrated the availability of the transformer for computer vision in many works (Carion et al., 2020; Girdhar, Carreira, Doersch, & Zisserman, 2019). For example, in 2021, Dosovitskiy et al. (2020) present a Vision Transformer(ViT), which transports a sequence of patches of images into a simple transformer for image classification. TransHash is then proposed by Chen et al. (2022) for image-searching missions, which was the first application of a pure transformer to deep hashing. In El-Nouby et al. (2021), El-Nouby et al. apply Vision Transformer to category-based level retrieval and particular object retrieval. Tan, Yuan, and Ordonez (2021) introduce Reranking Transformers (RRTs), which make use of a few local descriptors combined with global features to determine the similarity of input sample pairs. Recently, Vision Transformer-based

Hashing (VTS) (Dubey et al., 2022) is presented by Dubey et al., which adopts pre-trained ViT as the major network architecture and a hash learning module is added. As ViT has evolved, researchers have also proposed variants of ViT. For illustration, Liu et al. (2021) propose the hierarchical Transformer, called the Swin Transformer, where the strategy of sliding windows makes it possible to link between windows. In this way, it reduces the complexity of the model and improves its efficiency. Based on the Swin transformer structure, by incorporating the process of mining global descriptors and mining local descriptors, SwinFGHash is introduced by Lu et al. (2021) for fine-grained retrieval applications.

Unlike prior studies (Jiang & Li, 2018; Peng et al., 2023; Song, Gao et al., 2022; Yu et al., 2018) to explore the complex nature of remote sensing images, our work is one of the pioneers in employing the Swin Transformer architecture to not only explore feature representations but also to learn hash mappings, realizing intra-class confusion and inter-class diversity. Additionally, based on distribution matching constraints, the corrective triplet loss for deep hashing is designed to reduce the distribution shift. Meanwhile, the asymmetric loss and classification loss are employed to produce the similarity-preserving and discriminative compact binary codes.

3. Methodology

For the DGSSH framework, a feature extraction module based on Swin Transformer is introduced to efficiently extract global semantic information and multiscale features of sophisticated remote sensing images. And the parameters of the model are optimized by well-designed losses to generate discriminative hash codes. For the data set D with N images $D = \{x_i | i = 1, 2, \dots\dots, N\}$ corresponding to the category label $Y = \{y_i | i = 1, 2, \dots\dots, N\}$, there are a total of C categories. The purpose of hash learning is to translate raw images into discrete hash codes with hash mapping $H\colon x_i \to B \in \{-1, +1\}^K$.

Fig. 2 depicts the DGSSH framework, which mainly contains the feature representation module (Fig. 2 left part) and the hash learning module (Fig. 2 right part). The feature extraction framework established by relying on the Swin Transformer architecture is introduced to obtain fine-grained features of remote sensing images by jointly learning global semantic information and multiscale features. For hash learning, the corrective triple constraint, asymmetric strategy and classification loss are jointly optimized to learn compact and discriminative binary codes.

3.1. Network architecture

Most deep hashing methods commonly utilize CNNs to extract rich semantic information from remote sensing images. However, the restriction of convolutional kernels makes CNNs could not adequately capture the global semantic information of images (Liu et al., 2021; Lu et al., 2021; Peng et al., 2023). Unlike CNNs, as the remote sensing images are sophisticated, a feature extraction framework built on the Swin Transformer structure is developed to gain more fine-grained features of remote sensing images, which could not only extract the global semantic information of the images but also take into account the multiscale characteristics.

The Swin Transformer network undergoes several stages to produce the various-sized feature representations. Specifically, the $H \times W \times 3$ remote sensing image is imported into the patch partition module, which is split into non-overlapping patches. The patch is also known as a token, each of which has a size of 4 × 4, so the token size turns into $\frac{H}{4} \times \frac{H}{4} \times 48$ by the partition method. Later on, the linear embedding layer is utilized to transform the size of the characteristic to another value (notation A), then we add the Swin Transformer Blocks with modified self-attention computation mechanism on these patches, which produces the $\frac{H}{4} \times \frac{H}{4} \times A$ feature representation. The above linear embedding layer and Swin Transformer Blocks constitute the first stage of the network. In order to generate a hierarchical structure, it is considered that the patch merging layer is utilized to decrease the count of tokens within the deeper and deeper network. Concretely, the patch merging layer connects each 2 × 2 neighboring patches, then maps the connected feature dimension into the half with the linear layer. In other words, the patch merging layer performs a downsampling operation to tune the feature map resolution as well as its channel number. Immediately after, the feature transformation is conducted with the Swin Transformer Block. This patch merging layer and feature transformation represent stage 2, which produces the $\frac{H}{8} \times \frac{W}{8} \times 2A$ sized feature map. This process is duplicated twice again, denoted as stage 3 and stage 4, these two stages' outputs are $\frac{H}{16} \times \frac{W}{16} \times 4A$ and $\frac{H}{32} \times \frac{W}{32} \times 8A$ respectively. Through these stages, one can both capture the global semantic information and explore the multiscale descriptors of images.

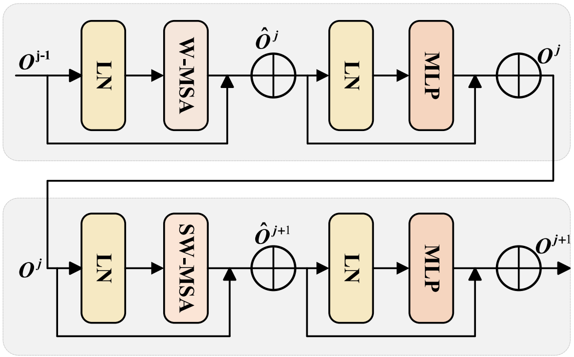

Fig. 3. Swin transformer block module.

3.2. Swin transformer block module

Essentially, the Swin Transformer block is a window-based multi-head self-attention module. As exhibited in Fig. 3, the Swin Transformer Block mainly is composed of three parts: the window-based multi-headed self-attention (W-MSA) module, the shifted window window-based multi-head self-attention mechanism (SW-MSA), and the Multilayer Perceptron (MLP). LayerNorm layer is also applied in front of W-MSA, SW-MSA, and MLP. The W-MSA performs self-attention calculations in a local window, yet this approach limits the interaction of information between windows and restricts its modeling capability. To enable window-to-window information interaction, the SW-MSA implements a shifted window strategy that allows neighboring windows to interact and explore global characteristics indirectly. On the basis of the above mechanism, the alternating Swin Transformer Block for image x_i can be defined as follows:

$$\hat{O}^j = W - MSA\left(LN\left(O^{j-1}\right)\right) + O^{j-1}$$
$$O^j = MLP\left(LN\left(\hat{O}^j\right)\right) + \hat{O}^j$$
$$\hat{O}^{j+1} = SW - MSA(LN(O^j)) + O^j$$
$$O^{j+1} = MLP(LN(\hat{O}^{j+1})) + \hat{O}^{j+1}$$

where j indicates the jth block, $\hat{O}$ indicates the W-MSA and the SW-MSA output, and O indicates the MLP output.

3.3. Hash code learning

3.3.1. Corrective triplet loss

Due to the complexity of image content, there are significant intra-class variations and low inter-class differences among remote sensing images. Usually, previous works employ a random sampling strategy or select the hardest triples to build the training samples, resulting in causes the distribution shift (Shan et al., 2020; Sumbul et al., 2022; Yu et al., 2018). In other words, in triplet learning, it is common for model to be trained with a few triples, but the distribution of the whole triples differs from the distribution of the selected triples. To minimize the distribution shift during triplet mining, relying on the distribution matching constraint, the corrective triplet loss for deep hashing is proposed to make the small portion of triples utilized for model training and the full set of triples have a similar distribution.

Let T_D stands all possible triplet samples, while T_S stands a small portion of triples selected from all triples, i.e., $T_S \in T_D$. In general, a small group of triple samples is selected to implement the triplet-based model training. We establish the training samples on the basis of a semi-hard triplet sampling strategy, which is characterized below: it establishes anchor-positive sample $\left(x_a, x_p\right)$ from the mini-batch then negative data x_n is randomly selected based on the constraint.

$$d_M^2\left(x_a, x_p\right) \leq d_M^2\left(x_a, x_n\right) \leq d_M^2\left(x_a, x_p\right) + \theta \tag{1}$$

where $\theta \geq 0$. d_M^2 is the Mahalanobis distance (Yu et al., 2018), which is widely used to measure the distance. Then, the triplet loss on T_S is given by.

$$\mathcal{L}_{triplet} = \sum_{(x_a, x_p, x_n) \in T_S} \left[d_M^2\left(x_a, x_p\right) - d_M^2\left(x_a, x_n\right) + \theta\right]_+ \tag{2}$$

However, T_S and T_D have different data distribution, i.e., there is a distribution shift. To measure the distribution shift, we define triplet-induced data $\hat{T}_S$ for T_S, i.e.,

$$\hat{T}_S = \left\{\left(x_a, y_a\right), \left(x_p, y_p\right), \left(x_n, y_n\right) | \forall \left(x_a, x_p, x_n\right) \in T_S\right\} \tag{3}$$

Likewise, $\hat{T}_D$ was created in the same manner. We measure the distribution shift in the triplet using the different distribution of $\hat{T}_S$ and $\hat{T}_D$. Let x be the input sample and $Z(X)$ represent the feature embedding learned from the network, i.e., a fixed-size feature vector. For the purpose of alleviating the distribution shift problem induced by the triple sampling method, the shared conditional invariant representation is learned in the triplet-induced data $\hat{T}_S$ and $\hat{T}_D$, so that all the triples and the selected triples have similar data distribution, i.e.,

$$F^S(Z(X)|Y) = F^D(Z(X)|Y) \tag{4}$$

where $F^S(x)$ and $F^D(x)$ refer to the probability density functions of $\hat{T}_S$ and $\hat{T}_D$ respectively. Usually, to assess the two distributions discrepancy, researchers have made extensive use of the Maximum Mean Discrepancy (MMD) (Huang, Gretton, Borgwardt, Schölkopf, & Smola, 2006) approach. However, it is possible to detect the discrepancy between $F^S(Z(X)|Y)$ and $F^D(Z(X)|Y)$ in the conditional mean feature embedding approach. Consequently, the distributed matching constraint loss is described below.

$$\mathcal{L}_{constraint} = \sum_y \left\|Z_y^S - Z_y^D\right\|_2^2 \tag{5}$$

where Z_y^S and Z_y^D denote the class-specific mean feature embedding of $\hat{T}_S$ and $\hat{T}_D$ respectively, i.e.,

$$Z_y^S = \sum_{(X, Y=y) \in \hat{T}_S} F^S(Z(X)|Y) * Z(X) \tag{6}$$

$$Z_y^D = \sum_{(X, Y=y) \in \hat{T}_D} F^D(Z(X)|Y) * Z(X) \tag{7}$$

The purpose of optimizing the triplet loss and distribution constraint matching loss by mining discriminative and conditional invariant feature representations is to deal with the distribution shift issue. The following is the definition of corrective triplet loss, which is listed below.

$$\mathcal{L}_{cor-triplet} = \mathcal{L}_{triplet} + \gamma * \mathcal{L}_{constraint} \tag{8}$$

where γ is the hyperparameter.

3.3.2. Asymmetric strategy

In general, supervised hashing methods utilize the symmetric strategy to generate binary codes for images, which increases the training cost and reduces the discriminative power of discrete hash codes (Jiang & Li, 2018; Song, Gao et al., 2022). With the aim of generating hash codes more efficiently, the asymmetric scheme is introduced into hash learning to strengthen hash code distinguishability.

To achieve effective retrieval, we should preserve the similarity between images, i.e., the hash codes corresponding to two semantically similar images should also be similar, which implies that the hash codes are relatively close together on the Hamming space; as the opposite, the hash codes of semantically dissimilar images are different, which is to say that they are also relatively distant on the Hamming space. It is often the case that pairwise sample similarity is measured by the inner product. In our work, the inner product is also utilized to measure pairwise similarity. Assuming the existence of a pair of image samples, one can get the conditional probability $p\left(s_{ij}|B\right)$ by means of the image's binary codes B and the similarity matrix.

$$\begin{aligned} p\left(s_{ij}|B\right) &= \begin{cases} \sigma(\Phi_{\mathrm{ij}}), s_{ij}=1 \\ 1-\sigma(\Phi_{ij}), s_{ij}=0 \end{cases} \\ &= \sigma\left(\Phi_{ij}\right)^{s_{ij}}\left(1-\sigma\left(\Phi_{ij}\right)\right)^{1-s_{ij}} \end{aligned} \tag{9}$$

While the value of s_{ij} is equal to 1 (or 0), it means that the two images are similar (or dissimilar). According to $d_H\left(b_i, b_j\right) = \frac{1}{2}\left(K - \left\langle b_i, b_j\right\rangle\right)$ (K indicates the number of bits of the hash code), one can notice that when both images' hash codes are close within the Hamming space, the corresponding inner product of the hash codes is also large and the conditional probability $p\left(s_{ij}|B\right)$ becomes $p(1|B)$, which also points to the fact that the two images x_i and x_j should be grouped into one class; vice versa, the conditional probability $p(0|B)$ is small, then the two images should not be classified into the same class. It is optimal to evaluate pairwise similarity with the weighted negative log-likelihood loss function.

$$\begin{aligned} \mathcal{L}_1 &= -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\log\left(p\left(s_{ij}|B\right)\right) \\ &= -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(s_{ij}\log\left(\sigma\left(\Phi_{ij}\right)\right)\right) \\ &+ \left(1-s_{ij}\right)\log\left(1-\sigma\left(\Phi_{ij}\right)\right) \end{aligned} \tag{10}$$

For Eq. (10), w_{ij} is introduced to alleviate the data pair imbalance. S_1 denotes the set of labeled image pairs with similar categories, and S_0 is the set of dissimilar sample pairs. $S_1 \in S$, $S_0 \in S$. w_{ij} is calculated as follows.

$$w_{ij} = \begin{cases} 1/\left|S_1\right|, s_{ij}=1 \\ 1/\left|S_0\right|, s_{ij}=0 \end{cases} \tag{11}$$

Since $\sigma(x) = \frac{1}{1+e^x}$ denotes the sigmoid function, $\mathcal{L}_1$ can be rewritten.

$$\begin{aligned} \mathcal{L}_2 &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(\log\left(1+e^{\Phi_{ij}}\right) - s_{ij}\Phi_{ij}\right) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(\log\left(1+e^{\frac{1}{2}b_i^T b_j}\right) - \frac{1}{2}s_{ij}b_i^T b_j\right) \end{aligned} \tag{12}$$

For Eq. (12), as it is hard to discrete optimization with binary code b, we use the output $\tilde{h}$ of the hash layer without b for accelerating the optimization speed. So Eq. (12) is reformulated as:

$$\mathcal{L}_3 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(\log\left(1+e^{\frac{1}{2}\tilde{h}_i^T\tilde{\mathrm{h}}_j}\right)\frac{1}{2}s_{ij}\tilde{h}_i^T\tilde{\mathrm{h}}_j\right) \tag{13}$$

By optimizing Eq. (13), the distance between samples that are of the same class is much smaller than the distance between samples that are not of the same class within Hamming space.

To reduce the cost of training, the asymmetric strategy is implemented to produce discriminative discrete hash codes for images.

Supposing we have randomly selected m data in the database, of which these samples constitute the query data, i.e., $Q = D^{\Omega}$. D^{Ω} indicates the data indexed by $\Omega = \{i_1, i_2, \ldots\ldots, i_m\}$ indicate the index of the sampled query samples and $\Psi = \{1, 2, 3, \ldots\ldots, n\}$ indicate the index of the database samples. As $\Omega \in \Psi$, the hash codes of a query sample q_k has two representations, either a tuple of the database hash matrix or a hash-like encoding learned through the hash layer. Because of the foregoing discussion, a constraint loss is imposed to ensure that both hash codes are as similar as possible, i.e.,

$$\mathcal{L}_4 = \sum_{i\in\Omega} \left(B_{q_k} - \tilde{h}_{q_k}\right)^2 \tag{14}$$

where B_{q_k} represents the element of the database matrix. $\tilde{h}_{q_k} = \tanh\left(W_h^T\left(\varphi\left(q_k\right);\Theta\right) + O_h\right)$ to be the hash layer output. φ is the network function, Θ to be the pre-trained model parameters, $W_\mathrm{h} \in \mathbb{R}^{K\times 768}$ and $O_\mathrm{h} \in \mathbb{R}^{K\times 1}$ stand for the weight matrix and bias vector of the hash layer respectively. The ultimate asymmetric loss can therefore be described as:

$$\mathcal{L}_{asymmetric} = \mathcal{L}_3 + \beta * \mathcal{L}_4 \tag{15}$$

where β is the hyper-parameter.

3.3.3. Classification loss

Previous research suggests that the semantic labeling of images is beneficial for producing compact binary codes (Li, Sun, He and Tan, 2017; Qin et al., 2022; Yang, Lin, & Chen, 2017). To enhance the discriminative capability, classification loss is introduced into remote sensing image retrieval to tune the obtained compact codes, defined as follows.

$$\mathcal{L}_{cla} = -\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) \tag{16}$$

where y_i indicates the image's semantic label information and $\hat{y}_i$ is the output of the classification layer, which is defined as $\hat{y}_i = soft\max\left(W_c^T \tilde{h}_i + V_c\right)$, $W_\mathrm{c} \in \mathbb{R}^{K\times C}$ and $V_\mathrm{c} \in \mathbb{R}^{C}$ represent the weight matrix and bias vector of the classification layer, respectively. The ultimate loss function is listed below.

$$\mathcal{L} = \mathcal{L}_{cor-triplet} + \mathcal{L}_{asymmetric} + \alpha * \mathcal{L}_{cla} \tag{17}$$

where α is the hyperparameter to trade off similarity information and semantic information. Corrective triplet loss addresses the distribution shift of the triplet sampling method, asymmetric loss maintains the similarity between images and the classification loss embeds label signals into the learning process of hashing, which further enhances the hash code discriminative. By optimizing Eq. (17), it is easy to generate the distinguished binary codes for fast and accurate image retrieval. As shown in Algorithm 1, we briefly describe the entire training process of the DGSSH framework.

3.4. Out-of-sample extension

After DGSSH is sufficiently trained, the trained network can generate binary codes for the query samples. To be specific, for any remote sensing image x_q, the deep feature embedding $f\left(x_q\right) = \mathbb{R}^{768\times 1}$ is captured once the image undergoes Swin Transformer network, and the hash codes of x_q can be produced by the following equation.

$$b_{x_q} = sign\left(\tanh\left(W_h^T f\left(x_q\right) + O_h\right)\right) \tag{18}$$

4. Experiments

4.1. Databases

The University of California, Merced dataset (UCMerced) (Yang & Newsam, 2010) is developed by the University of California, Merced. The image data information is extracted manually from the National Map of the United States Geological Survey (USGS). 21 classes of UCMerced are available, with 100 images in one class, so there are 2100 samples in total. Every image has 256 × 256 pixels, and the resolution of the pixels is one foot. A total of 1680 images are arbitrarily sampled to serve as the training samples, and the other samples are test samples.

Algorithm 1 The DGSSH learning algorithm

```
Require:
  Training samples $\mathcal{D} = \{x_i \mid i = 1, 2, \cdots, N\}$; Similarity Matrix $\mathcal{S}$;
  Binary codes length $K$; Hyper-parameters $\alpha$, $\beta$, $\gamma$.
Ensure:
  Output: Network Parameters $\Theta$ .
Initialize the parameters of Swin Transformer with weights obtained from the ImageNet.
for $epoch < max\_epoch$ do
    Randomly select $m$ samples from $Q = D^{\Omega}$ to build a mini-batch training set.
    Utilize pre-trained Swin Transformer framework to obtain the embedded representations and hash-like encodings by forward propagation.
    The ultimate loss is computed with Eq.(17).
    Update network parameters $\Theta$ by backpropagation.
end for
return The trained DGSSH model.
```

The aerial image dataset (AID) (Xia et al., 2017) collected from Google Earth, is a massive dataset constructed by Wuhan University in 2017. 10,000 samples are belonging to 30 classes, each of which has around 200–420 samples, and each sample is 600 × 600 in size. For the training part, the 5000 images are randomly picked and the remaining samples are the test subsets.

The WHU-RS19 dataset (WHURS) (Xia et al., 2009) is made by Wuhan University, whose data are also obtained from Google Earth. The dataset has 19 categories with about 50 images in each category, resulting in 1005 images in total. The image has a pixel value of 600 × 600, and the pixel resolution is up to 0.5 m. For it, the 809 images in every category are arbitrarily chosen as training samples, while the 196 remaining samples constitute the test set.

4.2. Experimental details

As proof of the validity and advantage of the DGSSH framework, we have done abundant comparison experiments between the DGSSH methods and several deep hashing methods, which include DPSH (Li et al., 2015), ADSH (Jiang & Li, 2018), DHCNN (Song et al., 2020), FAH (Liu et al., 2020), HHF (Xu, Chai, Xu, Li et al., 2022), AHCL (Song, Gao et al., 2022), HyP^2 Loss (Xu, Chai, Xu, Yuan et al., 2022), SWTH (Peng et al., 2023). Among these methods, except SWTH, all of them adopt the traditional Convolutional Neural Networks to explore the feature descriptors of remote sensing images. For the mentioned framework, the image pixel values are reset to 224 × 224, while the parameter values of the comparison method are the same as those of the initial paper. For the DGSSH framework, the Swin Transformer architecture which is a pre-trained model is deployed as a part of feature extraction. There are four stages in the Swin Transformer structure, each of which has a distinct number of blocks, i.e., 2, 2, 6, 2, respectively. These four stages will produce various feature representations, which are 56 × 56 × 96, 28 × 28 × 192, 14 × 14 × 384, 7 × 7 × 768, while the window size is set up as 7. In the training phase, AdamW is employed as an optimizer for the model, and the values of the learning rate, weight decay and batch size are fixed to 2×10^{-5}, 10^{-8} and 32 respectively. For the hyperparameters used in the model, we set α to 20, β to 200, and γ to 1. The algorithm for the DGSSH method is implemented with the PyTorch framework and all experiments are carried out on a machine with NVIDIA GeForce RTX 3090 GPU.

Table 1
Comparison of mAP results for different baselines with 16 bits, 32 bits, 48 bits, 64 bits, 128 bits on the UCMerced dataset.

Methods	16 bits	32 bits	48 bits	64 bits	128 bits
DPSH (Li et al., 2015)	0.8471	0.9313	0.9381	0.9576	0.9552
ADSH (Jiang & Li, 2018)	0.9247	0.9356	0.9554	0.9525	0.9575
DHCNN (Song et al., 2020)	0.9578	0.9605	0.9563	0.9683	0.9383
FAH (Liu et al., 2020)	0.9595	0.9608	0.9680	0.9702	0.9564
HHF (Xu, Chai, Xu, Li et al., 2022)	**0.9654**	0.9778	0.9749	0.9809	0.9763
AHCL (Song, Gao et al., 2022)	0.9599	0.9767	0.9781	0.9785	0.9808
HyP^2 Loss (Xu, Chai, Xu, Yuan et al., 2022)	0.9443	0.9774	0.9740	0.9738	0.9748
SWTH (Peng et al., 2023)	0.8273	0.8401	0.8675	0.8746	0.8858
DGSSH	0.9645	**0.9821**	**0.9872**	**0.9860**	**0.9858**

Table 2
Comparison of mAP results for different baselines with 16 bits, 32 bits, 48 bits, 64 bits, 128 bits on the AID dataset.

Methods	16 bits	32 bits	48 bits	64 bits	128 bits
DPSH (Li et al., 2015)	0.5797	0.7843	0.8082	0.8255	0.6025
ADSH (Jiang & Li, 2018)	0.8671	0.9306	0.9370	0.9365	0.9356
DHCNN (Song et al., 2020)	0.8667	0.9040	0.9076	0.9103	0.9145
FAH (Liu et al., 2020)	**0.9359**	0.9236	0.9485	0.9593	0.9280
HHF (Xu, Chai, Xu, Li et al., 2022)	0.9338	0.9418	0.9415	0.9397	0.9413
AHCL (Song, Gao et al., 2022)	0.9100	0.9019	0.9407	0.9478	0.9572
HyP^2 Loss (Xu, Chai, Xu, Yuan et al., 2022)	0.8702	0.9253	0.9354	0.9349	0.9405
SWTH (Peng et al., 2023)	0.7890	0.8685	0.8803	0.8708	0.8940
DGSSH	0.9317	**0.9531**	**0.9642**	**0.9661**	**0.9674**

Table 3
Comparison of mAP results for different baselines with 16 bits, 32 bits, 48 bits, 64 bits, 128 bits on the WHURS dataset.

Methods	16 bits	32 bits	48 bits	64 bits	128 bits
DPSH (Li et al., 2015)	0.9141	0.9562	0.9381	0.9587	0.9120
ADSH (Jiang & Li, 2018)	0.9526	0.9665	0.9735	0.9564	0.9756
DHCNN (Song et al., 2020)	0.9331	0.9552	0.9564	0.9321	0.8862
FAH (Liu et al., 2020)	0.9671	0.9587	0.9648	0.9548	0.9655
HHF (Xu, Chai, Xu, Li et al., 2022)	0.9850	0.9905	0.9862	0.9856	0.9925
AHCL (Song, Gao et al., 2022)	0.9693	0.9955	0.9866	0.9930	0.9912
HyP^2 Loss (Xu, Chai, Xu, Yuan et al., 2022)	0.9774	0.9824	0.9910	0.9936	0.9901
SWTH (Peng et al., 2023)	0.8548	0.8625	0.8631	0.9165	0.9267
DGSSH	**0.9866**	**0.9962**	**0.9984**	**0.9964**	**0.9946**

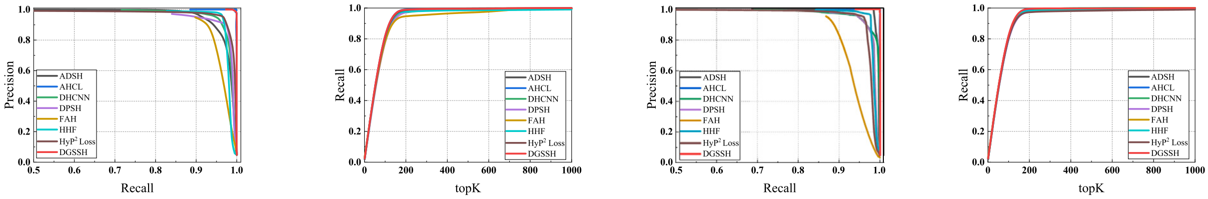

Fig. 4. The precision-recall curves and topK-Recall curves results on UCMerced w.r.t.32 bits and 64 bits.

4.3. Evaluation criteria

With the aim of validating the reliability of the DGSSH, there are three common criteria implemented to assess the quality of retrieval, including Mean Average Precision (mAP), Precision-Recall (PR) curves and topK-Recall curves. A more commonly utilized indicator to measure performance is mAP. For the given query sample, the model could rank the images of the database according to the Hamming distance. The sorted list of the query images could be obtained, and based on the ranking results, the average precision (AP) can be calculated. In fact, for a fixed number of query sets, mAP denotes the average of their APs. Precision-Recall (PR) curves help us to estimate the models' performance as a whole, and in general, the value of PR is proportional to the performance of the model. The topK-Recall curves calculate the proportion of samples that are similar to the query image among the returned topK retrieval results to the total number of samples. In the experiments, for the UCMerced and AID datasets, we set K to 1000, while K is set to 500 for WHURS.

4.4. Performance comparison of different methods

Comprehensive experiments and comparisons are conducted on three benchmark datasets. First, the full methods mAP values were calculated in the case of various hash lengths, followed by the Precision-Recall (PR) curves and topK-Recall curves for each method are plotted against 32 bits and 64 bits hash codes, which can prove the superiority of the DGSSH framework. Tables 1, 2, and 3 display the mAP results on the UCMerced, AID, and WHURS databases respectively. From the figures in the tables, it is revealed that the DGSSH method realizes promising results on the three publicly available datasets, which is attributed to the fact that we fully explore the global contextual information and multiscale characteristics of remote sensing images and the designed loss is optimized to generate distinguishing binary codes for enhanced retrieval accuracy. For the UCMerced dataset, the proposed method reaches up to 0.9872 mAP for 48 bits hash codes. For 16 bits hash codes, a competitive phenomenon emerges between the DGSSH method and the HHF method, yet on most of the hash bits, our results

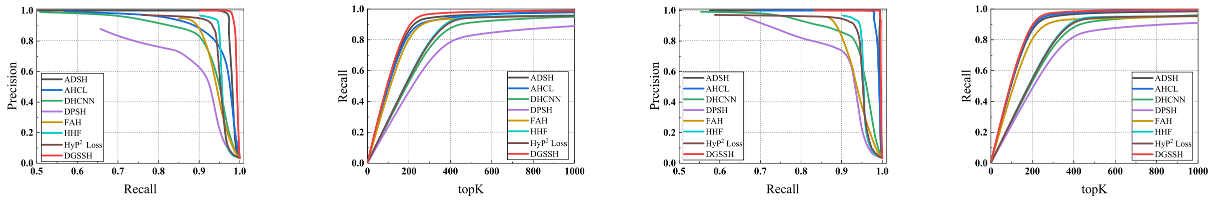

Fig. 5. The precision-recall curves and topK-Recall curves results on AID w.r.t.32 bits and 64 bits.

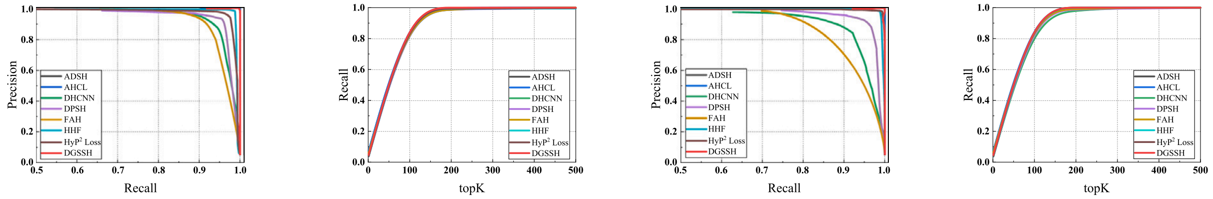

Fig. 6. The precision-recall curves and topK-Recall curves results on WHURS w.r.t. 32 bits and 64 bits.

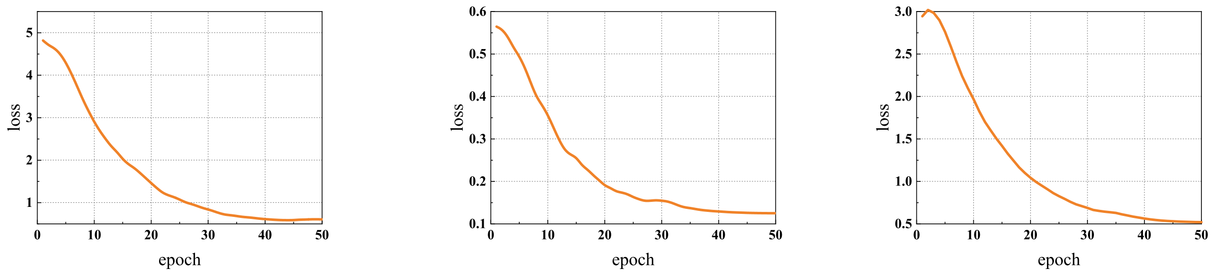

Fig. 7. The result of loss curve on the UCMerced, AID, and WHURS datasets w.r.t. 16 bits.

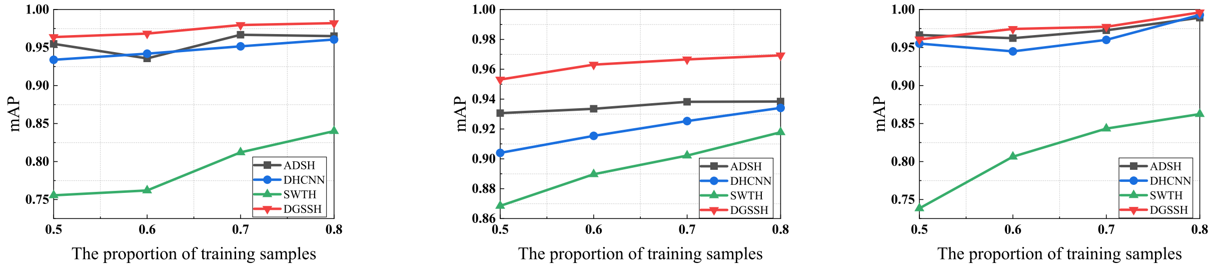

Fig. 8. The mAP results of different numbers of training samples on UCMerced, AID, and WHURS datasets w.r.t 32 bits.

still outperform the HHF method. Compared with the FAH method, except for the 16 bits hash codes, the DGSSH framework improves 2.95%, 1.57%, 0.68%, and 3.94% with different hash lengths on the AID dataset, and has promising results on the UCMerced dataset and WHURS dataset. The mAP results in Table 3 confirm that the proposed framework beats other deep hashing methods. Moreover, in comparison with the SWTH method, the DGSSH method has at least 10%, 7.34%, and 6.79% improvement on UCMerced, AID, and WHURS respectively. Besides the mAP metrics, Figs. 4, 5, and 6 present the Precision-Recall (PR) curves and topK-Recall curves on the UCMerced, AID, and WHURS benchmarks respectively. These results state that the DGSSH framework realizes desirable retrieval performance over other methods under most situations, which further illustrates the effectiveness of DGSSH. The loss curves of the model are also drawn against the 16 bits hash codes on three benchmarks, as illustrated in Fig. 7, where the loss gradually converges as the epoch increases.

4.5. Performance comparison of different proportions

In this section, with different training samples' proportions, several experiments are conducted to verify the feasibility and effectiveness of our designed DGSSH framework. Figs. 8 and 9 describe the detailed experimental results on the UCMerced, AID, and WHURS w.r.t 32 bits hash codes.

Firstly, for the experiments, there is a comparison between the DGSSH framework and different approaches on three datasets, consisting of ADSH, DHCNN, and SWTH methods, in which the percentage of training samples to the total samples is increased by 10% starting from 50% to 80%. As the results in Fig. 8, it can be witnessed that the SWTH method is more sensitive to the different training samples and the mAP value of the model consistently grows as the percentage of training samples gets higher compared to the other three methods. Besides, both the ADSH framework and the DHCNN framework yield competitive

Table 4
Accuracy results of different triplet loss functions on UCMerced, AID, and WHURS datasets..

Triplet loss	UCMerced			AID			WHURS		
	10epoch	30epoch	50epoch	10epoch	30epoch	50epoch	10epoch	30epoch	50epoch
Random triplet loss	0.8619	0.9548	0.9714	0.7626	0.8544	0.8678	0.9031	0.9847	0.9898
Hardest triplet loss	**0.8810**	0.9714	0.9786	0.8334	0.9382	0.9452	0.9082	0.9847	0.9847
Corrective triplet loss	0.8738	**0.9762**	**0.9786**	**0.8426**	**0.9482**	**0.9552**	**0.9133**	**0.9847**	**0.9949**

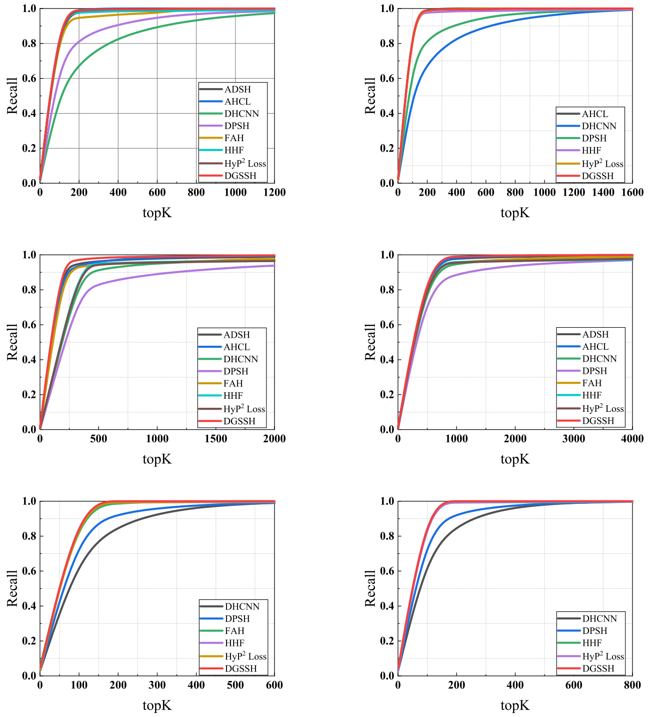

Fig. 9. TopK-Recall curve results on the UCMerced, AID, and WHURS datasets w.r.t. 32 bits.

performance on the UCMerced and WHURS datasets, yet the proposed DGSSH method has relatively high mAP values under various samples' proportions. Secondly, to further prove the performance of the DGSSH deep hashing framework, we draw the more detailed topK-Recall curves with various numbers of returned samples, as shown in Fig. 9. We perform topK-Recall curves with the top 60% and top 80% data on the UCMerced and WHURS datasets and plot the top 20% and top 40% curves on the AID dataset. From the related experiments, it seems clear that the DGSSH method achieves better performance on the three datasets under various lengths of returned samples. Based on the above-mentioned analysis, our proposed DGSSH framework achieves superiority over several state-of-the-art retrieval methods.

4.6. Parameter analysis

α is the hyperparameter that trades off the similarity loss and semantic loss, and β is the hyperparameter that controls the similarity of the two hash codes of the query image in the asymmetric loss. For the two hyperparameters α and β, parameter sensitivity experiments were executed on the UCMerced, AID, and WHURS benchmarks, in which $\alpha = \{0, 10, 20, 30, 40\}$, $\beta = \{0, 100, 200, 300, 400\}$. Fig. 10 reveals the mAP results for different settings of α and β with 48 bits hash codes. From Fig. 10, it is noticed that the best mAP results can be obtained when $\alpha = 20$ and $\beta = 200$. γ is the hyperparameter that controls the data distribution. Likewise, we carried out relevant parametric experiments on three benchmarks and set $\gamma = \{0, 0.5, 1.0, 1.5, 2.0\}$. Fig. 11 shows the influences of different values of γ on mAP under 32 bits hash codes. Fig. 11 shows that mAP reaches its maximum value when $\gamma = 1.0$ for the UCMerced and AID datasets, while for the WHURS, although mAP does not reach its maximum value when $\gamma = 1.0$, it is close to the maximum value. Therefore we set γ to 1.0 for the three benchmarks.

4.7. Training time analysis

To compare the training time of different models, we evaluate the time cost of several frameworks on UCMerced, AID, and WHURS datasets, including DHCNN, AHCL, HHF, SWTH, and DGSSH-Sym (which uses the symmetric approach). Comparative results of training overhead are displayed in Fig. 12. Since the Swin Transformer structure is utilized to explore the deep characteristics of the remote sensing image, the training time of our proposed DGSSH is longer than deep CNN-based hashing frameworks, but the performance of retrieval is not affected by the complexity of the model. What is more, to prove that the asymmetric strategy could effectively decrease the training complexity, we also evaluate the training costs of DGSSH, DGSSH-Sym, and SWTH framework on UCMerced, AID, and WHURS datasets, as illustrated in Fig. 12. By the comparison of experiment results, it can be clearly noticed that the asymmetric strategy decreases the time expense of training. This is mainly because we only employ the asymmetric strategy to learn the hash mapping and obtain the hash encoding for query samples. And, for the database images, the binary encoding is learned straightforwardly by optimizing the overall loss. It is distinct from the symmetric strategy of the DGSSH-Sym and the SWTH, which could binarize the output of the network to produce the hash code of the query samples and the database data. However, as data volumes grow, the training overhead will increase as well. Overall, the asymmetric mechanism could improve training productivity and boost the discriminating strength of binary codes.

4.8. Ablation study

4.8.1. Validity of corrective triplet loss

To demonstrate that the corrective triplet loss can address the distribution shift problem in the triplet selection method, ablation experiments were performed. In the training phase, it is designed to train the model by exploiting Random Triplet Loss, Hardest Triplet Loss, and Corrective Triplet Loss respectively. We compare the accuracy of the models on the three datasets with 32 bits hash codes. The curve in Fig. 13 reveals that the accuracy of the model attained with the corrective triplet loss has a higher value than the accuracy of the random triplet loss, at the same time, it also appears here that the corrective triplet loss competes with the hardest triplet loss. Table 4 displays the accuracy of 10epoch, 30epoch, and 50epoch on the three datasets for further result analysis, which proves the corrective triplet loss has desirable accuracy performance. Beyond that, to further illustrate the advantage of corrective triplet loss, for the extracted features, we mapped them into the 2D space through the t-SNE on the UCMerced. As shown in Fig. 14, it is noticeable that the embedding representations obtained with the corrective triplet have clear distinctions among categories, whereas the feature representations explored using Random Triplet Loss, Hardest Triplet Loss may not have clear edges between certain classes. It can explain that the corrective triplet loss makes it possible to achieve intra-class confusability and inter-class diversity of remote sensing images.

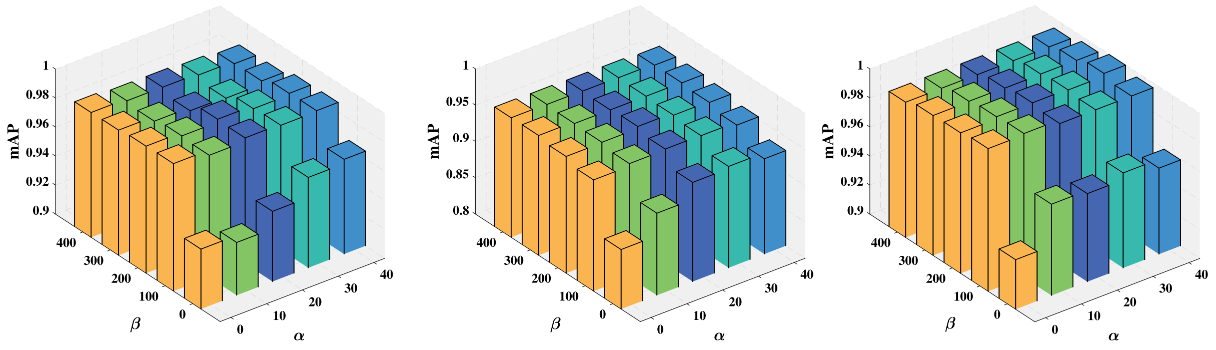

Fig. 10. The mAP results for different settings of α and β on the UCMerced, AID, and WHURS datasets with 48 bits hash codes.

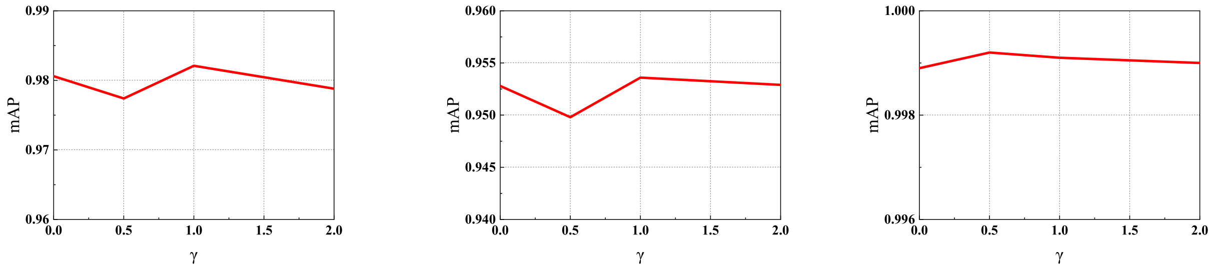

Fig. 11. The impact of parameter γ for the map on the UCMerced, AID, and WHURS datasets w.r.t.32 bits.

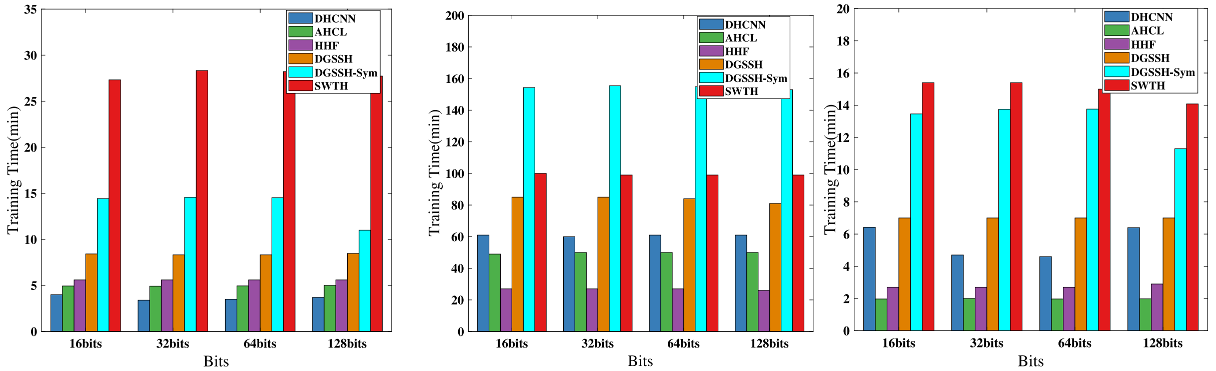

Fig. 12. Comparison of training time of different methods on UCMerced, AID, and WHURS datasets.

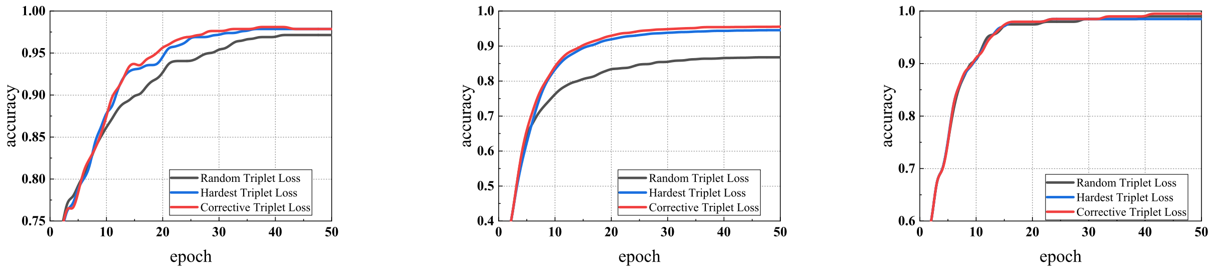

Fig. 13. The results of the accuracy curves of Random Triplet Loss, Hardest Triplet Loss, and Corrective Triplet Loss on UCMerced, AID, and WHURS datasets.

Table 5
The mAP results of the DGSSH combining different network structures and symmetric strategy.

Method	UCMerced			AID			WHURS		
	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
DGSSH-Alexnet	0.8440	0.8992	0.9264	0.7005	0.8520	0.8844	0.9140	0.9216	0.9356
DGSSH-ViT	0.9609	0.9808	0.9805	0.9540	0.9599	0.9621	0.9140	0.9477	0.9598
DGSSH-Sym	0.6807	0.8162	0.9125	0.4269	0.5380	0.6430	0.3900	0.6057	0.7945
DGSSH	**0.9645**	**0.9821**	**0.9860**	**0.9317**	**0.9531**	**0.9661**	**0.9866**	**0.9962**	**0.9964**

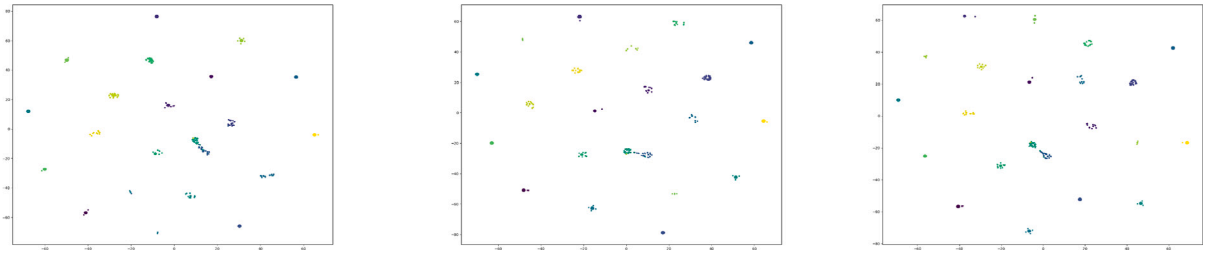

Fig. 14. The visualization of Random Triplet Loss, Hardest Triplet Loss, and Corrective Triplet Loss on UCMerced dataset with 32 bits hash codes.

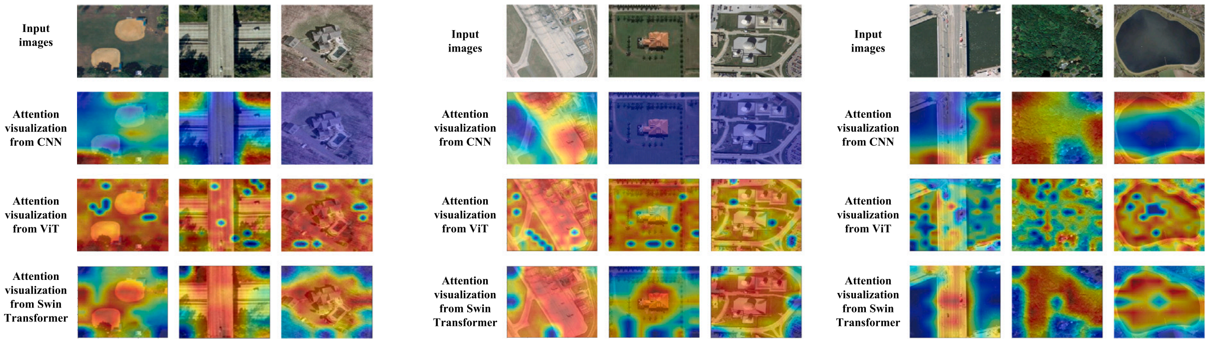
Input images
Attention visualization from CNN
Attention visualization from ViT
Attention visualization from Swin Transformer
Input images
Attention visualization from CNN
Attention visualization from ViT
Attention visualization from Swin Transformer
Input images
Attention visualization from CNN
Attention visualization from ViT
Attention visualization from Swin Transformer

Fig. 15. The result of attention visualization on UCMerced, AID, and WHURS datasets.

4.8.2. Advantages of extracting feature networks

In the DGSSH framework, the designed feature extraction part, which is built on the Swin Transformer architecture, could explore the global semantic features and multiscale information of remote sensing images well. To validate the Swin Transformer feature extraction reliability, it is worthwhile to compare the mAP results between the DGSSH method and traditional Convolutional Neural Networks framework and ViT-based framework. In the experiment, we exploit AlexNet and pretrained ViT as feature extraction to obtain the deep features of the image. Table 5 shows the values of mAP for the variant methods related to DGSSH. Apparently, the DGSSH method is superior to the two variants. Furthermore, to further confirm the outstanding power of the Swin Transformer framework, it is also necessary to visualize the sample attention map on three datasets. As is observed in Fig. 15, the examples of visualization utilizing Convolutional Neural Networks are mainly focused on local features, which is probably the reason that traditional Convolutional Neural Networks exploit the convolutional kernel with a fixed size to extract characteristics. Therefore, the global information of the image is not sufficiently exploited and ultimately affects performance improvement. Nevertheless, ViT-based methods can obtain more discriminative features of the image by establishing long-range dependencies. It is possible that ViT explores global contextual features containing complex and useless information due to the target-independent background information in remote sensing images, which also hinders the feature extraction capability. In comparison to traditional Convolutional Neural Networks and ViT, Swin Transformer-based frameworks can provide greater responses toward targets in images with complex background information by learning global contextual features and multiscale information to extract fine-grained characteristics.

4.9. Top-10 retrieval result

For the DGSSH method, we return the top-10 images of the query samples on three benchmarks. In Fig. 16, the green boxes indicate samples with the same labels as the query data and the red boxes indicate samples with different labels from the query data. In this work, the feature extraction module, which is built on the Swin Transformer architecture, is applied to learn the global semantic characteristics and multiscale information. Besides, the model parameters are optimized with a designed loss function to guarantee that the model is capable of retrieving similar samples to the query images. It is evident from the retrieval results in Fig. 16 that the DGSSH deep hashing has approach acceptable retrieval performance, which further clarifies its remarkable capability.

5. Conclusions and future work

In our work, we present the supervised deep hashing method called the Deep Global Semantic Structure-preserving Hashing framework via corrective triplet loss (DGSSH) for remote sensing image retrieval. DGSSH contains two modules: feature representation and hash learning. For the feature representation module, the Swin Transformer structure is employed to explore fine-grained features of remote sensing images by jointly learning global contextual information and multiscale features. In hash learning, the corrective triplet loss is utilized to address the distribution shift caused by random triplet sampling or the hardest triplet sampling. We introduce distribution matching constraints to make the distribution of all possible triplet samples and the selected triplet samples in a mini-batch as similar as possible. What is more, we make use of asymmetric learning to cut down the time overhead of training and in turn strengthen the discriminative power of the hash codes. Numerous experiments have been executed on UCMerced, AID, and WHURS benchmarks demonstrating the excellence of the DGSSH framework.

From a future perspective, we intend to apply DGSSH to the cross-modal remote sensing image retrieval scenario; and it is also our expectation to use unlabeled data to train DGSSH for unsupervised remote sensing image retrieval.

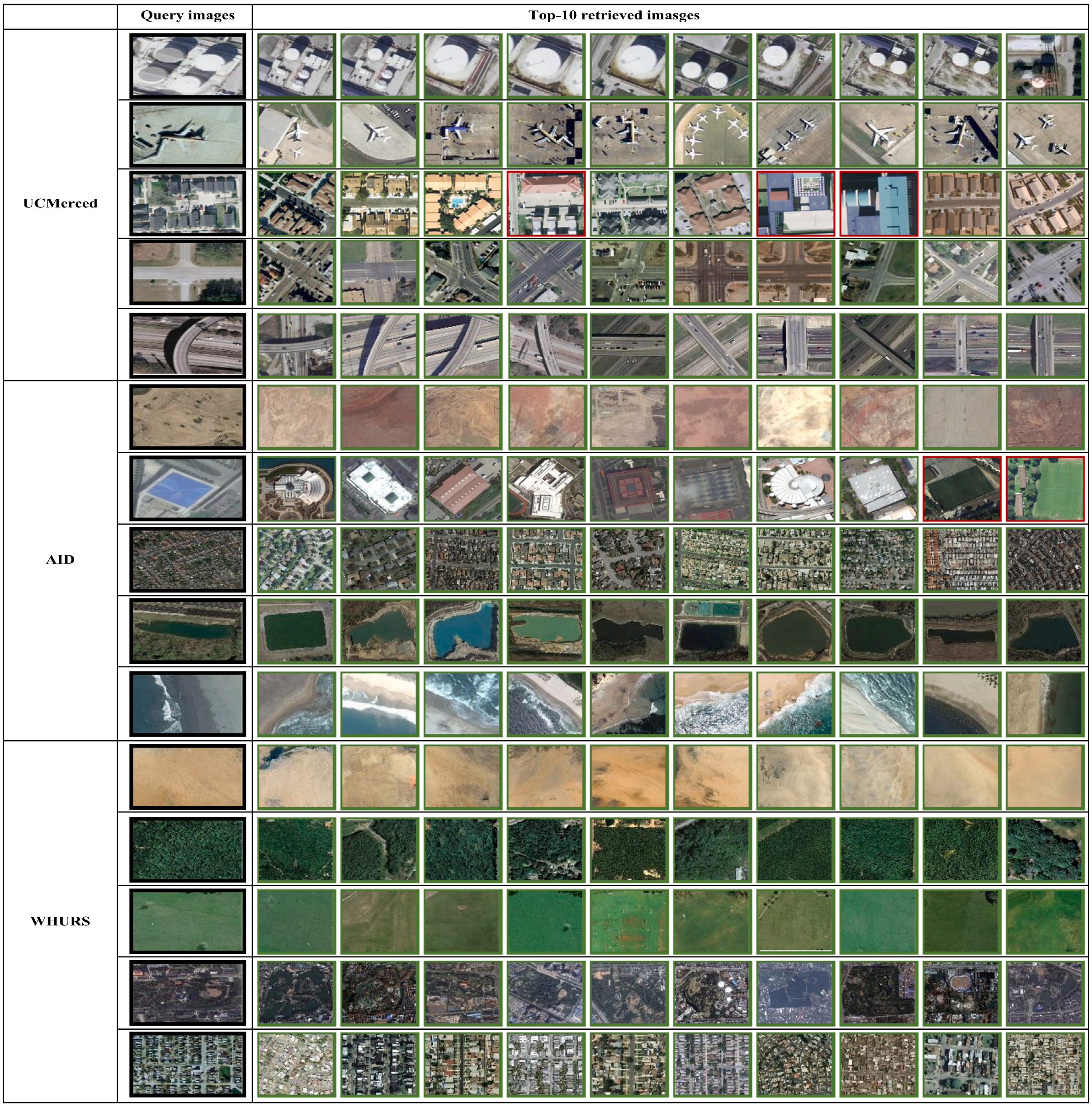

Fig. 16. The Top-10 images of the DGSSH method are shown on three baselines with 64 bits hash codes. Samples with green lines indicate positive samples retrieved; samples with red lines indicate negative samples retrieved.

CRediT authorship contribution statement

Hongyan Zhou: Identify ideas, Experiment, Analyze data, Write-revise. **Qibing Qin:** Overall comprehension and evaluation of the paper, Put forward revision suggestions for the paper. **Jinkui Hou:** Make corrections to the paper. **Jiangyan Dai:** Make corrections to the paper. **Lei Huang:** Propose revisions to the paper. **Wenfeng Zhang:** Propose revisions to the paper.

Declaration of competing interest

We declare that there is no conflict of interest between the authors in this work. We declare that there are no commercial or related conflicts of interest in this work.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62006174), Shandong Provincial Natural Science Foundation, China (ZR2022QF046), Science and Technology Research Program of Chongqing Municipal Education Commission, China under Grant KJQN202200551.

References

Alizadeh, S. M., Helfroush, M. S., & Müller, H. (2023). A novel siamese deep hashing model for histopathology image retrieval. *Expert Systems with Applications*, *225*, Article 120169.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, august 23–28, 2020, proceedings, Part I 16* (pp. 213–229). Springer.

Chen, Y., Zhang, S., Liu, F., Chang, Z., Ye, M., & Qi, Z. (2022). Transhash: Transformer-based hamming hashing for efficient image retrieval. In *Proceedings of the 2022 international conference on multimedia retrieval* (pp. 127–136).

Chen, C., Zou, H., Shao, N., Sun, J., & Qin, X. (2018). Deep semantic hashing retrieval of remotec sensing images. In *IGARSS 2018-2018 IEEE international geoscience and remote sensing symposium* (pp. 1124–1127). IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Dubey, S. R., Singh, S. K., & Chu, W.-T. (2022). Vision transformer hashing for image retrieval. In *2022 IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). IEEE.

El-Nouby, A., Neverova, N., Laptev, I., & Jégou, H. (2021). Training vision transformers for image retrieval. arXiv preprint arXiv:2102.05644.

Fernandez-Beltran, R., Demir, B., Pla, F., & Plaza, A. (2020). Unsupervised remote sensing image retrieval using probabilistic latent semantic hashing. *IEEE Geoscience and Remote Sensing Letters*, *18*(2), 256–260.

Girdhar, R., Carreira, J., Doersch, C., & Zisserman, A. (2019). Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 244–253).

Gong, Y., Lazebnik, S., Gordo, A., & Perronnin, F. (2012). Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(12), 2916–2929.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., & Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, *19*.

Jiang, Q.-Y., & Li, W.-J. (2018). Asymmetric deep supervised hashing. *vol. 32*, In *Proceedings of the AAAI conference on artificial intelligence*.

Kulis, B., & Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th international conference on computer vision* (pp. 2130–2137). IEEE.

Li, Q., Sun, Z., He, R., & Tan, T. (2017). Deep supervised discrete hashing. *Advances in neural information processing systems*, *30*.

Li, W.-J., Wang, S., & Kang, W.-C. (2015). Feature learning based deep supervised hashing with pairwise labels. arXiv preprint arXiv:1511.03855.

Li, Y., Zhang, Y., Huang, X., Zhu, H., & Ma, J. (2017). Large-scale remote sensing image retrieval by deep hashing neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(2), 950–965.

Li, T., Zhang, Z., Pei, L., & Gan, Y. (2022). HashFormer: Vision transformer based deep hashing for image retrieval. *IEEE Signal Processing Letters*, *29*, 827–831.

Li, Y., Zhang, Y., Tao, C., & Zhu, H. (2016). Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion. *Remote Sensing*, *8*(9), 709.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Liu, C., Ma, J., Tang, X., Liu, F., Zhang, X., & Jiao, L. (2020). Deep hash learning for remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(4), 3420–3443.

Lu, D., Wang, J., Zeng, Z., Chen, B., Wu, S., & Xia, S.-T. (2021). SwinFGHash: Fine-grained image retrieval via transformer-based hashing network. In *Proceedings of British machine vision conference* (pp. 1–13).

Lu, X., Zhu, L., Cheng, Z., Li, J., Nie, X., & Zhang, H. (2019). Flexible online multi-modal hashing for large-scale multimedia retrieval. In *Proceedings of the 27th ACM international conference on multimedia* (pp. 1129–1137).

Mikriukov, G., Ravanbakhsh, M., & Demir, B. (2022). Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing. arXiv preprint arXiv:2201.08125.

Peng, L., Qian, J., Wang, C., Liu, B., & Dong, Y. (2023). Swin transformer-based supervised hashing. *Applied Intelligence*, 1–13.

Qin, Q., Huang, L., Wei, Z., Nie, J., Xie, K., & Hou, J. (2021). Unsupervised deep quadruplet hashing with isometric quantization for image retrieval. *Information Sciences*, *567*, 116–130.

Qin, Q., Huang, L., Wei, Z., Xie, K., & Zhang, W. (2020). Unsupervised deep multi-similarity hashing with semantic structure for image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(7), 2852–2865.

Qin, Q., Xian, L., Xie, K., Zhang, W., Liu, Y., Dai, J., et al. (2022). Deep multi-similarity hashing with semantic-aware preservation for multi-label image retrieval. *Expert Systems with Applications*, *205*, Article 117674. http://dx.doi.org/10.1016/j.eswa.2022.117674.

Roy, S., Sangineto, E., Demir, B., & Sebe, N. (2020). Metric-learning-based deep hashing network for content-based retrieval of remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, *18*(2), 226–230.

Shan, X., Liu, P., Gou, G., Zhou, Q., & Wang, Z. (2020). Deep hash remote sensing image retrieval with hard probability sampling. *Remote Sensing*, *12*(17), 2789.

Song, W., Gao, Z., Dian, R., Ghamisi, P., Zhang, Y., & Benediktsson, J. A. (2022). Asymmetric hash code learning for remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–14.

Song, W., Li, S., & Benediktsson, J. A. (2020). Deep hashing learning for visual and semantic retrieval of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(11), 9661–9672.

Song, B. D., Park, H., & Park, K. (2022). Toward flexible and persistent UAV service: Multi-period and multi-objective system design with task assignment for disaster management. *Expert Systems with Applications*, *206*, Article 117855.

Sumbul, G., Ravanbakhsh, M., & Demir, B. (2022). Informative and representative triplet selection for multilabel remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1–11. http://dx.doi.org/10.1109/TGRS.2021.3124326.

Tan, X., Jiao, J., Zhong, Y., Ma, A., Xu, Y., Sha, Z., et al. (2022). The CNRIEEEMC: A communication-navigation-remote sensing-integrated ecological environment emergency monitoring chain for tailings areas. *International Journal of Applied Earth Observation and Geoinformation*, *108*, Article 102710.

Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781–10790).

Tan, F., Yuan, J., & Ordonez, V. (2021). Instance-level image retrieval using reranking transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 12105–12115).

Tang, X., Liu, C., Ma, J., Zhang, X., Liu, F., & Jiao, L. (2019). Large-scale remote sensing image retrieval based on semi-supervised adversarial hashing. *Remote Sensing*, *11*(17), 2055.

Tang, X., Liu, C., Zhang, X., Ma, J., Jiao, C., & Jiao, L. (2019). Remote sensing image retrieval based on semi-supervised deep hashing learning. In *IGARSS 2019-2019 IEEE international geoscience and remote sensing symposium* (pp. 879–882). IEEE.

Tang, X., Zhang, X., Liu, F., & Jiao, L. (2018). Unsupervised deep feature learning for remote sensing image retrieval. *Remote Sensing*, *10*(8), 1243.

Tong, X.-Y., Xia, G.-S., Hu, F., Zhong, Y., Datcu, M., & Zhang, L. (2019). Exploiting deep features for remote sensing image retrieval: A systematic investigation. *IEEE Transactions on Big Data*, *6*(3), 507–521.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Weiss, Y., Torralba, A., & Fergus, R. (2008). Spectral hashing. *Advances in neural information processing systems*, *21*.

Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., et al. (2017). AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), 3965–3981.

Xia, R., Pan, Y., Lai, H., Liu, C., & Yan, S. (2014). Supervised hashing for image retrieval via image representation learning. *vol. 28*, In *Proceedings of the AAAI conference on artificial intelligence*.

Xia, G.-S., Yang, W., Delon, J., Gousseau, Y., Sun, H., & Maître, H. (2009). Structural high-resolution satellite image indexing.

Xu, C., Chai, Z., Xu, Z., Li, H., Zuo, Q., Yang, L., et al. (2022). HHF: Hashing-guided hinge function for deep hashing retrieval. *IEEE Transactions on Multimedia*.

Xu, C., Chai, Z., Xu, Z., Yuan, C., Fan, Y., & Wang, J. (2022). Hyp2 loss: Beyond hypersphere metric space for multi-label image retrieval. In *Proceedings of the 30th ACM international conference on multimedia* (pp. 3173–3184).

Yang, H.-F., Lin, K., & Chen, C.-S. (2017). Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(2), 437–451.

Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems* (pp. 270–279).

Yu, B., Liu, T., Gong, M., Ding, C., & Tao, D. (2018). Correcting the triplet selection bias for triplet loss. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 71–87).

Zhang, W., Huang, L., Wei, Z., & Nie, J. (2021). Appearance feature enhancement for person re-identification. *Expert Systems with Applications*, *163*, Article 113771.

Zhu, L., Lu, X., Cheng, Z., Li, J., & Zhang, H. (2020). Deep collaborative multi-view hashing for large-scale image search. *IEEE Transactions on Image Processing*, *29*, 4643–4655.

Zhu, L., Zheng, C., Guan, W., Li, J., Yang, Y., & Shen, H. T. (2023). Multi-modal hashing for efficient multimedia retrieval: A survey. *IEEE Transactions on Knowledge and Data Engineering*.