Proxy-Based Rotation Invariant Deep Metric Learning for Remote Sensing Image Retrieval

Zhoutao Cai, Yukai Pan, and Wei Jin

***Abstract*—Convolutional neural networks (CNNs) are frequently utilized in content-based remote sensing image retrieval (CBRSIR). However, the features extracted by CNNs are not rotationally invariant, which is problematic for remote sensing (RS) images where objects appear at variable rotation angles. In addition, because RS images contain a wealth of content and detailed information, CNNs may lead to information loss by superimposing multiple convolutional and pooling layers, affecting the ability of the model to extract features. To address these problems, this article proposes a proxy-based feature fusion network. By designing a proxy-based Euclidean distance contrast loss that combines contrast learning within the framework of metric learning, such that the distance between the source image and its rotated image embedding vector in the metric space is closer than any other image, thus endowing the model with a certain degree of rotation invariant. Meanwhile, the global correlation map is generated by multilayer fusion, under whose guidance the features of each layer are fused to improve the feature extraction capability of the model and to reduce the loss in the image flow process. Extensive experiments based on two public RS datasets show that the method achieves better performance compared to other methods.**

***Index Terms*—Deep neural network, feature fusion, metrics learning, remote sensing (RS) image retrieval, rotation invariant.**

I. Introduction

IN RECENT years, with the continuous development of satellite observation technology in recent years, millions of remote sensing (RS) images have been sent back to massive databases via various satellite remote sensors [1]. Meanwhile, due to the large amount of geographic scene information contained in RS images, there are significant difficulties in processing. Therefore, much research on image processing has focused on RS images. Some of the most common methods among them are target recognition and detection [2], [3], [4], image classification, image retrieval [5], etc. In this article, we investigate image retrieval based on RS images. The goal of image retrieval is to search large databases accurately and efficiently for specific information based on user requirements, returning all images similar to a provided reference image [6], [7].

Content-based remote sensing image retrieval (CBRSIR), as a branch of remote sensing image retrieval, has become increasingly popular in the RS community. In general, CBRSIR mainly consists of two components: feature extraction and similarity measurement [8]. Feature extraction focuses on extracting representative features from RS images, while similarity measurement measures the similarity between the query image and the target image to identify the most similar image from the database.

Most traditional feature extraction methods rely on manually designed features that are constructed by applying color, texture, or histograms of oriented gradient descriptors to the image. However, the use of manually designed features cannot accurately describe RS scenes because RS images are complex and contain a wealth of content and detailed information, which often leads to inferior retrieval results. Advancements in deep learning have led to the widespread adoption of convolutional neural networks (CNNs) for extracting high-level semantic information from RS images through an end-to-end manner [9], [10], [11]. CNNs utilize extensive training data, layer convolutional and pooling operations to produce feature maps at various scales. These maps encapsulate the image's features at different levels, from high-level semantics to low-level spatial details [12], [13]. However, as more layers are added to the network, more image information is lost as images move through the network. This bottleneck effect in many deep models often results in feature maps that might lack crucial shape or texture details, consequently leading to a diminished perception of detail in the features and degraded retrieval performance.

In contrast to natural images, RS images are captured by airborne or satellite-based sensors, and different projection directions can result in the same object appearing at different rotation angles in RS images. However, the features extracted by the CNN are not rotation invariant, which is problematic. For this scenario, we would like the trained CNN model to possess some rotation invariant capability, such that the retrieval results rank the rotated image of the query image higher than other similar images. Deep metric learning (DML) uses labeled images as input for end-to-end network training to extract representative features by enhancing intra-class compactness and inter-class separability. The approach has been successfully applied to RS image retrieval tasks [14], [15], [16]. In general, DML explores the relationships between sample pairs by constructing a metric space. However, information retention between sample pairs is typically heavily dependent on the construction of the sample pairs, and as more samples are trained, the training process

Manuscript received 13 October 2023; revised 15 December 2023, 4 February 2024, and 11 March 2024; accepted 24 March 2024. Date of publication 28 March 2024; date of current version 10 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42071323 and in part by the Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant LZJMZ24D050002. *(Corresponding author: Wei Jin.)*

The authors are with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China (e-mail: stormczt@outlook.com; panyuk@outlook.com; jinwei@nbu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3382845

becomes longer and converges slower. In recent years, proxy-based DML [17], [18], [19] has garnered a lot of attention due to factors like its fast convergence and low time complexity. The method assigns some learnable proxy points to each class to represent the overall features of that class. During network training, input samples are attracted to locations close to the proxy points of the same class to generate a meaningful embedding space.

Inspired by this, we utilize proxy-based DML to effectively learn deep embedding and steer network optimization toward creating a hierarchical metric space. This space preserves local structural integrity among RS images. Specifically, we propose a feature fusion module (FFM) that leverages the hierarchical extraction of information from backbone networks to generate a global correlation graph. The global correlation graph is used to guide the supervision of feature fusion at each layer, thereby reducing information loss, maximizing the use of features at different scales and levels, and enhancing the model's ability to extract features. After that, we design a proxy-based Euclidean distance contrast loss such that samples from the same class are closer to the proxy point of that class in the metric space, and the distance between an image and its rotated image embedding vector is closer than that of any other image. The main contributions of this article can be summarized as follows.

1) To fully integrate information between different layers, we propose a FFM that utilizes a global correlation graph to fuse features from each layer. This module improves image feature extraction and strengthens the interaction between features across different layers.
2) We propose a proxy-based Euclidean distance contrast loss for network training. By employing this contrast loss, we minimize the distance in the metric space between the source image and its rotated image embedding vector, compared to all other images. This allows the model to be endowed with certain class discriminative ability and rotation invariance at the same time.
3) The proposed method is evaluated on two public datasets, and the results show that the method achieves 98.07% and 99.49% mAP on UCMD and PatternNet respectively, which is better than other existing methods.

II. Related Works

A. Content-Based RS Image Retrieval

In recent years, deep learning has received considerable attention within the RS community. The hierarchical structure of CNNs can model extremely complex nonlinear functions and automatically learn the hyper-parameters of the CNN during training. As CNN models can extract high-level semantic features from images, these extracted features have been widely applied in CBRSIR. To depict the semantic content of RS scenes, extensive research has been conducted. For example, Zhou et al. [20] retrained mainstream CNN models for RS scenes, and experimental results demonstrate that their retrieval performance is noticeably superior to that of traditional techniques. Yu et al. [21] introduced a novel light-weighted nonlocal semantic fusion network based on hypergraph structure for CBRSIR, which is better to understand the global features of RS images with fewer parameters and less computation. A series of experimental results show the method achieves optimal retrieval performance. Hou et al. [22] proposed an attention-enhanced end-to-end discriminative network to effectively capture salient features in RS images. Xu et al. [23] developed a sketch-based RS image retrieval framework to search for images in RS databases based on hand-drawn sketches. Chen et al. [24] proposed a new deep-significance smoothed hashing algorithm to focus on the local fine-grained features and saliency information for drone images. Zhao et al. [25] obtained finer-grained multiscale features and achieved a larger receptive field by incorporating the proposed multiscale residual blocks, and the proposed multicontext attention modules increase the perceptual field by aggregating contextual information along channels and spatial dimensions. Experimental results show that this method achieves excellent results.

However, the number of labeled samples in the RS domain is relatively small compared to large-scale natural image datasets (e.g., ImageNet), which hinders the learning of CNN models. Li et al. [26] generated unsupervised features by designing unsupervised networks and fusing multiple features of an image for retrieval. Tan et al. [27] proposed a deep contrastive self-supervised hashing to learn precise hash codes by developing a new loss function using unlabeled images. Sumbul et al. [28] used a self-supervised approach to model the mutual information between different modalities, preserving intermodal similarity and eliminating intermodal differences. Liu et al. [29] proposed a lightweight similarity-based model, SBS-CNN, which uses multiple CNN models to pseudo-label unlabeled RS images by transfer learning. In addition, a new loss function was proposed to obtain compact features by taking into account the sequential relationship between categories. Experiments show that this approach achieves promising retrieval performance.

B. Proxy-Based DML

DML focuses on finding appropriate similarity measures between data pairs to maintain the required distance structure. In particular, contrast loss [30] explores the distance between two samples. It separates pairs from different classes in the embedding space while bringing pairs from the same class closer together. Triple loss [31] explores distances between three samples, each comprising a positive sample, a negative sample, and an anchor point. Triple loss aims to learn an embedding space where the similarity of negative pairs is lower than that of positive pairs by giving a margin. Song et al. [32] argued that for either contrast loss or triple loss, it is challenging to fully examine paired relationships between small batches of samples. To take into account the relationship between a similar pair and several different pairs, they proposed a lifted structured loss. Wang et al. [33] proposed multisimilarity loss that fully considers the three similarities of sample pair weighting and combines the two iterative steps of sampling and weighting. Song et al. [34] focused on the correlation between single-source and cross-source data samples and proposed a hashing-based DML method, which elaborated a label-based semantic loss and a hash-based metric loss to classify the extracted features and a decision-level fusion strategy was used to further improve the classification results, which yielded excellent results. However, the majority of twinned samples produced by pair-based metric learning methods may contain a considerable number of highly redundant or uninformative samples. In addition, choosing too many samples for network training can result in excessive time consumption, slow convergence, and significantly lower model performance.

Proxy-based DML has been an emerging approach in recent years. The method initializes the proxy using network parameters and optimizes the proxy as the network parameters are optimized. Movshovitz-Attias et al. [18] proposed the first proxy-based loss proxy-NCA, which significantly reduces training time by allowing similar samples to be clustered together and different samples to be separated based on assigning a proxy to each category. Yuan et al. [35] proposed a compact proxy-based deep learning framework designed with a compatible two-metric loss function to optimize the embedding space distribution and enhance the network's robust generalization ability. Liu et al. [36] used a prototype network to study the classification of hyperspectral images by extracting spectral spatial features to reduce uncertainty in sample labeling, and experiments showed that better results were obtained compared to conventional semi-supervised methods.

In this study, we utilize proxy-based metric learning to acquire distinctive features, offering improved efficiency in network training compared to the prevalent sample-based metric learning techniques. However, the effectiveness of the proxy-based network is substantially influenced by the choice of a suitable metric and loss function. To address this, we integrate contrast learning within the metric learning framework and devise a proxy-based contrast loss leveraging the Euclidean distance metric. This approach enables the concentration of similar image features around the agent points corresponding to their respective classes, while also aligning with our requirement for an embedding space with a hierarchical structure.

C. Multiscale Feature Representations

Deep learning has achieved excellent performance in almost all areas of computer vision. Due to its ability to fully explore the high-level semantic information of images, this method is very popular in the field of image retrieval. Many CNN-based models attempt to learn the deep semantic information of an image to further improve retrieval performance. However, deep semantic features do not include finer details such as an image's shape, texture, or color, which can compromise retrieval performance. To address this problem, many researchers have investigated multiscale feature representation. Lin et al. [37] proposed a top-down feature pyramid structure based on learning multiscale features for target detection through lateral connections. Ronneberger et al. [38] proposed U-Net by adding jump connections between the corresponding layers of the encoder and decoder on FCN. Hou et al. [39] introduced short connections in the jump layer structure of HED to address the scale space issue. However, they simply combine high-level features with low-level features, which inadequately extracts features. Zhao and Wu [40] used pyramidal feature attention networks to fuse multiscale features to generate saliency maps. Mari et al. [41] adopted a transformer-based approach to enhance the multiscale feature map information extracted by the backbone and achieve good retrieval performance. Chu et al. [42] employed a multiscale visual attention mechanism to extract features, combined it with the product quantization method, reduced dimensionality, and reduced the computational cost of retrieving RS images. Song et al. [43] proposed a DFFN model for fusing different levels of outputs, extracting more discriminative features of hyperspectral images, and improving classification accuracy.

Inspired by these approaches, we believe that combining multiscale features from different levels of CNNs can enhance the performance of image retrieval. Our method deviates from traditional approaches by adopting a multilayer fusion strategy, utilizing features from multiple layers to create a global correlation map. This approach takes into account the complementarity and contextual associations between diverse layers. Guided by the global correlation map, we selectively extract the portions of the feature maps from each layer that are rich in information and those that facilitate interlayer interactions. Subsequently, we exploit this interactive data to merge the characteristics of each layer, thereby minimizing informational loss. In the experimental part, it is verified that our method has favorable performance.

III. Method

To make full use of low-level features and high-level semantic features to generate more meaningful image representations, we propose a novel proxy-based feature fusion proxy-based feature fusion network (PBFFN) network. In addition, we introduce a new DML loss function to generate rotationally invariant RS image features, which addresses the rotationally variant problem from a DML perspective. The proposed approach consists of two main components: 1) a backbone CNN model for extracting deep fusion features and 2) a new DML loss for training this model in a rotation-invariant manner. Later, we will describe all these components in detail.

A. Network Architecture

In CNN-based network architectures, there is a notable loss of information as images pass through the network. Therefore, in order to fully leverage features from both higher and lower layers, minimize information loss, and enable the network to generate more meaningful image representations, we adopt a multilayer fusion strategy to integrate them. We adopt a deep neural network to extract features from the different layers of the image. Since ResNet effectively addresses the issue of deep network degradation, we adopt it as the backbone of our network architecture. The PBFFN network architecture is shown in Fig. 1. Accounting for variations in input image resolution, we resize the input image to 256 × 256 pixels, to preserve more fine details of the image while enhancing the robustness of batch processing.

We define $X = \{x_1, x_2, \ldots, x_N\}$ as the set of input RS images in each batch, where N represents the size of the batch. The corresponding labels for these images are given by $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i \in \{1, \ldots, C\}$, and C is the

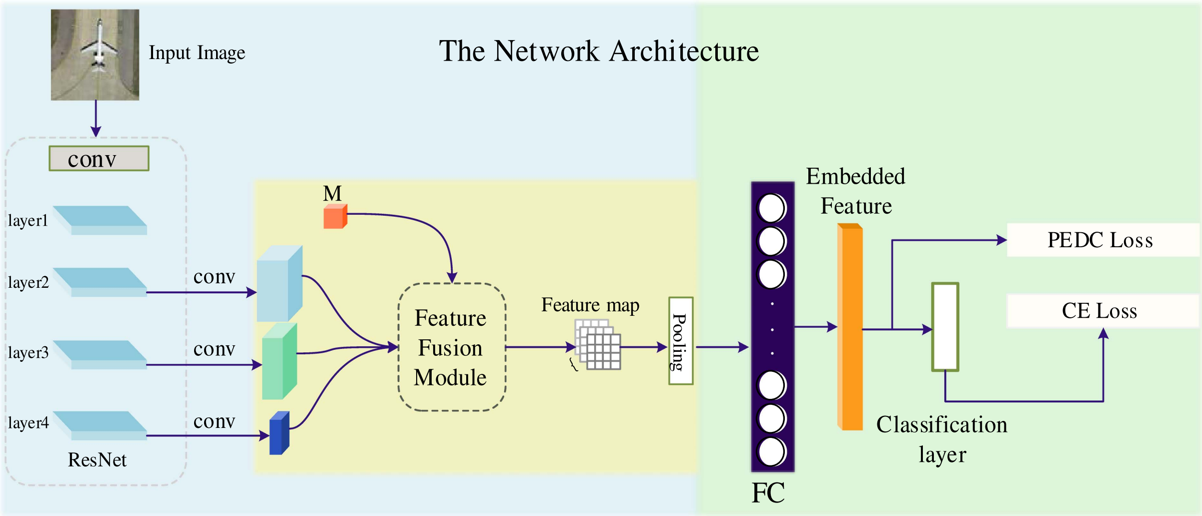
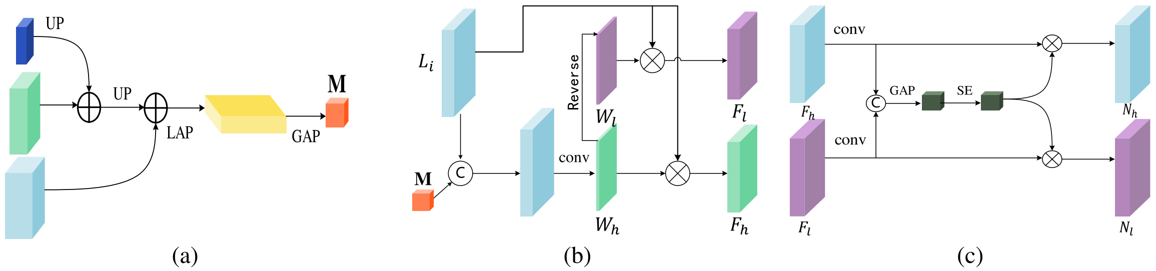


Fig. 1. Proposed PBFFN diagram. First, corresponding convolution operations are performed on the features extracted from layers 2, 3, and 4 of the network, and then enter the FFM module for feature fusion guided by the global correlation map M. The fused features undergo the global average pooling operation and then enter the metric space through an FC layer for learning. Finally, we introduce a classification layer and a jointly embedding layer to train the network model. Two loss functions will be employed to train the network.



(a) (b) (c)

Fig. 2. (a) Global correlation map M. The "LAP" and "GAP" denote the layer average pool and global average pool, respectively. (b) Feature decomposition module. (c) Feature update module (FUM).

total number of categories in the dataset images. To enhance the utilization of multiscale multilevel features, inspired by the article by Dai et al. [44], we adopt a global correlation graph to guide the feature fusion process at each level. The specific process is shown in Fig. 2(a). First, to ensure that multilevel features have the same channel dimensions at different heights and widths, we apply 1×1 convolution operations on the input image X successively at layers layer2, layer3, and layer4 within the backbone network, and then obtain the convolved features L_2, L_3, and L_4. Subsequently, L_4 is up-sample using bilinear interpolation and fused with L_3, and adopts the same operation to fuse L_2. Finally, layer average pooling and global average pooling operations are applied to obtain the global correlation map M

$$M = L_2 \oplus (\mathrm{UP}\,(L_3 \oplus \mathrm{UP}\,(\,L_4; L_3)\,; L_2))\,/3 \times H \times W \quad (1)$$

where H and W represent the length and width, respectively, $\mathrm{UP}(x; y)$ refers the up sampling x to the same size as y by bilinear interpolation, and $\oplus$ denotes the summing element values. We view the computed M as a global correlation map between features at each level.

After obtaining the global correlation graph M, we enhance the feature representation by interactively combining M with features from different layers using the global correlation map as guidance. We consider that a portion of the features in each layer contains rich information specific to that layer, while others are less informative and are only used for information exchange between layers. Therefore, we refer to the features containing rich information as the high-relevance features of the layer and the others as the low-relevance features of the layer. In order to obtain these two types of features, we first need to obtain their weight values. The specific structure is shown in Fig. 2(b). Here, L_i is the input to the module, and $i \in \{2, 3, 4\}$. First M and L_i perform a concatenation operation to interact with each other, and then they sequentially pass through a 1×1 convolution layer, batch normalization (BN) layer, and to keep the output between [0,1], we use a Sigmoid activation function to obtain the weights $W_h = \{W_{hi}\}_{i=2}^{4}$ of the highly correlated features. The above process can be expressed as

$$W_h = \phi\,(\mathrm{BN}\,(\mathrm{conv}\,(\mathrm{Concat}\,(L_i, M)))) \quad (2)$$

where Concat() denotes the concatenation operation, conv denotes the convolution operation to change the number of channels of the features, BN denotes the BN operation, and $\phi()$ denotes the Sigmoid activation function. Based on this, we acquire the high relevance weight W_h. Subsequently, we perform a reverse operation on W_h by using a matrix with all elements 1 to subtract W_h to obtain the low-relevance feature weights W_{l}. Finally, we multiply L_i with the high relevance feature weight W_h and the low relevance feature weight W_{l} to obtain the corresponding high relevance feature $F_h = \{F_{hi}\}_{i=2}^{4} \, \epsilon \mathcal{R}^{C\times H\times W}$ and the corresponding low relevance feature $F_{\mathrm{l}} = \{F_{li}\}_{i=2}^{4} \, \epsilon \mathcal{R}^{C\times H\times W}$, respectively.

In order to fully fuse the high- and low-correlation features obtained for each layer, we further interactively update the fused features to obtain the new high-correlation features $N_h = \{N_{hi}\}_{i=2}^{4}$ and the new low correlation features $N_{\mathrm{l}} = \{N_{li}\}_{i=2}^{4}$. The specific module structure is shown in Fig. 2(c). Specifically, for layer4, we take the low correlation features $F_{\mathrm{l}4}$ and high correlation features F_{h4}, pass them through a 1 × 1 convolutional layer and the ReLU activation function, respectively. After the channel-level concatenation operation, a global average pool operation and an squeeze-and-exaction operation are performed to obtain the channel weights W. Subsequently, We multiply W by F_{h4} and $F_{\mathrm{l}4}$ to obtain the new high- and low-correlation features N_{h4} and $N_{\mathrm{l}4}$, respectively. Furthermore, to fully leverage the connection information within the low-correlation features, for layer2 and layer3, we update the low-correlation feature F_{l} of this layer by concatenating and convolving it with the N_{l} of the next layer, the same operation is then performed with the high-correlation features of that layer. Finally, we discard the low-correlation features N_{l} that only serve as information linking, and then up-sample and concatenates the high-correlation features N_h of each layer to obtain the final multilevel fused features F_m. The above-mentioned process can be summarized as follows:

$$F_m = \mathrm{Concat}\left(N_{h2}, \mathrm{UP}\left(\mathrm{Concat}\left(N_{h3}, \mathrm{UP}\left(N_{h4}; N_{h3}\right)\right); N_{h2}\right)\right). \tag{3}$$

Subsequently, we apply the obtained F_m through average pooling, reduce the dimensions using a FC layer, and then enter the metric space for learning. Furthermore, we introduce a classification layer to optimize the model for training.

B. Loss Function

In order to make the whole framework end-to-end trainable while satisfying the retrieval requirements, the loss function needs to meet three criteria.

1) The loss function must be derivable for both CNNs and proxy.
2) The loss function is optimized to ensure that the embedding features of the image are close to the proxy of its corresponding class while maintaining a distance between proxies of different classes.
3) The source image and its rotated image are closer to each other in the embedding space than other images. To address the mentioned requirements, we propose a proxy-based Euclidean distance contrast loss function L_{PEDC}.

For metric learning, two different metric functions, Euclidean distance and cosine similarity, are commonly used to measure the similarity between two vectors. Euclidean distance is a measure of the straight-line distance between two points in a multidimensional space; the larger the Euclidean distance, the less similar the two points are, whereas cosine similarity is more concerned with the directional difference between the vectors. In order to construct a metric space with a hierarchical structure, we use the Euclidean distance function as a metric function, which more intuitively reflects the distance between the embedding features of the image and the class proxy from the scale size.

Intuitively, in the embedding space, instance feature points are often scattered on a low-dimensional manifold. On this manifold, if an instance feature is closer to a proxy in space than other features, then the instance should be assigned to the class represented by that proxy. We use the probability between each instance feature vector and its proximity proxy to design the proxy-based Euclidean distance loss L_{PED}.

Initially, we utilize the Euclidean distance between the feature vectors of an image and the proxy points to predict the correct class. Moreover, the probability that a feature is correctly classified $P(X \in C_i)$ is positively correlated with the distance from the image sample to its corresponding proxy point

$$P\left(X \in C_i\right) \propto - \left\| f\left(X;\theta\right) - p\right\|_{i2}^{2} \tag{4}$$

where p_i represents the category C_i to which the proxy corresponds and θ is a parameter of the model. It is important to note that for each category we set only one proxy point for calculation. The classical cross-entropy (CE) loss is expressed as $L_{\mathrm{CE}}\left(P\right) = \sum_{i=1}^{m} -y_i \log P$, where m is the number of training in a batch y_i is the one-hot label of x_i and P indicates the probability of an input sample belonging to a specific class. Similarly, we normalize the nonzero sum of distance-related probabilities to

$$P\left(X \in C_i\right) = \frac{\exp\left(-\alpha \left\| f\left(X;\theta\right) - p\right\|_{i2}^{2}\right)}{\sum_{n=1}^{C} \exp\left(-\alpha \left\| f\left(X;\theta\right) - p\right\|_{n2}^{2}\right)} \tag{5}$$

where C is the number of categories, i.e., the number of proxy points, α is a hyper parameter that controls the distance scale constraint. We then place this probability function into the loss function L_{PED} as follows:

$$\begin{aligned} L_{\mathrm{PED}} &= -\log\left(P\left(X \in C_i\right)\right) \\ &= -\log\left(\frac{\exp\left(-\alpha \left\| f\left(X;\theta\right) - p\right\|_{i2}^{2}\right)}{\sum_{n=1}^{C} \exp\left(-\alpha \left\| f\left(X;\theta\right) - p\right\|_{n2}^{2}\right)}\right). \end{aligned} \tag{6}$$

With this loss function, we can easily compute and optimize the network. The distance of each feature vector to its correct class of proxy points is decreasing.

In addition, in order to ensure that the model has a certain degree of rotation invariance and the embedding space has a certain level of hierarchy, it is vital to ensure that source images and their rotations are closer to each other than to similar category images in the same metric space. Contrast learning forces positive sample pairs to be similar to different negative sample pairs. This approach is commonly used in unsupervised learning tasks [45], [46], and effectively groups instances under their respective labels. It treats different augmentations of the same sample as positives, while others are considered negatives. Inspired by this, we consider the source image and its rotated image as one class, and the remaining images and their rotated images are assigned to other classes. This approach enables us to perform binary classification exercises for each training batch, alleviating the difficulty of training. By incorporating unsupervised ideas, we can improve the model training process for learning rotation invariant image properties. Specifically, given a set of input images $X = \{x_1, x_2, \ldots, x_N\}$, we use a randomly chosen rotation angle to generate the corresponding rotated images $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}$. We consider $(x_i, \hat{x}_i)$ as positive sample pairs and $(x_i, \hat{x}_j)(i \neq j)$ as negative sample pairs, all of which come from the same batch. Based on this framework, we design a contrast loss function L_C

$$L_C = \frac{1}{N} \sum_{i\,=\,1}^{N} \left(\ell \left(x_i, \hat{X} \right) \right) \tag{7}$$

$$l \left(x_i, \hat{X} \right) = -\log \frac{\exp \left(\frac{S(x_i, \hat{x}_i)}{\tau} \right)}{\sum_{n\,=\,1(n \neq i)}^{N} \exp \left(\frac{S(x_i, \hat{x}_n)}{\tau} \right)} \tag{8}$$

where $S(i, j)$ denotes the cosine similarity of features i, j, and τ denotes the temperature coefficient. Thus, the final proxy-based Euclidean distance contrast loss function L_{PEDC} is, where δ is the penalty factor

$$L_{\text{PEDC}} = L_{\text{PED}} \ + \delta L_C. \tag{9}$$

To further enhance the classification accuracy, we add a classification layer as a branch to increase the model's ability to distinguish between different classes. We use the cross-entropy loss L_{CE}. Finally, the total loss function is defined as

$$L_{\text{total}} = \ \lambda L_{\text{PEDC}} + \xi L_{\text{CE}} \tag{10}$$

where λ, δ, and ξ are the penalty parameters in order to better balance the individual losses.

IV. Experiment

To evaluate the effectiveness of our proposed PBFFN framework, several experiments are conducted in this section. First, we introduce the experimental dataset and the experimental environment. Second, we present the implementation details of the experiments. Finally, we present the experimental results and provide a detailed analysis.

A. Dataset and Settings

There are two public commonly used RS image datasets in our experiments UCMD and PatternNet.

1) The UCMD [47] is a public, free dataset of RS imagery. The dataset is manually extracted from the USGS National Map Urban Area Imagery series of large images. It has 21 images of different land-use categories, each containing 100 images. The image pixel size is 256 × 256 and the spatial resolution is 0.3 m.
2) The PatternNet [20] is a high-resolution RS image dataset for large-scale image retrieval. The dataset images are derived from Google Earth imagery or the Google Maps API for US cities. It contains a total of 30 400 images divided into 38 categories, each with 800 images of 256 × 256 pixels in size, each with a spatial resolution of 0.062–4.693 m.

In order to compare the effectiveness of different retrieval methods, we need to use evaluation criteria that are commonly employed in this field. We adopt mean average precision (mAP) and P@k as evaluation criteria. Higher values of mAP and P@k indicate better performance. Specifically, given a number of query images of Q, the value of mAP can be calculated by the following equation:

$$\text{mAP} = \frac{1}{Q} \sum_{r=1}^{Q} \text{AP}\,(r) \tag{11}$$

where AP denotes average precision and is defined as

$$\text{AP} = \frac{1}{R} \sum_{k=1}^{K} P\,(k)\,r\,(k) \tag{12}$$

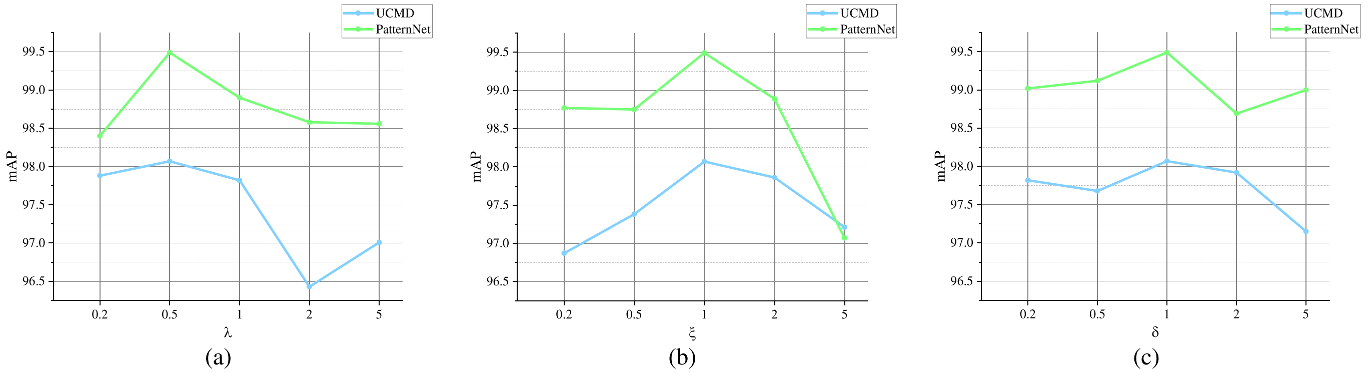
where $P(k)$ denotes the precision of the first k images retrieved, $r(k)$ is an indicator function that specifies whether the kth image is relevant to the query image: the value is 1 if it is relevant to the query image and 0 if it is not. K denotes the number of images retrieved and R is the number of ground-truths retrieved.

In general, the precision of a retrieval is the ratio of correct retrieval results to the retrieval results obtained. We use P@k as an auxiliary performance measure, which represents the precision when the number of returned results is K. It can be calculated by the following equation, where $R_k(r)$ represents the number of images associated with the query image in the first k images retrieved

$$P@k = \frac{1}{Q} \sum_{r\,=\,1}^{Q} \frac{R_k\,(r)}{k}. \tag{13}$$

We use a pretrained ResNet50 network as the backbone architecture to extract the depth features of the image, and then fine-tune its parameters using our designed RS dataset and loss function with the aim of better addressing our retrieval requirements for RS images. In addition, to enhance the generalization performance of the model and mitigate overfitting, we normalize the input images and randomly apply color augmentation, Gaussian blurring, and pixel noise operations. The penalty coefficients of the loss function λ, ξ, and δ are set to 0.5, 1, and 1, respectively. The initial learning rate is set to 0.0005 and decays by 50% after every 30 epochs. The optimizer for the experiments is set to Adam, where β_1 is 0.9 and β_2 is 0.999. The batch size for training is set to 32. Each class is assigned one proxy, which is initialized as the centroid of the class embedding.

The hardware environment for this experiment is Intel Core i7-7700 CPU@3.60 GHz, NVIDIA GeForce GTX 1080Ti 12G RAM.



(a) (b) (c)

Fig. 3. Effects of different penalty coefficient on mAP. (a) Effect of λ on mAP. (b) Effect of ξ on mAP. (c) Effect of δ on mAP.

B. Experimental Result

In this section, we will present the findings of our experiments on UCMD and PatternNet and conduct an analysis of the obtained results. For UCMD, we split the training and test datasets in a 1:1 ratio, randomly selecting 50% of the images in each class for training and using the remaining 50% for testing. For PatternNet, we split the training and test datasets in a ratio of 8:2.

1) Penalty Coefficient Analysis of the Loss Function: As shown in (9) and (10), our loss function has three penalty coefficients, i.e., λ, ξ, and δ. Specifically, λ and ξ are used to control the contribution of the image feature similarity information and the overall semantic information to the objective function, respectively. δ is used to control the contribution of the distance between the source image and its rotated image to the image feature similarity. We designed a series of experiments for analyzing the effects of all the above-mentioned parameters on the retrieval results, and it should be noted that when analyzing one parameter, the other two parameters are set to fixed values.

The ultimate outcomes of our experimentation are presented in Fig. 3. Specifically, Fig. 3(a) illustrates the evolution of mAP values as the λ parameter increases across two datasets. The graph reveals that as $\lambda < 0.5$, the mAP value experiences an upward trajectory. Conversely, as λ continues to rise, the mAP value demonstrates a descending trend. On the UCMD dataset, the retrieval performance hits the lowest point when $\lambda = 2$. Therefore, based on these observations, our proposed method yields satisfactory retrieval results for all datasets when λ is set to 0.5. Fig. 3(b) depicts the fluctuation of mAP values corresponding to increasing values of ξ on both datasets. It becomes evident that for both datasets, the peak mAP value is attained when $\xi = 1$. In addition, Fig. 3(c) showcases the variation of mAP values in response to escalating δ values on both datasets. Apparently, across both datasets, the mAP value reaches its highest point when $\delta = 1$, and the change in mAP value appears relatively smooth and consistent. In conclusion, after careful consideration, we have opted to set the penalty coefficients as follows: λ at 0.5, ξ at 1, and δ at 1.

2) Impact of Embedding Vector Size on Retrieval Performance: For the convenience of subsequent experiments, we first investigated the impact of different embedding sizes{32, 64, 128, 256} on the model retrieval performance. The experimental results are shown in Table I. Our model follows the same trend on both the UMCD and PatternNet datasets, with the mAP values increasing with embedding size and achieving the highest value at an embedding dimension of 128. Subsequently, the mAP value decreases again when the embedding dimension increases beyond 128. In addition, we observe that there is a greater variance in mAP values on the UCMD dataset compared to the PatternNet dataset. For example, when the embedding size is 128 on UCMD, the mAP value is 98.07%, which is 10.08% (32), 1.39% (64), and 1.99% (256) higher than the other embedding sizes. On PatternNet, meanwhile, the mAP value is 99.49% when the embedding size is 128, which represents an increase of 0.76% (32), 0.54% (64), and 0.13% (256) over the other embedding sizes. The possible explanation for this observation is that the UCMD dataset is small and has a limited number of training samples, which results in a stronger impact of embedding size on subsequent retrievals. Conversely, the PatternNet dataset is larger with a rich variety of categories and contains a significant amount of data from the same category. This makes changes in embedding size have relatively little impact on subsequent retrieval performance. Therefore, we ultimately decide to set the model embedding length to 128.

TABLE I
MAP VALUES OF DIFFERENT EMBEDDING SIZES WITHIN UCMD AND PATTERNNET

	32(%)	64(%)	128(%)	256(%)
UCMD	87. 99	96. 68	**98. 07**	96. 08
PatternNet	98. 73	98. 95	**99. 49**	99. 36

The bold values represent the best values in a column of data.

3) Comparison Experiments With State-of-the-Art Methods: To validate the effectiveness of the model proposed in this article, we conduct additional comparisons to evaluate its performance relative to existing methods. The comparison methods we employed are ResNet-50, Transformer [48], Swim-Transformer [49], FMT-RAN [50], ST-RAN [50], FAH [51], DHPL [52], DHCNN [53], and AHCL [54], respectively. FMT-RAN trains

TABLE II
COMPARISON OF UCMD WITH STATE-OF-THE-ART METHODS

Methods	mAP	P@5	P@10	P@20	P@50
ResNet50	83.23	93.35	91.66	90.22	87.82
Transformer	84.56	94.76	91.81	91.02	88.45
Swim-Transformer	93.95	97.52	96.73	96.10	95.62
DHPL	94.52	98.18	97.64	96.99	96.01
FMT-RAN	87.76	97.22	96.83	96.27	92.40
ST-RAN	81.87	96.10	95.39	94.39	88.61
FAH	95.75	98.74	98.60	98.51	97.22
DHCNN	95.22	99.08	98.82	98.73	97.09
AHCL	97.29	99.20	98.99	98.86	97.19
ours	**98.07**	**99.53**	**99.41**	**99.37**	**98.71**

The bold values represent the best values in a column of data.

TABLE III
COMPARISON OF PATTERNNET WITH STATE-OF-THE-ART METHODS

Methods	mAP	P@5	P@10	P@50	P@100
ResNet50	95.28	98.84	98.48	97.14	95.62
Transformer	97.54	98.26	98.03	97.24	96.41
Swim-Transformer	98.26	98.53	98.29	97.65	96.73
DHPL	98.66	99.10	98.42	98.15	97.78
FMT-RAN	97.73	98.80	98.63	98.36	98.01
ST-RAN	89.80	97.79	97.12	94.02	89.93
FAH	99.28	99.36	99.34	99.40	99.44
DHCNN	96.89	98.62	98.32	97.87	97.52
AHCL	98.50	99.06	98.48	98.36	97.87
ours	**99.49**	**99.60**	**99.57**	**99.58**	**99.59**

The bold values represent the best values in a column of data.

the subsequent layers to be rotation invariant by feeding the input image into the pretrained VGG and rotating the feature map generated by its last pooling layer generated by four different angles. ST-RAN retrieves objects of arbitrary angles by introducing an STN module, using the original image as input to generate an affine image that matches a rotated image, and finally, the RAN module is used to learn rotation-invariant feature representations simultaneously. FAH consists of two modules, DFLM and AHLM. It enhances retrieval quality by extracting dense features from the image. DHPL is a proxy-based hash retrieval method, which combines hash learning with metric learning. DHCNN retrieves similar images and classifies semantic labels within the same framework, which utilizes CNN to extract features and convert them into compact hash codes through a hash layer. All six aforementioned methods were designed for RS image retrieval tasks. AHCL generates hash codes for query and database images in an asymmetric manner and uses the semantic information of each image and the similarity information of pairs of images as supervisory information to train the deep hash network, which improves the representation of deep features and hash codes. Transformer is a sequence model that relies on attention mechanism. And swim-transformer was developed to address limitations introduced by Transformer in computer vision. It adopts a hierarchical architecture for adapting images of different scales and utilizes a sliding window approach to enable linear-complexity attention computations. To ensure a fair comparison, we established a unified feature vector length of 128 for each model and trained them under identical training conditions.

The experimental results are shown in Tables II and III. Based on the experimental data, it can be observed that PBFFN network model achieves the most excellent performance both on the UCMD and PatternNet datasets. On the UCMD dataset, the PBFFN model exhibits the best mAP value of 98.07%, which is 0.80% (AHCL), 2.42% (FAH), 2.99% (DHCNN), 3.76% (DHPL), 4.38% (Swim-Transformer), 10.31% (FMT-RAN), 13.76% (Transformer), and 16.2% (ST-RAN) higher than the comparison methods, respectively. We argue that the performance of the ST-RAN model relies on the quality of the affine image generated after passing the image through the STN module. However, the network model of this method is more challenging to train and will cause poor quality in the generated affine images, resulting in unsatisfactory final retrieval results. In addition, we also argue that the transformer model is particularly prone to overfitting due to its large number of parameters, leading to unsatisfactory performance when dealing with small datasets. Compared to these approaches, our method demonstrates good performance by using proxy points to capture the overall characteristics of each category and incorporating contrast loss, which leads to a more compact arrangement of similar images in the metric space and more consistent P@K. On the PatternNet dataset, our method achieves 99.49% mAP, 0.21% (FAH), 0.83% (DHPL), 1.01% (AHCL), 1.25% (Swim-Transformer), 1.76% (FMT-RAN), 1.98% (Transformer), 2.68% (DHCNN), 4.21% (ResNet-50), and 9.69% (ST-RAN) higher than the comparison models, respectively.

To visually illustrate the retrieval capabilities of our method using the dataset, we display selected examples of the retrieval outcomes in Figs. 4 and 5. For each dataset, we randomly select an image as a query object and acquire the retrieval results by sorting the similarity measurements between the query object and the target image. The query image is presented in the first column, whereas the remaining columns display the retrieved results. Images within red bounding boxes symbolize incorrect retrieval results. Restricted by space, we only present the top 10 retrieval results. For UCMD and PatternNet, we summarize the numbers of correct results within the top 40 and top 100 retrieval results, respectively. These examples underline the effectiveness of our method.

Besides the overall results, we also calculate the mAP values for the different semantic categories. Due to table size constraints, we only display the results of five methods. The experimental results are shown in Tables IV and V. We can observe that our method is almost significantly better numerically than other methods, and has stable performance in most categories. On the UCMD dataset, our method performs slightly worse in the categories "dense residential," and "medium residential," but the mAP values still reach 92.16% and 83.46%. In some complex categories, such as "baseball diamond," and "sparse residential,"

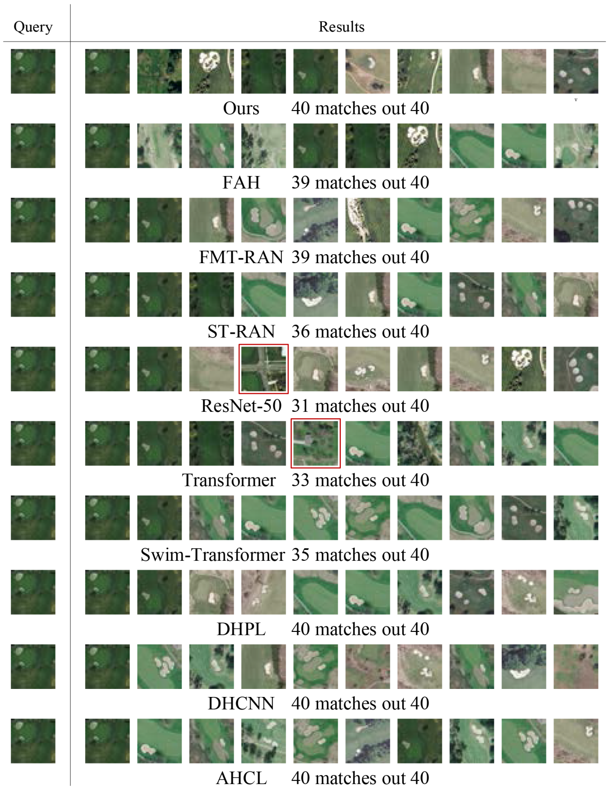


Fig. 4. Retrieval examples within UCMD through different methods.

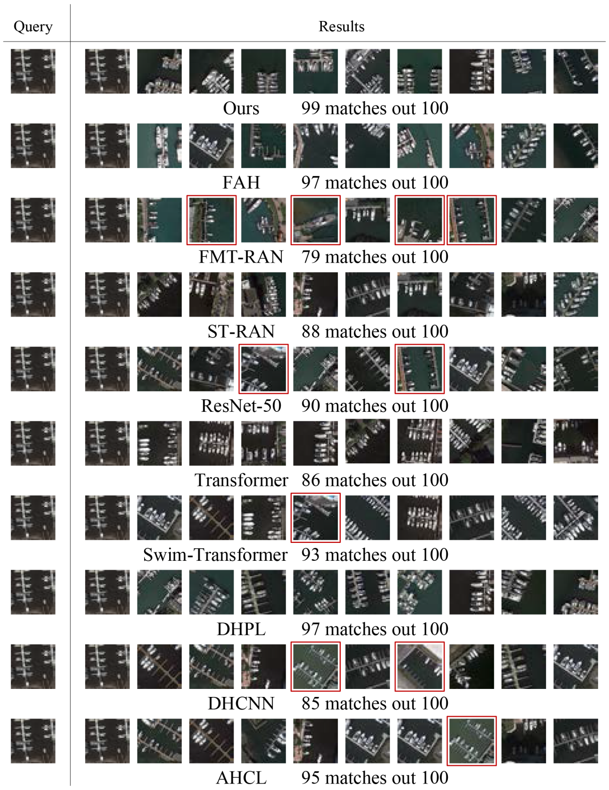

Query
Results
Ours 99 matches out 100
FAH 97 matches out 100
FMT-RAN 79 matches out 100
ST-RAN 88 matches out 100
ResNet-50 90 matches out 100
Transformer 86 matches out 100
Swim-Transformer 93 matches out 100
DHPL 97 matches out 100
DHCNN 85 matches out 100
AHCL 95 matches out 100

Fig. 5. Retrieval examples within PatternNet through different methods.

TABLE IV
mAP VALUES OF EACH INDIVIDUAL CATEGORY FOR THE UCMD DATASET THROUGH DIFFERENT METHODS

	Ours	FMT-RAN	FAH	Swin-Tranform	AHCL
agricultural	98.00	93.52	100.0	91.03	97.11
airplane	100.0	85.86	94.12	93.99	99.21
baseball diamond	99.69	92.73	94.12	92.79	97.12
beach	100.0	99.76	100.0	99.76	99.76
buildings	100.0	73.34	86.88	92.73	92.00
chaparral	100.0	100.0	100.0	97.18	100.0
dense residential	92.16	84.37	93.58	94.54	96.13
forest	94.12	93.62	94.12	81.64	98.12
freeway	100.0	93.55	94.12	93.52	100.0
golfcourse	100.0	88.25	96.04	92.73	97.25
harbor	100.0	99.92	100.0	94.57	98.50
intersection	100.0	92.55	94.12	97.61	96.82
medium residential	83.46	73.43	90.4	85.75	93.16
mobile home park	100.0	87.94	100.0	97.55	100.0
overpass	100.0	86.77	95.6	94.49	97.38
parking lot	100.0	96.00	100.0	94.13	99.58
river	100.0	86.13	100.0	89.35	97.28
runway	100.0	99.37	100.0	100.0	98.70
sparse residential	97.92	93.43	89.1	93.63	94.86
storage tanks	94.05	34.50	88.48	98.55	91.68
tennis court	100.0	87.85	100.0	97.44	98.39

our method has a clear advantage. For example, the mAP value of our method for "baseball diamond" is 99.69%, whereas other methods have mAP values ranging from 92.73% to 97.12%. On the PatternNet dataset, our model performs better and more consistently than other methods. Our method is slightly inferior to FAH in terms of "closed road," "nursing home," and "sparse residential." Moreover, our method achieves better performance in the categories of "intersection," "tennis court," "ferry terminal" categories. Taking "ferry terminal" as an example, our method improves the mAP values by 2.97% (FAH), 3.26% (AHCL), 6.58% (DHPL), and 11.21% (FMT-RAN) respectively. These results demonstrate that our method is effective for the CBRSIR task.

In addition to the quantitative metrics, retrieval efficiency is also an important factor when designing a retrieval algorithm. We conduct a comparative analysis of the proposed method and other methodologies, with a particular focus on retrieval time. Given that the model training phase is an offline procedure that requires only a one-time execution, the time investment for various models is deemed reasonable. Our primary consideration here lies in assessing the time expenditure associated with retrieving images through these models. Specifically, we carry

TABLE V
mAP Values of Each Individual Category (20 Categories) for the PatternNet Dataset Through Different Methods

	Ours	FMT-RAN	FAH	Swin-Tranform	AHCL
airplane	100.0	98.89	100.0	96.64	98.49
baseball field	100.0	97.85	99.98	97.85	100.0
basketball court	98.71	90.42	97.85	98.23	96.96
closed road	98.47	96.64	99.33	94.37	93.83
ferry terminal	98.88	87.67	95.91	95.38	95.75
football field	99.38	98.96	99.38	97.81	98.44
harbor	100.0	97.03	96.83	97.96	96.96
intersection	99.76	97.03	98.33	97.08	96.20
mobile home park	100.0	99.29	100.0	100.0	98.86
nursing home	96.93	92.52	98.51	94.25	93.37
oil gas field	100.0	99.38	100.0	99.10	100.0
parking space	100.0	99.98	100.0	100.0	98.82
railway	100.0	99.32	100.0	100.0	98.59
river	100.0	100.0	100.0	99.95	100.0
solar panel	100.0	99.23	100.0	100.0	100.0
sparse residential	93.36	84.31	97.64	92.16	92.58
storage tank	99.98	99.26	100.0	92.21	99.28
tennis court	99.98	97.08	98.07	97.64	97.81
transformer station	100.0	98.86	99.09	99.77	98.61
wastewater plant	99.92	99.29	100.0	98.65	97.81

out ten experiments for each method, randomly selecting 50 images to serve as the query set and retrieving the top 100 images from the dataset. The final result is determined by calculating the average value of these experiments.

The experimental results are shown in Table VI. We can find that the hash retrieval methods are generally faster than other methods for retrieval, in which the retrieval time of the AHCL method on both datasets is much lower than that of other methods, which may be due to the fact that this asymmetric strategy method has some effectiveness on hash code learning. Although our proposed method is not as fast as the retrieval speed of hash retrieval methods, it still achieves 0.2257 s (UCMD) and 0.3176 s (PatternNet), which is acceptable in terms of time performance. Besides, our method has high retrieval accuracy, which, taken together, verifies the effectiveness of the proposed method.

4) Ablation Study: In this section, ablation experiments are conducted to investigate the contributions of individual components in the PBFFN model. In particular, the proposed PBFFN network is composed of two key components: the FFM module and the proxy-based Euclidean distance contrast loss. Specifically, the proxy-based Euclidean distance contrast loss L_{PEDC} consists of L_{PED} and L_C. Therefore, we propose the following methods for ablation experiments: ours (withoutL_C), ours (withoutL_{PED}), ours (without L_{PEDC}), ours (without FUM), ours (without FFM), ours (without FFM and L_C), and ours (without FUM and L_C). The first five models lack the components L_C, L_{PED}, L_{PEDC}, FUM, and FFM. The latter two models represent the lack of FFM, L_C components and FUM, L_C components, respectively. Then we select mAP as a metric to numerically evaluate these methods, and the final experimental results are shown in Tables VII and VIII.

The results clearly indicate that our PBFFN method not only achieves superior performance but also shows that each of its components contributes positively to the overall retrieval effectiveness. Specifically, compared to the model without the FFM module, the model with the FFM module increases mAP by 3.78% and 1.27% on UCMD and PatternNet, respectively. This indicates that the FFM module enhances the model's ability to extract image features and improves retrieval performance. Furthermore, for the FFM module, incorporating the FUM module versus not incorporating it leads to an increase in mAP of 2.17% and 0.93% on UCMD and PatternNet, respectively. Compared to the model without L_C, adding L_C leads to an increase in mAP of 0.50% and 0.38% on UCMD and PatternNet, respectively. In addition, compared to the model without L_{PED}, adding L_{PED} causes an increase in mAP of 1.76% and 0.79% on UCMD and PatternNet, respectively. We observe that adding L_C and L_{PED} individually both enhances the final retrieval results to some extent. Furthermore, when we combine the two, using proxy-based Euclidean distance comparison loss L_{PEDC}, it is encouraging to observe that mAP increases by 2.72% and 1.69% on UCMD and PatternNet, respectively. This result is superior to using either loss function alone, highlighting the effectiveness of our proposed loss function. We also discover that our method, without the use of L_C, yields superior results when compared to the other ablation techniques. This is likely because L_C aims to enhance compactness between the source image and its rotated version within the same class metric space. Meanwhile, L_{PED} intends to bring the features of images belonging to the same class closer to their respective proxy while increasing their separation from proxy of differing classes. Therefore, adding L_{PED} alone has better retrieval performance than adding L_C alone. Meanwhile, our method without FFM achieves a performance enhancement of 0.127% compared to our current approach without both FFM and L_C on UCMD. In addition, our method without FUM attains a performance boost of 0.282% compared to our current method without both FUM and L_C. This indicates that L_C plays a significant role in enhancing retrieval performance.

We also investigate the impact of different feature fusion on retrieval performance. Specifically, we reduce the number of

TABLE VI
COMPARISON OF RETRIEVAL TIME (IN SECONDS) OF DIFFERENT METHODS

	ours	ResNet50	Transformer	Swim-Transformer	DHPL	FMT-RAN	ST-RAN	FAH	DHCNN	AHCL
UCMD	0.2257	0.2163	0.2148	0.3141	0.2220	0.2314	0.1968	0.1978	0.1848	0.0826
PatternNet	0.3176	0.3140	0.3002	0.3940	0.3180	0.3306	0.2786	0.2834	0.2708	0.1752

TABLE VII
RESULTS OF THE ABLATION EXPERIMENT ON UCMD

Methods	mAP	P@5	P@10	P@20	P@50
ours(without L_C)	97.57	99.46	99.37	99.28	98.41
ours(without L_{PED})	96.31	98.91	98.79	98.73	97.57
ours(without L_{PEDC})	95.35	98.83	98.60	98.51	96.98
ours(without FFM)	94.49	99.24	98.97	98.59	96.57
ours(without FUM)	95.99	99.27	99.05	98.94	97.47
ours(without FUM and L_C)	95.72	99.12	99.06	98.91	97.32
ours(without FFM and L_C)	94.37	99.05	98.72	98.33	96.50
ours	**98.07**	**99.53**	**99.41**	**99.37**	**98.71**

The bold values represent the best values in a column of data.

TABLE VIII
RESULTS OF THE ABLATION EXPERIMENT ON PATTERNNET

Methods	mAP	P@5	P@10	P@20	P@100
ours(without L_C)	99.11	99.34	99.32	99.30	99.31
ours(without L_{PED})	98.70	99.25	99.14	98.99	98.95
ours(without L_{PEDC})	97.80	99.05	98.84	98.47	98.21
ours(without FFM)	98.24	99.22	98.98	98.64	98.49
ours(without FUM)	98.56	99.31	99.22	98.98	98.86
ours(without FUM and L_C)	98.40	99.27	99.10	99.01	98.72
ours(without FFM and L_C)	97.96	99.14	98.94	98.53	98.38
ours	**99.49**	**99.60**	**99.57**	**99.58**	**99.59**

The bold values represent the best values in a column of data.

TABLE IX
RESULTS OF FUSING DIFFERENT FEATURES IN FFM ON UCMD

Methods	mAP	P@5	P@10	P@20	P@50
FFM(L2+L4)	97.56	99.39	99.28	99.24	98.43
FFM(L3+L4)	97.62	99.33	99.30	99.30	98.52
FFM(L2+L3)	96.81	99.16	99.01	98.97	97.93

TABLE X
RESULTS OF FUSING DIFFERENT FEATURES IN FFM ON PATTERNNET

Methods	mAP	P@5	P@10	P@20	P@50
FFM(L2+L4)	98.84	99.31	99.22	99.11	99.06
FFM(L3+L4)	98.96	99.48	99.41	99.21	99.10
FFM(L2+L3)	98.76	99.30	99.20	99.08	99.03

feature fusions and select different combinations of convolutional layer features for fusion. The experimental results are shown in Tables IX and X. We find that the combination of convolutional layer features L3 and L4 achieves better results than other combinations. On UCMD, this combination is 0.062% and 0.84% higher than the other two combinations, respectively. On PatternNet, this combination is 0.121% and 0.203% higher than the other two combinations, respectively. This may be because the deeper features extracted by the backbone network help improve the retrieval performance of the model. Our method utilizes three layers of features, L2, L3, and L4, for fusion. Although it improves the complexity of the network to some extent, it can achieve better results than ablation methods.

These findings indicate that the FFM module enhances feature learning capabilities, while L_{PEDC} causes a more compact embedding of similar images in the metric space, and the proposed method is effective.

5) Verification of Rotation Invariant: In this section, we conduct experimental analysis to investigate whether rotated images' deep embedding lies adjacent to each other in the embedding space, i.e., whether the model exhibits a certain degree of rotation invariance after training. To present the experimental results more intuitively, we first rotate the images in the retrieval database by different angles (90°, 180°, 270°) to create a new retrieval database. For the new database, we randomly select one image per category as the query object and obtain the retrieval results by ranking the similarity measure between the query object and the target image. We will provide some retrieval examples showing the top-10 images retrieved from this new database. As both FMT-RAN and ST-RAN are used to extract rotation invariant features in the above comparison experiments, with FMT-RAN performing better than ST-RAN, we utilize FMT-RAN as our comparison method.

The experimental results are shown in Figs. 6 and 7. Based on the figures, we observe that the FMT-RAN method on UCMD does not fully prioritize the rotated images of the source images in the retrieval results for the query images belonging to the

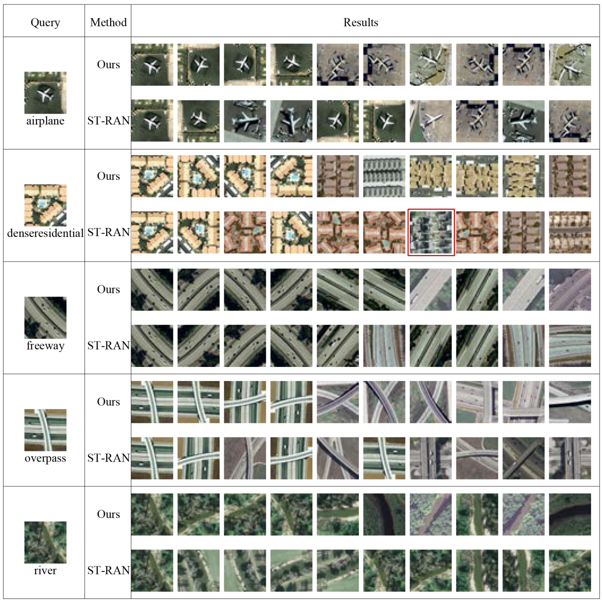


Fig. 6. Top-10 search results on the expanded UCMD dataset.

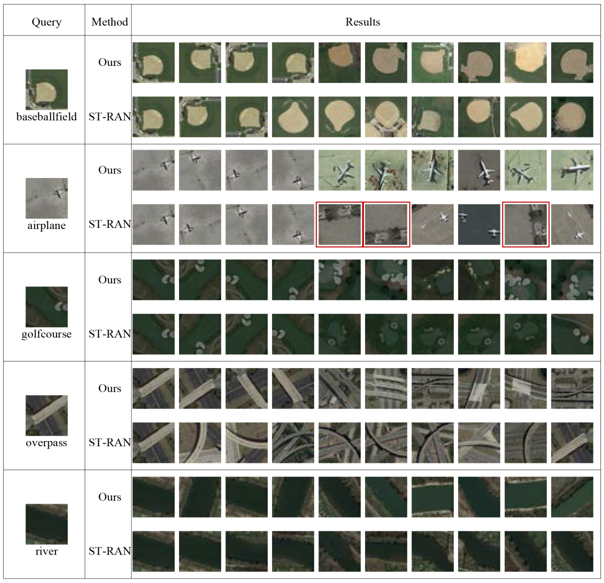


Fig. 7. Top-10 search results on the expanded PatternNet dataset.

categories “airplane,” “freeway,” and “overpass.” In addition, retrieval errors are detected in the results for query images from some categories. When it comes to PatternNet, the FMT-RAN method does not fully prioritize the retrieval of the rotated images of the source image for the query images in the categories “baseball field” and “overpass.” In contrast to the FMT-RAN method, our proposed method not only retrieves the correct image in the new database but also prioritizes the retrieval of the rotated image of the source image. Therefore, we conclude that our method not only enhances the feature extraction capability of CNN models to enable them to effectively handle RS image retrieval tasks but also endows the model with some rotation invariant ability.

V. Conclusion

In this article, we propose a novel PBFFN framework for the retrieval of RS images. RS images contain rich structural and textural information, but this information may be lost during the piping of images through a deep network, leading to features that inadequately characterize the images and degrading retrieval performance. To address this issue, we introduce a FFM that applies a hierarchical fusion strategy to merge high- and low-level features, thus reducing the loss of image information during the feedforward pass of the network. In addition, we integrate metric learning by utilizing proxy to represent class-level feature embedding, which enhances intraclass compactness and interclass separability by gauging the similarity between sample features. To adapt to this proxy-based training mechanism, we design a loss function that gives the model a certain degree of rotation invariance and class discrimination ability. We validate the effectiveness of the proposed method through comprehensive experiments conducted on two publicly available RS image datasets.

In our work, there are still several limitations that need to be overcome. For example, only one proxy point is selected to represent each category, the feature extraction of the model lacks a certain degree of interpretability, and a large amount of labeled data is required for model training. In future work, we aim to enhance the interpretability of the feature extraction process within the model, and further optimize the loss function, refer to existing hash methods, and use unsupervised hash methods to train the network.

References

[1] Y. Ma et al., “Remote sensing big data computing: Challenges and opportunities,” *Future Gener. Comput. Syst.*, vol. 51, pp. 47–60, Oct. 2015.

[2] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *Int. Soc. Photogrammetry Remote Sens. J. Photogrammetry Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[3] P. Wang, X. Sun, W. Diao, and K. Fu, “FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3377–3390, May 2020.

[4] G. S. Xia et al., “DOTA: A large-scale dataset for object detection in aerial images,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3974–3983.

[5] O. E. Dai, B. Demir, B. Sankur, and L. Bruzzone, “A novel system for content-based retrieval of single and multi-label high-dimensional remote sensing images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 7, pp. 2473–2490, Jul. 2018.

[6] X.-Y. Tong, G.-S. Xia, F. Hu, Y. Zhong, M. Datcu, and L. Zhang, “Exploiting deep features for remote sensing image retrieval: A systematic investigation,” *IEEE Trans. Big Data*, vol. 6, no. 3, pp. 507–521, Sep. 2020.

[7] H. Daschiel and M. Datcu, “Information mining in remote sensing image archives: System evaluation,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 1, pp. 188–199, Jan. 2005.

[8] B. Demir and L. Bruzzone, “A novel active learning method for content based remote sensing image retrieval,” in *Proc. IEEE 23rd Signal Process. Commun. Appl. Conf.*, 2015, pp. 2130–2133.

[9] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[10] Z. Yuan et al., “A lightweight multi-scale crossmodal text-image retrieval method in remote sensing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5612819.

[11] G. Sumbul et al., “BigEarthNet-MM: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and

retrieval [software and data sets]," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 3, pp. 174–180, Sep. 2021.

[12] T.-N. Le and A. Sugimoto, "Video salient object detection using spatiotemporal deep features," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5002–5015, Oct. 2018.

[13] K. Ishikura, N. Kurita, D. M. Chandler, and G. Ohashi, "Saliency detection based on multiscale extrema of local perceptual color differences," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 703–717, Feb. 2018.

[14] L. Yan, R. Zhu, N. Mo, and Y. Liu, "Cross-domain distance metric learning framework with limited target samples for scene classification of aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3840–3857, Jun. 2019.

[15] J. Kang, R. Fernandez-Beltran, D. Hong, J. Chanussot, and A. Plaza, "Graph relation network: Modeling relations between scenes for multilabel remote-sensing image classification and retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4355–4369, May 2021.

[16] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[17] E. W. Teh, T. DeVries, and G. W. Taylor, "ProxyNCA++: Revisiting and revitalizing proxy neighborhood component analysis," presented at the Comput. Vis. – ECCV 2020: 16th Eur. Conf., Glasgow, U.K., August 23–28, 2020, Proc., Part XXIV, Glasgow, United Kingdom, 2020.

[18] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No fuss distance metric learning using proxies," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 360–368.

[19] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3235–3244.

[20] W. Zhou, S. Newsam, C. Li, and Z. Shao, "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *Int. Soc. Photogrammetry Remote Sens. J. Photogrammetry Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018.

[21] H. Yu et al., "A light-weighted hypergraph neural network for multimodal remote sensing image retrieval," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 2690–2702, Mar. 2023.

[22] D. Hou, S. Wang, X. Tian, and H. Xing, "An attention-enhanced end-to-end discriminative network with multiscale feature learning for remote sensing image retrieval," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8245–8255, Sep. 2022.

[23] F. Xu, W. Yang, T. Jiang, S. Lin, H. Luo, and G. S. Xia, "Mental retrieval of remote sensing images via adversarial sketch-image feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7801–7814, Nov. 2020.

[24] Y. Chen, J. Huang, L. Mou, P. Jin, S. Xiong, and X. X. Zhu, "Deep saliency smoothing hashing for drone image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4700913.

[25] D. Zhao, Y. Chen, and S. Xiong, "Multiscale context deep hashing for remote sensing image retrieval," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7163–7172, Jul. 2023.

[26] Y. Li, Y. Zhang, C. Tao, and H. Zhu, "Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion," *Remote Sens.*, vol. 8, no. 9, 2016, Art. no. 709, doi: 10.3390/rs8090709.

[27] X. Tan, Y. Zou, Z. Guo, K. Zhou, and Q. Yuan, "Deep contrastive self-supervised hashing for remote sensing image retrieval," *Remote Sens.*, vol. 14, no. 15, 2022, Art. no. 3643, doi: 10.3390/rs14153643.

[28] G. Sumbul, M. Müller, and B. Demir, "A novel self-supervised cross-modal image retrieval method in remote sensing," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 2426–2430.

[29] Y. Liu, L. Ding, C. Chen, and Y. Liu, "Similarity-based unsupervised deep transfer learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7872–7889, Nov. 2020.

[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 1735–1742.

[31] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2015, pp. 84–92.

[32] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012.

[33] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5017–5025.

[34] W. Song, Y. Dai, Z. Gao, L. Fang, and Y. Zhang, "Hashing-based deep metric learning for the classification of hyperspectral and LiDAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Oct. 2023, Art. no. 5704513.

[35] Y. Yuan, C. Wang, and Z. Jiang, "Proxy-based deep learning framework for spectral–Spatial hyperspectral image classification: Efficient and robust," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Feb. 2021, Art. no. 5501115.

[36] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.

[37] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Berlin, Germany: Springer-Verlag, 2015, pp. 234–241.

[39] Q. Hou, M. M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, 2019.

[40] T. Zhao and X. Wu, "Pyramid feature attention network for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3080–3089.

[41] C. R. Mari, D. V. Gonzalez, and E. Bou-Balust, "Multi-scale transformer-based feature combination for image retrieval," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 3166–3170.

[42] J. Chu, L. Li, and X. Xiao, "Remote sensing image retrieval by multi-scale attention-based CNN and product quantization," in *Proc. IEEE 40th Chin. Control Conf.*, 2021, pp. 8292–8297.

[43] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[44] Y. Dai, W. Song, Y. Li, and L. D. Stefano, "Feature disentangling and reciprocal learning with label-guided similarity for multi-label image retrieval," *Neurocomputing*, vol. 511, pp. 353–365, Oct. 2022.

[45] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[46] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.

[47] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inform. Syst.*, 2010, pp. 270–279.

[48] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[49] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[50] Z.-Z. Wu, C. Zou, Y. Wang, M. Tan, and T. Weise, "Rotation-aware representation learning for remote sensing image retrieval," *Inf. Sci.*, vol. 572, pp. 404–423, Sep. 2021.

[51] 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.

[52] X. Shan, P. Liu, Y. Wang, Q. Zhou, and Z. Wang, "Deep hashing using proxy loss on remote sensing image retrieval," *Remote Sens.*, vol. 13, no. 15, 2021, Art. no. 2924.

[53] W. Song, S. Li, and J. A. Benediktsson, "Deep hashing learning for visual and semantic retrieval of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 11, pp. 9661–9672, Nov. 2021.

[54] W. Song, Z. Gao, R. Dian, P. Ghamisi, Y. Zhang, and J. A. Benediktsson, "Asymmetric hash code learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5617514.

Zhoutao Cai received the bachelor's degree in automation from Hangzhou Dianzi University, Hangzhou, China, in 2018. He is currently working toward the master's degree in computer technology with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China.

His research interests include remote sensing image processing and deep learning.

Yukai Pan received the bachelor's degree in internet of things engineering from Huzhou University, Huzhou, China, in 2021. He is currently working toward the master's degree in computer technology with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China.

His research interests include remote sensing image processing and deep learning.

Wei Jin received the Ph.D. degree in optical engineering from Chongqing University, Chongqing, China, in 2006.

He is currently a Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China.

His research interests include remote sensing image processing, image retrieval, deep learning, and computer vision.