A Triplet Nonlocal Neural Network With Dual-Anchor Triplet Loss for High-Resolution Remote Sensing Image Retrieval

Maoding Zhang, Qimin Cheng, *Member, IEEE*, Fang Luo, and Lan Ye

***Abstract*—Conventional deep-learning-based retrieval models are generally trained under the framework of scene classification with cross-entropy loss, this way focuses only on the output probability corresponding to the label of input samples, while ignoring the predictive information of other categories, which makes the retrieval accuracy susceptible to the intraclass difference of the image samples. And conventional methods often used fixed-size convolution kernels that only consider the local area with fixed sizes, thus largely ignoring the global information. In response to the above problems, this article constructs a triplet nonlocal neural network (T-NLNN) model that combines deep metric learning and nonlocal operation. The proposed T-NLNN follows the three-branch network design, with shared weights in each branch. We evaluate T-NLNN on three public high-resolution remote sensing datasets, and the experimental results suggest that T-NLNN has discriminative feature learning ability and outperforms other existing algorithms. In addition, we propose a dual-anchor triplet loss function to facilitate the utilization of information in the input samples. The experimental results prove that the proposed dual-anchor triplet loss function works better than the traditional triplet loss function on all datasets.**

***Index Terms*—Deep metric learning (DML), dual-anchor triplet loss, high-resolution remote sensing image (HRRSI) retrieval, triplet nonlocal neural network (T-NLNN).**

I. Introduction

NOWADAYS, the resolution and volume of remote sensing images have considerably grown, thanks to an increasing number of satellites with their enhanced imaging capabilities [1]. How to effectively manage such large-volume data by retrieving interesting targets or scenes from massive images still remains a challenge for researchers.

Content-based remote sensing image retrieval (CBRSIR) has aroused an increasing interest of scholars in the remote sensing community and has become a popular option due to its capability in searching images from databases with query images.

Most traditional CBRSIR algorithms [2]–[5] use artificially designed features from images for retrieval, falling short of describing the complex content and rich details from high-resolution remote sensing images (HRRSI) in an accurate manner. The emergency of deep learning technology provides a new possibility to learn features of HRRSI automatically with its strong feature learning ability, and many studies have proved the remarkable performance of deep learning algorithms on CBRSIR task [6]–[17] and remote sensing image classification task [18]–[23]. In recent literature, the existing related approaches can be generally divided into two main categories: 1) approaches based on unsupervised feature learning [6]–[9], and 2) approaches based on convolutional neural networks (CNNs) [10]–[16]. Unsupervised algorithms, e.g., autoencoders, can learn image features from unlabeled data in an automatic manner, taking advantage of massive unlabeled remote sensing images and reducing the burden of manual annotation. However, unsupervised models usually have shallow networks, which limit their feature learning capabilities. CNNs are regularized versions of multilayer perceptron with more complex structures and deeper networks, leading to stronger feature learning ability and higher retrieval accuracy. Commonly used CNNs include AlexNet [24], VggNet [25], GoogleNet [26], and ResNet [27], to list a few.

Early CNNs-based CBRSIR methods directly migrated models pretrained on natural image datasets (such as ImageNet [28]) to the remote sensing domain as the feature extractor. Compared with traditional CBRSIR methods, pretrained CNN models bring a certain accuracy improvement. However, due to the large differences between the two types of images (i.e., natural pictures and remote sensing images), the features extracted by pretrained models, to some extent, fail to accurately describe the characteristics in remote sensing images, thus limiting the final retrieval accuracy. With the purpose of further enhancing the feature learning performance of pretrained CNN models, researchers tried to fine-tune it using remote sensing datasets. The experiment results showed that fine-tuned models could obtain higher retrieval accuracy. In addition, some built their own

Manuscript received October 29, 2020; revised January 2, 2021 and February 2, 2021; accepted February 6, 2021. Date of publication February 11, 2021; date of current version March 8, 2021. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB0505401; in part by the National Natural Science Foundation of China under Grants 42090012, 41771452, 41771454; and in part by the Key R&D Program of Yunnan Province in China under Grant 2018IB023. *(Corresponding author: Qimin Cheng.)*

Maoding Zhang is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: zhangmaoding@whu.edu.cn).

Qimin Cheng is with the School of Electronics Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: chengqm@hust.edu.cn).

Fang Luo is with the Department of Computer Science and Technology, Wuhan University of Technology, Wuhan 430063, China (e-mail: luof@whut.edu.cn).

Lan Ye is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: yelan0224@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3058691

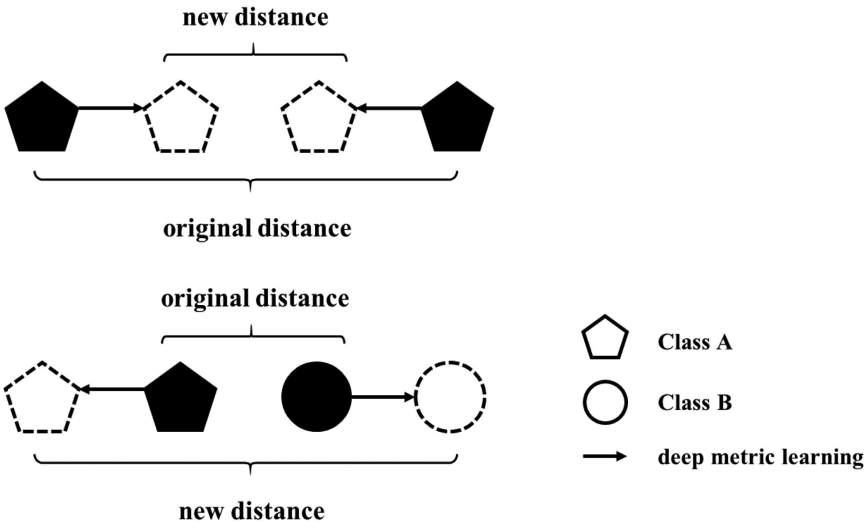

Fig. 1. Conceptulization of deep metric learning.

model and trained it from scratch with HRRSI [11]. However, training-from-scratch approaches usually demand datasets with a large volume of labeled data, which requires much labor and time. Due to its good performance, fine-tuned models are usually utilized for CBRSIR. In [10], the authors combined the fine-tuned linear convolutional layers of VGGM model and mlpconv layer to extract low-dimension image feature vector, aiming to reduce the storage space. In [13], the authors first inserted the compact bilinear pooling (CBP) to a fine-tuned CNN model. The output of CBP captures the pairwise correlation between feature channels, facilitating the retrieval performance. Despite the improved performance, these models are trained under the framework of scene classification with cross-entropy loss. In the training process, cross-entropy loss only largely focuses on the output probability corresponding to the input sample label, while ignoring the predictive information of other categories, which makes the retrieval accuracy susceptible to the intraclass difference of the image samples. To avoid this issue, deep metric learning (DML) is an emerging alternative to model training.

DML is the combination of deep learning and metric learning, whose purpose is to learn a proper measure from input data and to assist in projecting input samples to a new metric space, which is able to make similar samples closer and dissimilar samples farther (as shown in Fig. 1). DML has been widely used in the field of computer vision, such as image classification [29], video understanding [30], person reidentification [31], verification [32], and feature matching [33]. In addition, for some extreme classification tasks (i.e., many classes with only a few samples per class), DML performs considerably well [34]. Given that the purpose of DML is in accordance with remote sensing image retrieval, scholars started to apply DML to CBRSIR task [9], [35]–[42]. In [39], a triplet DML network was proposed for CBRSIR and achieved great performance on two public datasets, proving the validity of DML for the CBRSIR task. In [40], the authors proposed a TLDCNN model that combines the advantages of DML and low-dimension image feature vectors. In [42], the authors proposed a rotation invariance spatial transformation network, trained by means of a Siamese network, with the capability of extracting rotation invariance object features. However, the aforementioned methods often used fixed-size convolution kernels (usually 3×3 or 5×5) and the receptive field of each convolutional kernel is limited. In feature map generation, such fixed-size kernels only consider the local area of input with fixed sizes, thus largely ignoring the global information.

In this article, inspired by the nonlocal operation in [43], we propose a novel triplet nonlocal neural network (T-NLNN) for high-resolution CBRSIR. T-NLNN contains three branches, and each branch includes feature layers in VGG16 and a nonlocal module with shared weights. Different from conventional convolution operations with a certain size neighborhood of a certain point in images, a nonlocal module benefits capturing the global information in the image by weighting the calculation results of all points in images. The application of a nonlocal module obtains richer global information of images, leading to improved retrieval accuracy. In addition, the original triplet loss function aims to make anchor and positive images closer, anchor and negative farther in the feature space, while ignores the information between the positive and negative sample images. To fully utilize the information from input samples, we propose a dual-anchor triplet loss function modified from the original triplet loss function. During the training process, the dual-anchor triplet loss function can further shrink the distances among similar samples and enlarge the distance among dissimilar samples in the feature space. The major contributions of this article include the following two aspects:

1) We construct a novel T-NLNN for the CBRSIR task with the advantage of capturing the long-range dependence in the image.
2) We propose a dual-anchor triplet loss, aiming to make features of HRRSI more discriminative in the feature space.

The remainder of this article is organized as follows: Section II introduces high-resolution CBRSIR based on the T-NLNN

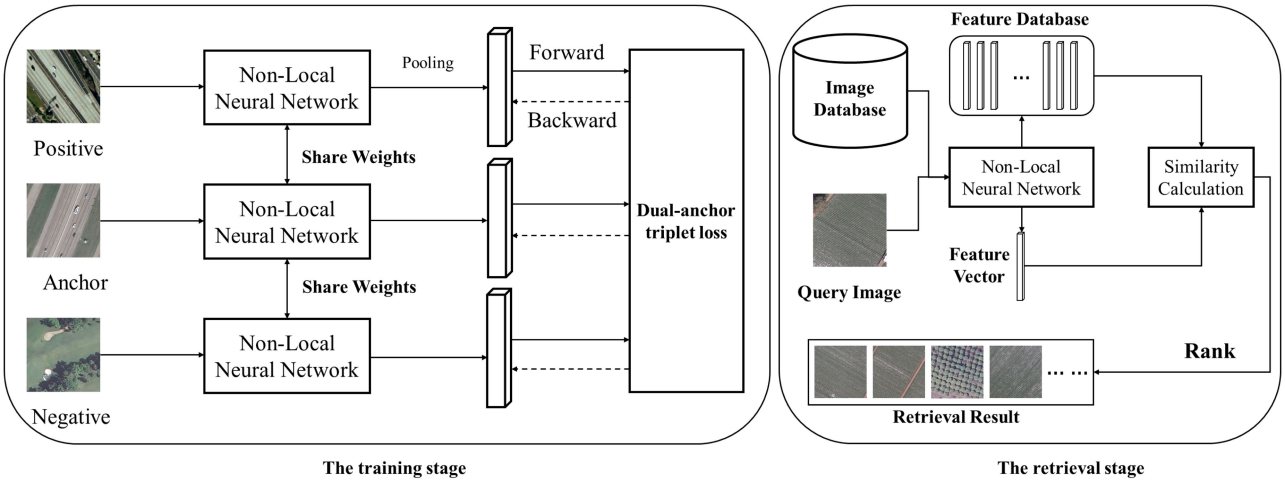

Fig. 2. General workflow of high-resolution CBRSIR based on T-NLNN.

model. Section III presents the experiments and results in detail. Section IV provides a summary of our work.

II. High-Resolution CBRSIR Based on T-NLNN

Different from conventional CNNs-based CBRSIR approaches that trained model under the framework of scene classification, T-NLNN does not focus on the classification of each input sample but focus on making samples with the same label closer and samples with different label farther in the feature space. In the following sessions, we introduce the training and retrieval stage of T-NLNN, loss function, and its optimization algorithm in detail.

A. Model Learning and Retrieval Based on T-NLNN

For DML, commonly used networks can be categorized into two-branch structures and three-branch structures. In [40], the authors conducted a comparative experiment on these two types of networks, showing that the feature extraction ability of three-branch networks is better than two-branch networks. Thus, we choose to build our model in the form of three-branch structures. To capture the long-range dependence in images, we add a nonlocal module in the network. The specific calculation formula of a nonlocal module can be presented as

$$y_i = \frac{1}{C(x)} \sum_{\forall_j} f(x_i, x_j) g(x_j) \tag{1}$$

where i is the output location index and j is any location index participating in the weighted average calculation. x and y (with the same size) mean the input and output, respectively. The f function aims to calculate the similarity between x_i and x_j, and the g function aims to calculate the feature of position j. C denotes the response factor used to normalize the calculation results. There are four commonly used functions of f: Gaussian, Embedded Gaussian, Dot Product, and Concatenation.

Fig. 2 shows the general workflow of high-resolution CBRSIR based on T-NLNN, which contains two main stages: the training stage and the retrieval stage. In the training stage, the input image samples are fed to the nonlocal neural network, followed by a global average pooling to obtain the feature vector. The corresponding loss is calculated according to the feature vector of each image sample. The calculated loss is further used to update the weight parameters through the backpropagation algorithm. The training process continues until the network parameters are converged. As shown in Fig. 2, T-NLNN follows the three-branch design with the same structure (shared weights) in each branch. The nonlocal neural network is composed of convolutional layers in the VGG16 model and the nonlocal module (we replace the classifier from the VGG16 model with the nonlocal module). The specific structure of the nonlocal module with Embedded Gaussian as the similarity calculation function is shown in Fig. 3, where input x is obtained from the last convolutional layer in VGG16. g, θ, and φ are obtained from x through 1×1 convolution operation. In this process, the channel is reduced to one-half of the original, and g, θ, and φ are compressed to obtain the corresponding 2-D matrix. g', θ', φ', θ', and $\varphi'$are multiplied to obtain f, which is later normalized by softmax function, leading to f_c. y' is the multiplication between f_c and g'. After y' is expanded to y, a 1×1 convolution is applied to y, aiming to double its channels, and the output z is obtained by adding y with x, meaning that x and z have the same dimension. In the retrieval stage, a single-branch network of trained T-NLNN is used for image feature extraction. The similarity between the two images is measured by the Euclidean distance in their feature spaces. Note that the feature vector of each image is L2 normalized before calculation. The final retrieval result is achieved by sorting the similarity calculation of all image pairs.

B. Dual-Anchor Triplet Loss

The triplet loss is the basic loss function for a triplet neural network. Its calculation formula can be described as (2), where A represents the anchor image, P represents the positive sample image with a same label as A, N represents the negative sample image with a different label, α is the margin, ω is the network parameters, and $\varphi(\cdot)$ is the L2 normalized output feature vector of the network. From the formulation, it can be seen that triplet

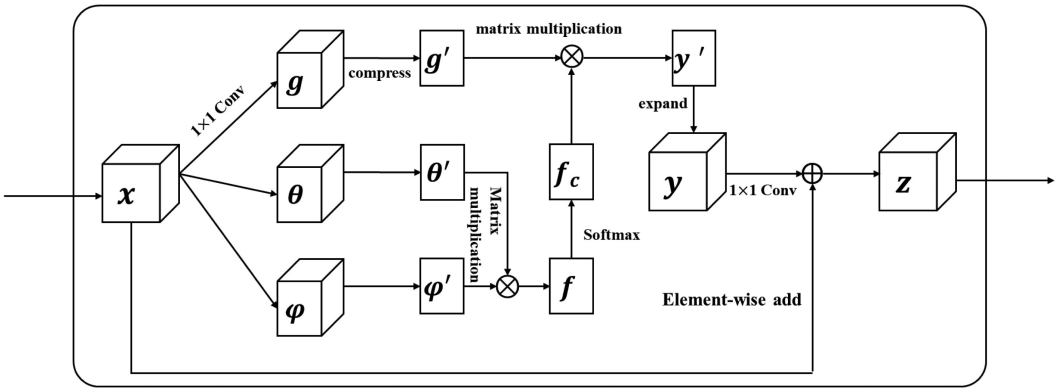

Fig. 3. Specific structure of a nonlocal module with Embedded Gaussian as the similarity calculation function.

loss aims to make $A,\ P$ closer, and $A,\ N$ farther in the feature space. However, the triplet loss ignores the information between the positive and negative sample images. In some cases, when $A,\ P$ become closer and $A,\ N$ become farther, the distance of $P,\ N$ might be shorten too. To mitigate the impact of such cases on retrieval accuracy and fully mine the information of the input images, we propose a dual-anchor triplet loss, whose formulation is shown in (3) shown at the bottom of this page. The dual-anchor triplet loss function contains three parts, with the first part the same as the original triplet loss function. The second part aims to enlarge the distance between positive samples and negative samples, and shorten the distance between positive samples and anchor samples in the feature space, where P is regarded as another anchor to calculate distance difference. To further shorten the distance between similar images, a distance constraint between A and P is added to the loss function, where λ serves as a weight parameter

$$L(A,P,N,\omega)=\frac{1}{N}\sum_{i=1}^{N}\{\max(\|\varphi(A,\omega)-\varphi(P,\omega)\|^2 - \|\varphi(A,\omega)-\varphi(N,\omega)\|^2+\alpha,0)\} \tag{2}$$

In the training process, Adam optimizer is selected for model optimization. The gradient calculation formulas of the dual-anchor triplet loss function are list below, where l_i and $h_i\ (i=1,\ 2,\ 3)$ are intermediate variables, and $d_{ij}\ (i,j=A,\ P,\ N)$ represents the Euclidean distance between sample i and j. According to (4) to (10), the gradient of the dual-anchor triplet loss function for each batch can be obtained by $\varphi(A,\ \omega)$, $\varphi(P,\ \omega)$, $\varphi(N,\ \omega)$, $\frac{\partial\varphi(A,\ \omega)}{\partial\omega}$, $\frac{\partial\varphi(P,\ \omega)}{\partial\omega}$, $\frac{\partial\varphi(N,\ \omega)}{\partial\omega}$, which are calculated in the forward and backward propagation of the network. The gradient is further used for updating network parameters. The details of the training process are shown in Algorithm 1:

$$\frac{\partial L^*(\mathrm{A,P,N},\omega)}{\partial\omega}=\frac{1}{N}\sum_{i=1}^{N}(l_1+l_2+l_3) \tag{4}$$

$$l_1=\begin{cases}h_1 d_{AP}-d_{AN}+\alpha>0\\ 0 d_{AP}-d_{AN}+\alpha<0\end{cases} \tag{5}$$

$$l_2=\begin{cases}h_2 d_{PA}-d_{PN}+\alpha>0\\ 0 d_{PA}-d_{PN}+\alpha<0\end{cases} \tag{6}$$

$$l_3=\begin{cases}h_3 d_{AP}-\alpha>0\\ 0 d_{AP}-\alpha<0\end{cases} \tag{7}$$

$$h_1=\ 2*(\varphi(A,\omega)-\varphi(P,\omega))\frac{\partial\varphi(A,\omega)-\partial\varphi(P,\omega)}{\partial\omega} -2*(\varphi(A,\omega)-\varphi(N,\omega))\frac{\partial\varphi(A,\omega)-\partial\varphi(N,\omega)}{\partial\omega} \tag{8}$$

$$h_2=\ 2*(\varphi(P,\omega)-\varphi(A,\omega))\frac{\partial\varphi(P,\omega)-\partial\varphi(A,\omega)}{\partial\omega} -2*(\varphi(P,\omega)-\varphi(N,\omega))\frac{\partial\varphi(P,\omega)-\partial\varphi(N,\omega)}{\partial\omega} \tag{9}$$

$$h_3=2*\lambda*(\varphi(A,\omega)-\varphi(P,\omega))\frac{\partial\varphi(A,\omega)-\partial\varphi(P,\omega)}{\partial\omega}. \tag{10}$$

III. Experiments and Results

Section III-A briefly introduces the datasets involved in this article. Section III-B introduces the experiment implementation details and evaluation metrics. Section III-C describes the hyperparameter settings of the dual-anchor triplet loss function and the similarity calculation of the nonlocal module. Section III-D, III-E evaluate the effectiveness of the dual-anchor triplet loss function and the T-NLNN model.

A. Datasets

The experiments in this article involve three public HRRSI datasets: UCM [44], AID [45], and PatternNet [46]. Table I gives a brief introduction regarding the class numbers, the number of images, resolution, and image sizes. Sample images of each dataset are present in Figs. 4–6.

$$L^*(A,P,N,\omega)=\frac{1}{N}\sum_{i=1}^{N}\begin{array}{l}\{\max(\|\varphi(A,\omega)-\varphi(P,\omega)\|^2-\|\varphi(A,\omega)-\varphi(N,\omega)\|^2+\alpha,0)\\ +\max(\|\varphi(P,\omega)-\varphi(A,\omega)\|^2-\|\varphi(P,\omega)-\varphi(N,\omega)\|^2+\alpha,0)\\ +\lambda*\|\varphi(A,\omega)-\varphi(P,\omega)\|^2\}.\end{array} \tag{3}$$

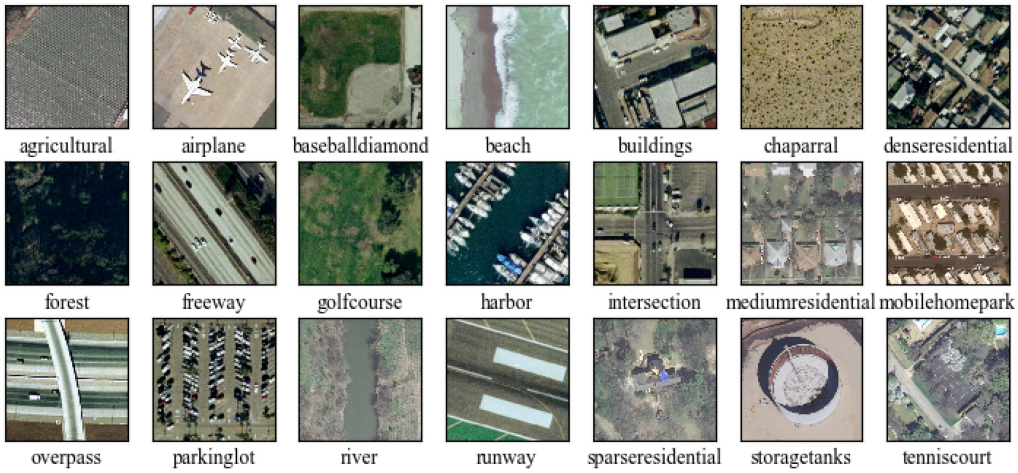

Fig. 4. Example images in the UCM dataset.

TABLE I
ATTRIBUTES OF THE DATASETS

Dataset	Classes	Images	Resolution(m)	Size
UCM	21	2,100	0.3	256 * 256
AID	30	10,000	0.5-8	600 * 600
PatternNet	38	30,400	0.062-4.693	256 * 256

Algorithm 1: Optimization for Learning T-NLNN

Input:
training sample images $\{\mathrm{A},\ \mathrm{P},\ \mathrm{N}\}$
Output:
network parameters $\{\omega\}$
Parameter setting:
learning rate ϵ, epoch κ, exponential decay rates for the moment estimates ρ_1 , ρ_2, constant δ
Initialization:
network parameters ω_0, 1^{st} moment vector m_0, 2^{nd} moment vector v_0, time step t_0
while $\kappa_t < \kappa$ do:
$t \leftarrow t + 1$
$g_t \leftarrow \frac{\partial L_t^*(A,P,N,\omega_{t-1})}{\partial \omega}$
$m_t \leftarrow \rho_1 * m_{t-1} + (1-\rho_1) * g_t$
$v_t \leftarrow \rho_2 * v_{t-1} + (1-\rho_2) * g_t * g_t$
$\widehat{m_t} \leftarrow m_t/(1-\rho_1^t)$
$\widehat{v_t} \leftarrow v_t/(1-\rho_2^t)$
$\omega_t \leftarrow \omega_{t-1} - \epsilon * \widehat{m_t}/(\sqrt{\widehat{v_t}} + \delta)$
end

B. Implementation Details and Evaluation Metrics

1) Implementation Details: The datasets used in the following experiments are divided randomly in the form of 50% for model training and 50% for testing. As for data preprocessing, we first unify the size of input images to a size of 224∗224, and then performs random flipping on them in both horizontal and vertical directions (50% probability of flipping). For each batch, the image triplets (i.e., anchor, positive, and negative image) are selected randomly. The Adam optimizer is selected for model optimization with the learning rate set to 1e-4, epoch set to 30, and batch size set to 30. All experiments are implemented with Pytorch 1.3, run on Ubuntu 16.04 with NVIDIA Geforce RTX 2080ti.

2) Evaluation Metrics: To compare T-NLNN with existing retrieval algorithms, we choose the average normalized modified retrieval rank (ANMRR), mean average precision (mAP), precision at *k* (*P@k*), and recall as the evaluation metrics to assess the retrieval performance. The definitions of these metrics are presented below.

ANMRR ranges from 0 to 1, with a lower value meaning better retrieval performance. For a query image q, $R(k)$ denotes the rank of the k th similar image in the returned image sequence, which is defined as (11). $K(q)$ is usually set as $2NG(q)$, where $NG(q)$ represents the number of similar images corresponding to image q in the whole image database. We can then obtain the average rank $AR(q)$ as shown in (12):

$$R(k) = \begin{cases} R(k), R(k) \le K(q) \\ 1.25K(q), \text{ otherwise} \end{cases} \tag{11}$$

$$AR(q) = \frac{1}{NG(q)} \sum_{k=1}^{NG(q)} R(k). \tag{12}$$

The normalized modified retrieval rank is defined as (13). After NQ queries, the ANMRR can be calculated as (14):

$$\mathrm{NMRR} = \frac{AR(q) - 0.5[1 + NG(q)]}{1.25K(q) - 0.5[1 + NG(q)]} \tag{13}$$

$$\mathrm{ANMRR} = \frac{1}{NQ} \sum_{q=1}^{NQ} \mathrm{NMRR}(q). \tag{14}$$

The recall refers to the the proportion of returned similar images to the number of all images in the image dataset. The precision P refers to the proportion of similar images to the number of all returned images. $P@k$ represents the precision at cutoff k. mAP, defined as (15), is the average of all average precision from all queries, where $\mathrm{rel}(k)$ equals 1 if the image

Fig. 5. Example images in the AID dataset.

TABLE II
MAP VALUES OF DIFFERENT HYPERPARAMETERS COMBINATION ON UCM

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$	$\alpha = 1.2$
λ =0.15	0.9136	0.9316	0.9260	0.9231	0.9060	0.9194
λ =0.2	0.9111	0.9233	0.9272	0.9190	0.9277	0.9207
λ =0.25	0.9159	0.9251	0.9336	**0.9482**	0.9371	0.9191
λ =0.3	0.8702	0.9308	0.8893	0.9394	0.9353	0.9357
λ =0.35	0.8763	0.9073	0.9309	0.9375	0.9290	0.9288

at rank k is a similar image and 0 otherwise. n is the number of the returned images and N_s represents the number of similar images in returned images.

$$\mathrm{mAP} = \frac{1}{Q}\sum_{q-1}^{Q}\frac{\sum_{k=1}^{n}(P(k) * \mathrm{rel}(k))}{N_s}. \tag{15}$$

C. Key Parameters Grid Search of the Dual-Anchor Triplet Loss and the Similarity Function Selection of the Nonlocal Module

The dual-anchor triplet loss function contains two key hyperparameters, i.e., α, λ, and their values play an important role in model training. To find an appropriate combination of parameter values, we adopt the grid search method. UCM dataset is used in the parameter section process, and the experimental results are present in Table II, where the bold value (0.9482) suggests the highest mAP obtained under different parameter value combinations. From Table II, we observe that the mAP tends to increase first and then decrease, with the increase of α or λ. We believe that, as the value of α or λ increases, the similar images are closer and dissimilar images are farther in feature space, which is conducive to the improvement of retrieval accuracy. However, when the value of α or λ becomes too large, they make the network difficult to converge, thus affecting the final retrieval accuracy. According to Table II, a hyperparameter setting of $\alpha = 0.8$ and $\lambda = 0.25$ is applied to all subsequent experiments.

Given that the similarity calculation function in the nonlocal module also plays an essential role in model training, we explore the performance of four similarity calculation functions on the UCM dataset: Concatenation, Gaussian, Embedded Gaussian, and Dot product. The experimental results are present in Fig. 7. We observe that the precision-recall curves corresponding to the four similarity calculation functions have a high degree of overlap and no particular function is superior than others. Given the slight performance advantage of Embedded Gaussian over other functions, we choose Embedded Gaussian as the similarity calculation function of the nonlocal module in subsequent experiments.

Fig. 6. Example images in the PatternNet dataset.

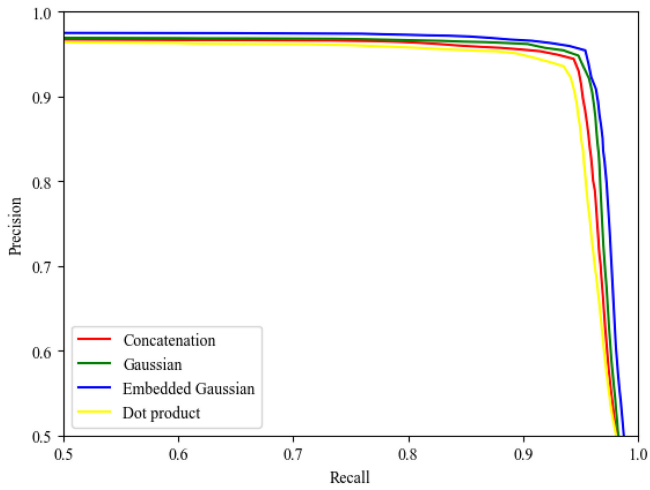

Fig. 7. Precision-recall curves of different similarity calculation functions in nonlocal module.

D. Comparison between the Dual-Anchor Triplet Loss and the Original Triplet Loss

In this section, we compare the effectiveness of two loss functions on three public datasets (UCM, AID, and PatternNet). The experimental results are present in Fig. 8. We find that the retrieval performance of the dual-anchor triplet loss has varying degrees of accuracy improvement compared with the original triplet loss on all three datasets, with the largest mAP improvement on the UCM dataset. Later, we use the model trained by the two loss functions to extract image features of the UCM dataset. To better understand the difference in feature distribution in the feature space, we use t-sne [47] to visualize image fatures obtained with different loss function by projecting the high-dimension image features into a 2-D feature space (Fig. 9). Features in (a) are obtained by T-NLNN optimized with the original triplet loss function, while features in (b) are obtained by T-NLNN optimized with the dual-anchor triplet loss function. From Fig. 9, we can see from the 2-D map that the

TABLE III
OVERALL RESULTS ON UCM

Method	Metric						
	ANMRR	mAP	P@5	P@10	P@20	P@50	P@100
VGG16_FT	0.201	0.7506	0.9053	0.8642	0.8206	0.7008	0.4224
LDCNN	0.186	0.7730	0.8474	0.8246	0.8013	0.7328	0.4288
DFLA	0.108	0.8574	0.9480	0.9227	0.8988	0.8126	0.4652
SiameseNN	0.093	0.8706	0.9309	0.9138	0.8983	0.8314	0.4753
DML(VGG16)	0.064	0.9127	0.9509	0.9398	0.9264	0.8726	0.4831
TLDCNN	0.076	0.8874	0.9179	0.9025	0.8920	0.8619	0.4836
T-NLNN+TL	0.053	0.9208	0.9482	0.9379	0.9279	0.8883	0.4891
T-NLNN+DATL	0.040	0.9482	0.9644	0.9581	0.9523	0.9277	0.4885

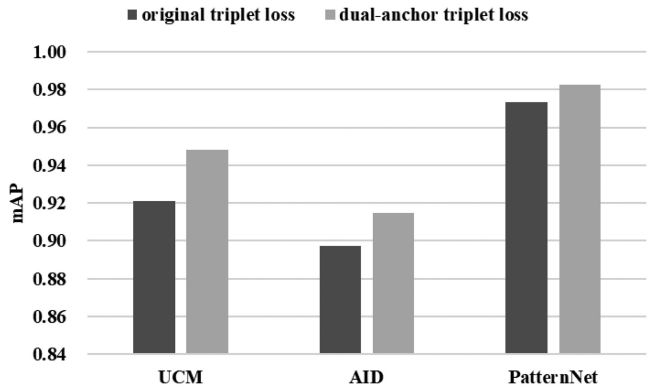

Fig. 8. mAP of the two loss functions on different datasets.

distance between dissimilar image features is larger and the distribution of similar image features is more compact in (b), compared to (a). For example, the features of sparse-residential and medium-residential (in red circle) have a certain degree of overlapping in (a), while the overlapping between the two in (b) is significantly reduced. Features of beach and chaparral (in blue circle) are distributed in long strips in (a), while the same features in (b) are distributed in a more compact manner. As highly distinguishable features generally lead to better retrieval results, our improvement based on the original triplet loss function is proven to be effective.

E. Performance Evaluation of the Proposed T-NLNN

To verify validity of the proposed T-NLNN model on high-resolution CBRSIR task, this section compares T-NLNN with six other retrieval algorithms (fine-tuned VGG16 (VGG16_FT), LDCNN [10], DFLA [17], Siamese Neural Network (SiameseNN), DML [39], and TLDCNN [40]). Three public datasets are used: UCM, AID, and PatternNet. It should be noted that all methods utilize the same data partition strategy and training hyperparameters during the experiment. VGG16_FT, LDCNN, and DFLA are trained within the framework of scene classification, and others are trained following a DML strategy. For SiameseNN and DFLA, we take the features of the VGG16 model as the backbone. For T-NLNN, we use the original triplet loss function and the dual-anchor triplet loss function, separately, to train the model. Then, we use them for subsequent retrieval evaluation. The retrieval evaluation results of each method on each dataset are present in Tables III–V ("T-NLNN+TL" means T-NLNN trained by the original triplet loss" and "T-NLNN+DATL" means T-NLNN trained by the dual-anchor triplet loss). It can be seen that VGG16_FT, LDCNN, and DFLA perform worse than other methods on all datasets, which indicate the effectiveness of DML in the CBRSIR task. From the retrieval results of SiameseNN and DML (VGG16), we observe that three-branch networks outperform two-branch networks with the same backbone. For the three-branch network, the T-NLNN model trained with the original triplet loss function performs better than DML (VGG16) and TLDCNN, while the retrieval accuracy is slightly lower on the PatternNet dataset. This phenomenon is presumably due to the high image quality in the PatternNet dataset, which leads to less demanding requirement for feature extraction. After training with the proposed dual-anchor triplet loss function, the retrieval accuracy obtained by T-NLNN model can be further improved. The above results suggest that the proposed T- NLNN method achieves higher retrieval performance that surpasses the other methods, with the largest improvement in the UCM dataset.

Besides accuracy, we also need to consider efficiency that determines a model's practicability in the retrieval process. We use the UCM dataset to evaluate the retrieval time consumption of each model. We count the time it takes for each model to finish a complete retrieval process, including the feature extraction of the query images and the return of retrieved images. The time consumption for each model is shown in Table VI. For single-branch networks, VGG16_FT and LDCNN take less time than DFLA in the image feature extraction process. Due to the addition of channels and spatial attention modules, the time taken for DFLA to extract image features has increased significantly. For the return of retrieved images, the time consumption varies little among different methods, with DML (VGG16) taking the longest retrieval time (0.042 s). All in all, we find that

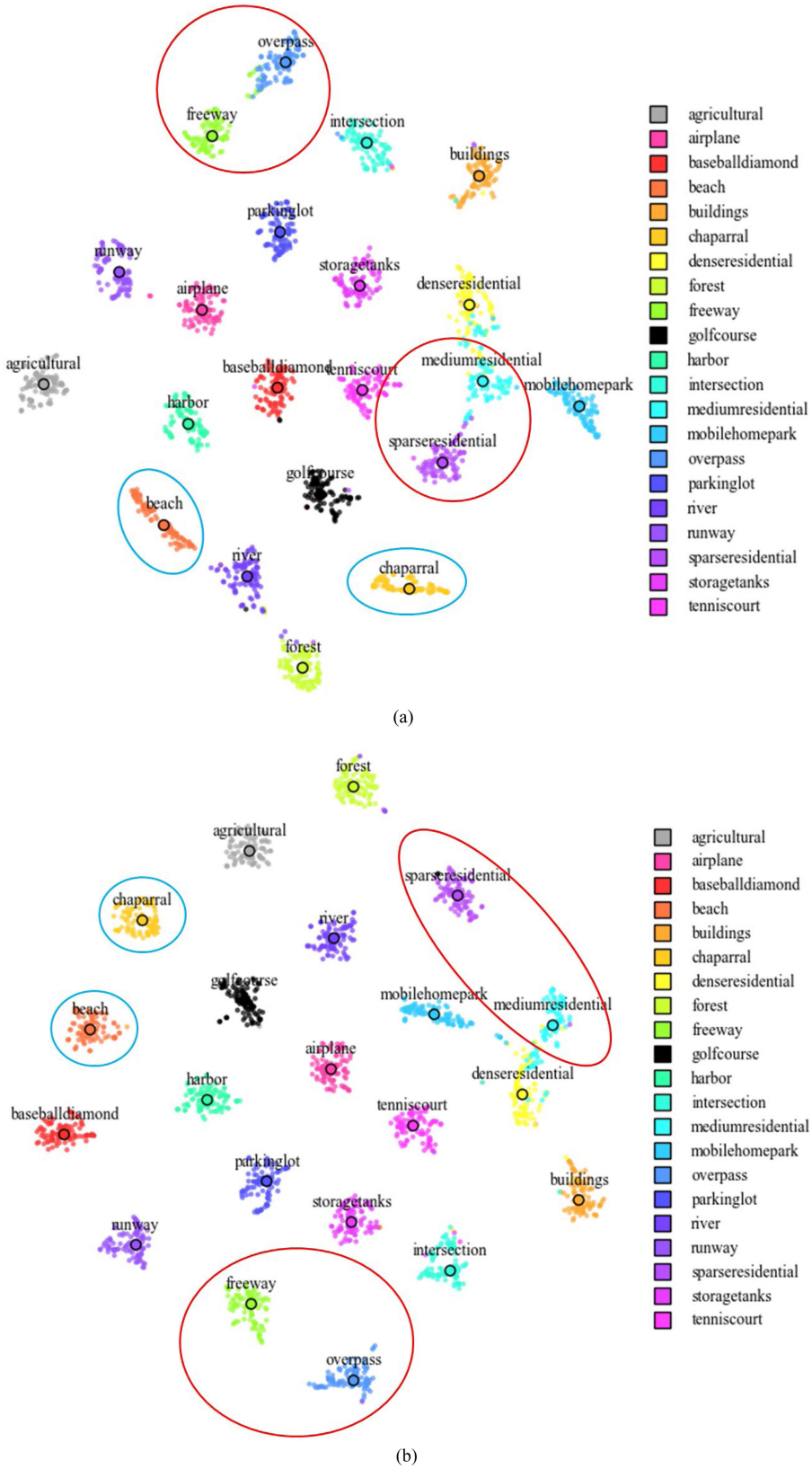

Fig. 9. Feature visualization of images on UCM dataset using t-sne. Features in (a) are obtained by T-NLNN optimized with the original triplet loss function, while features in (b) are obtained by T-NLNN optimized with the dual-anchor triplet loss function.

TABLE IV
OVERALL RESULTS ON AID

Method	Metric						
	ANMRR	mAP	P@5	P@10	P@20	P@50	P@100
VGG16_FT	0.196	0.7830	0.9144	0.8928	0.8727	0.8402	0.7972
LDCNN	0.126	0.8652	0.8951	0.8832	0.8745	0.8659	0.8562
DFLA	0.127	0.8638	0.9337	0.9193	0.9065	0.8882	0.8599
SiameseNN	0.128	0.8735	0.9070	0.8902	0.8808	0.8709	0.8524
DML(VGG16)	0.118	0.8862	0.9140	0.9007	0.8914	0.8808	0.8636
TLDCNN	0.128	0.8719	0.8983	0.8820	0.8736	0.8674	0.8518
T-NLNN+TL	0.102	0.8975	0.9225	0.9103	0.9004	0.8891	0.8739
T-NLNN+DATL	0.094	0.9146	0.9218	0.9125	0.9072	0.9036	0.8938

TABLE V
OVERALL RESULTS ON PATTERNNET

Method	Metric						
	ANMRR	mAP	P@5	P@10	P@20	P@50	P@100
VGG16_FT	0.039	0.9464	0.9881	0.9848	0.9816	0.9768	0.9720
LDCNN	0.086	0.8943	0.9248	0.9176	0.9111	0.9040	0.9013
DFLA	0.025	0.9652	0.9907	0.9880	0.9858	0.9831	0.9805
SiameseNN	0.019	0.9716	0.9883	0.9863	0.9851	0.9826	0.9796
DML(VGG16)	0.020	0.9740	0.9878	0.9853	0.9836	0.9814	0.9793
TLDCNN	0.017	0.9754	0.9865	0.9834	0.9816	0.9798	0.9786
T-NLNN+TL	0.020	0.9732	0.9883	0.9862	0.9842	0.9822	0.9802
T-NLNN+DATL	0.011	0.9827	0.9903	0.9890	0.9879	0.9870	0.9865

TABLE VI
TIME CONSUMPTION OF EACH METHOD

Method	Feature Extraction (s)	Retrieval (s)	Total(s)
VGG16_FT	5.569541	0.034000	5.603541
LDCNN	5.092187	0.026592	5.118779
DFLA	7.548865	0.041645	7.590510
SiameseNN	6.561665	0.028869	6.590534
DML(VGG16)	6.471147	0.041902	6.513049
TLDCNN	6.847029	0.026924	6.873953
T-NLNN	6.492981	0.03578	6.528761

multibranch networks bring improvement of retrieval accuracy at the expense of the reduction in efficiency. In practical applications, the retrieval accuracy and retrieval efficiency should be weighed to select the most suitable model.

In addition, we evaluated each category's retrieval accuracy of four deep metric-learning-based methods (SiameseNN, DML, TLDCNN, and T-NLNN) on each dataset. The results are present in Figs. 10 –12, where the ordinate is the ANMRR (the smaller the value, the higher the retrieval accuracy). We observe that all four methods have great retrieval performance on simple scenes such as beach, forest, and parking. In the UCM and PatternNet dataset, T-NLNN outperforms other methods in most categories, especially in buildings of different densities. In the AID dataset, T-NLNN has a greater retrieval accuracy improvement in

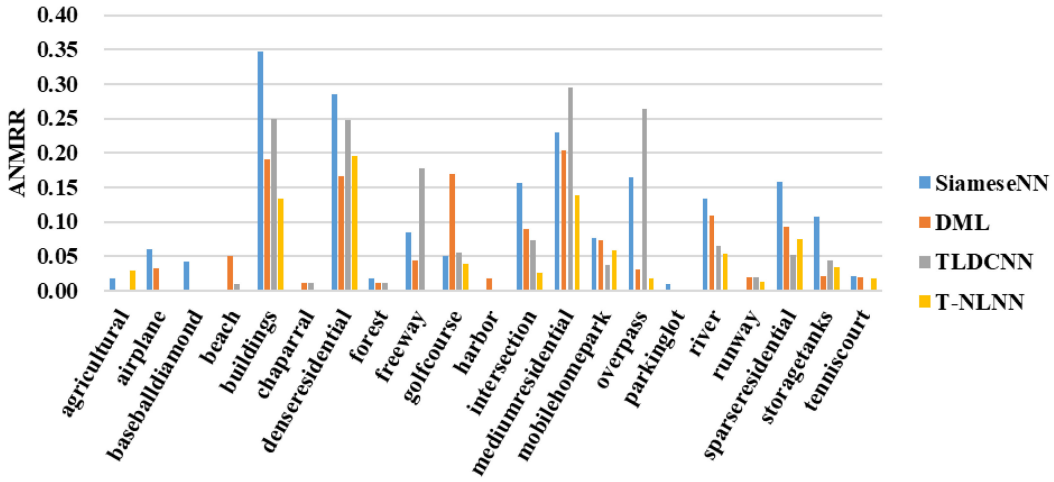

Fig. 10. Retrieval accuracy (ANMRR) of each class on UCM dataset.

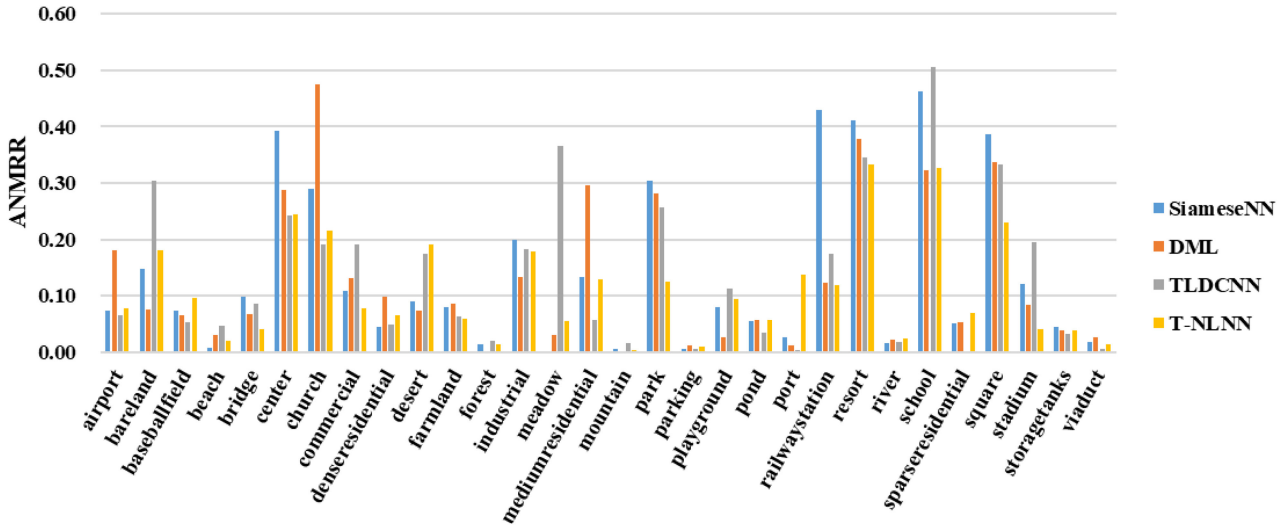

Fig. 11. Retrieval accuracy (ANMRR) of each class on AID dataset.

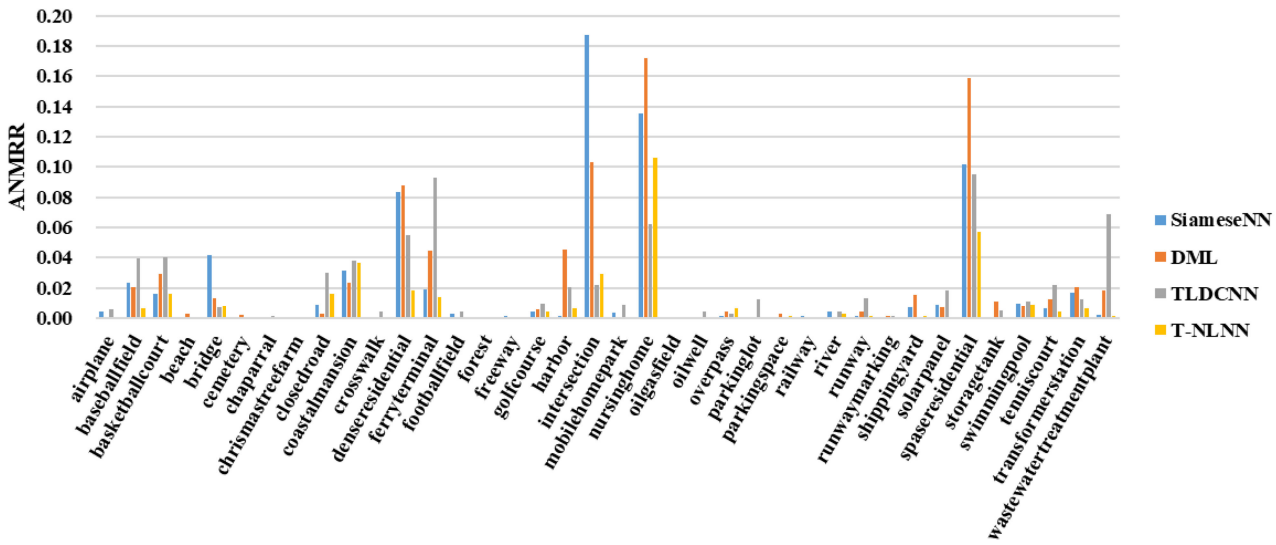

Fig. 12. Retrieval accuracy (ANMRR) of each class on PatternNet dataset.

categories such as commercial, park, and square. We believe that the images of the above categories tend to have more complex content and large objects. Better feature extraction of these images requires more attention to global information, which is the advantage of T-NLNN. At the same time, TNLNN also has its limitations, evidenced by the limited retrieval accuracy for images with simple backgrounds, e.g., bareland and desert.

IV. Conclusion

In this article, we design a T-NLNN for high-resolution CBRSIR. The proposed T-NLNN follows the three-branch network design, with shared weights in each branch. With the purpose of fully mining the input sample information, we further propose a dual-anchor triplet loss function. We evaluate T-NLNN on three public high-resolution remote sensing datasets, and the experimental results suggest that T-NLNN has discriminative feature learning ability and outperforms other existing algorithms. Experimental results further show that the proposed dual-anchor triplet loss function outperforms the original triplet loss function on all three public datasets. While T-NLNN still has its limitations, evidenced by its limited retrieval accuracy for images with simple backgrounds, we believe its advantages still outweigh its disadvantages, and we aim to solve this problem by further improving the T-NLNN design in our future work.

Acknowledgment

The authors would like to thank the editors and anonymous reviewers for their valuable comments, which helped them improve this work.

References

[1] D. Li, L. Zhang, and G.-S. Xia, "Automatic analysis and mining of remote sensing big data," *Acta Geodaetica Cartographica Sinica*, vol. 43, no. 12, pp. 1211–1216, 2014.

[2] Y. Yang and S. Newsam, "Geographic image retrieval using local invariant features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 818–832, Feb. 2012.

[3] Z. Shao, W. Zhou, L. Zhang, and J. Hou, "Improved color texture descriptors for remote sensing image retrieval," *J. Appl. Remote Sens.*, vol. 8, no. 1, 2014, Art. no. 083584.

[4] J. Yang, J. Liu, and Q. Dai, "An improved Bag-of-Words framework for remote sensing image retrieval in large-scale image databases," *Int. J. Digit. Earth*, vol. 8, no. 4, pp. 273–292, 2015.

[5] Z. Shao, W. Zhou, Q. Cheng, C. Diao, and L. Zhang, "An effective hyperspectral image retrieval method using integrated spectral and textural features," *Sens. Rev.*, vol. 35, pp. 274–281, 2015.

[6] W. Zhou, Z. Shao, C. Diao, and Q. Cheng, "High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder," *Remote Sens. Lett.*, vol. 6, no. 10, pp. 775–783, 2015.

[7] 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, Sep. 2016, Art. no. 709.

[8] X. Tang, X. Zhang, F. Liu, and L. Jiao, "Unsupervised deep feature learning for remote sensing image retrieval," *Remote Sens.*, vol. 10, no. 8, Aug. 2018, Art. no. 1243.

[9] 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.

[10] W. Zhou, S. Newsam, C. Li, and Z. Shao, "Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval," *Remote Sens.*, vol. 9, no. 5, May 2017, Art. no. 489.

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

[12] R. Imbriaco, C. Sebastian, E. Bondarev, and P. H. N. de With, "Aggregated deep local features for remote sensing image retrieval," *Remote Sens.*, vol. 11, no. 5, Jan. 2019, Art. no. 493.

[13] Y. Wang, S. Ji, M. Lu, and Y. Zhang, "Attention boosted bilinear pooling for remote sensing image retrieval," *Int. J. Remote Sens.*, vol. 41, no. 7, pp. 2704–2724, 2020.

[14] F. Ye, H. Xiao, X. Zhao, M. Dong, W. Luo, and W. Min, "Remote sensing image retrieval using convolutional neural network features and weighted distance," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 10, pp. 1535–1539, Oct. 2018.

[15] D. Hou, Z. Miao, H. Xing, and H. Wu, "Exploiting low dimensional features from the mobilenets for remote sensing image retrieval," *Earth Sci. Inform.*, vol. 13, no. 4, pp. 1437–1443, Dec. 2020.

[16] Z. Xiao, Y. Long, D. Li, C. Wei, G. Tang, and J. Liu, "High-resolution remote sensing image retrieval based on CNNs from a dimensional perspective," *Remote Sens.*, vol. 9, no. 7, Jul. 2017, Art. no. 725.

[17] W. Xiong, Y. Lv, Y. Cui, X. Zhang, and X. Gu, "A discriminative feature learning approach for remote sensing image retrieval," *Remote Sens.*, vol. 11, no. 3, Jan. 2019, Art. no. 281.

[18] J. Liang, Y. Deng, and D. Zeng, "A deep neural network combined CNN and GCN for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, Jul. pp. 4325–4338, Jul. 2020.

[19] R. Tombe and S. Viriri, "Adaptive deep co-occurrence feature learning based on classifier-fusion for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, Dec. pp. 155–164, Dec. 2021.

[20] H. Xu, H. Zhang, and L. Zhang, "A superpixel guided sample selection neural network for handling noisy labels in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, to be published.

[21] Z. Zhang, T. Jiang, C. Liu, and L. Zhang, "An effective classification method for hyperspectral image with very high resolution based on encoder-decoder architecture," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, no. 12, Apr.. pp. 1509–1519, Apr. 2021.

[22] Y. Zhang, X. Zheng, Y. Yuan, and X. Lu, "Attribute-cooperated convolutional neural network for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8358–8371, Dec. 2020.

[23] S. Zhang, G. Wu, J. Gu, and J. Han, "Pruning convolutional neural networks with an attention mechanism for remote sensing image classification," *Electronics*, vol. 9, no. 8, Aug. 2020, Art. no. 1209.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Stamford, CT, USA: Curran Associates, 2012, pp. 1097–1105.

[25] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7-9, 2015, 2015, [Online]. Available: https://arxiv.org/abs/1409.1556

[26] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[29] X. Zhe, S. Chen, and H. Yan, "Directional statistics-based deep metric learning for image classification and retrieval," *Pattern Recognit.*, vol. 93, pp. 113–123, Sep. 2019.

[30] J. Lee, S. Abu-El-Haija, and B. Varadarajan, and A. (Paul) Natsev, "Collaborative deep metric learning for video understanding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 481–490.

[31] X. Yang, P. Zhou, and M. Wang, "Person reidentification via structural deep metric learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 2987–2998, Oct. 2019.

[32] J. Wang, K. Wang, M. T. Law, F. Rudzicz, and M. Brudno, "Centroid-based deep metric learning for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 3652–3656.

[33] C.B. Choy, J.Y. Gwak, S. Savarese, and M. Chandraker, "Universal correspondence network," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA, Dec., 2016, pp. 2414–2422.
[34] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823. Accessed: Dec. 22, 2020. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2015/html/Schroff_FaceNet_A_Unified_2015_CVPR_paper.html
[35] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Comput. Vis. Image Understanding*, vol. 184, pp. 22–30, Jul. 2019.
[36] Y. Cao *et al.*, "DML-GANR: Deep metric learning with generative adversarial network regularization for high spatial resolution remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8888–8904, Dec. 2020.
[37] S. Roy, E. Sangineto, B. Demir, and N. Sebe, "Metric-learning-based deep hashing network for content-based retrieval of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 2, pp. 226–230, Feb. 2021.
[38] U. Chaudhuri, B. Banerjee, A. Bhattacharya, and M. Datcu, "CMIR-NET : A deep learning based model for cross-modal retrieval in remote sensing," *Pattern Recognit. Lett.*, vol. 131, pp. 456–462, Mar. 2020.
[39] R. Cao *et al.*, "Enhancing remote sensing image retrieval using a triplet deep metric learning network," *Int. J. Remote Sens.*, vol. 41, no. 2, pp. 740–751, Jan. 2020.
[40] Y. Boualleg, M. Farah, and I. R. Farah, "TLDCNN: A triplet low dimensional convolutional neural networks for high-resolution remote sensing image retrieval," in *Proc. Mediterranean Middle-East Geosci. Remote Sens. Symp.*, Mar. 2020, pp. 13–16.
[41] M.-S. Yun, W.-J. Nam, and S.-W. Lee, "Coarse-to-fine deep metric learning for remote sensing image retrieval," *Remote Sens.*, vol. 12, no. 2, Jan. 2020, Art. no. 2.
[42] P. Ding, S. Wan, P. Jin, and C. Zou, "A rotation invariance spatial transformation network for remote sensing image retrieval," in *Proc. 12th Int. Conf. Digit. Image Process.*, Jun. 2020, Art. no. 115191P.
[43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Non-Local_Neural_Networks_CVPR_2018_paper.html
[44] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.
[45] G.-S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.
[46] W. Zhou, S. Newsam, C. Li, and Z. Shao, "PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 197–209, Nov. 2018.
[47] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Maoding Zhang received the B.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2018, where he is currently working toward the M.S. degree in photogrammetry and remote sensing.

His research interests include remote sensing scene classification and retrieval.

Qimin Cheng (Member, IEEE) received the Ph.D. degree in cartography and geographic information system from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2004.

She is currently an Associate Professor with Huazhong University of Science and Technology, Wuhan, China. Her research interests include image retrieval and annotation, remote sensing images understanding, and analysis.

Fang Luo received the Ph.D. degree in computer science and technology from Wuhan University of Technology, Wuhan, China, in 2011.

She is currently an Associate Professor at Wuhan University of Technology, Wuhan, China. Her research interests include intelligent information processing, data fusion, and data mining.

Lan Ye received the B.S. degree in surveying engineering from the Central South University, Changsha, China, in 2019. She is currently working toward the M.S. degree in surveying engineering from Wuhan University, Wuhan, China.

Her research interests include deep learning, object detection, remote sensing scene classification, and retrieval.