High-Resolution Remote Sensing Image Retrieval Based on Classification-Similarity Networks and Double Fusion

Yishu Liu, *Member, IEEE*, Conghui Chen, Zhengzhuo Han, Liwang Ding, and Yingbin Liu

***Abstract*—In high-resolution remote sensing image retrieval (HRRSIR), convolutional neural networks (CNNs) have an absolute performance advantage over the traditional hand-crafted features. However, some CNN-based HRRSIR models are classification-oriented, they pay no attention to similarity, which is critical to image retrieval; whereas others concentrate on learning similarity, failing to take full advantage of information about class labels. To address these issues, we propose a novel model called *classification-similarity network (CSN)*, which aims for image classification and similarity prediction at the same time. In order to further improve performance, we build and train two CSNs, and two kinds of information from them, i.e., deep features and similarity scores, are consolidated to measure the final similarity between two images. Besides, the optimal fusion theorem in biometric authentication, which gives a theoretical scheme to make sure that fusion will definitely lead to a better performance, is used to conduct score fusion. Extensive experiments are carried out over publicly available datasets, demonstrating that CSNs are distinctly superior to usual CNNs and our proposed "two CSNs + feature fusion + score fusion" method outperforms the state-of-the-art models.**

***Index Terms*—Classification-similarity network (CSN), double fusion, feature fusion, high-resolution remote sensing image retrieval (HRRSIR), optimal fusion weights, score fusion.**

I. Introduction

With the rapid advancement in remote sensing (RS) sensors, the last decade has witnessed an unprecedented proliferation of high-resolution RS (HRRS) images, which have highly complex geometrical structures and spatial patterns, and are of great significance for earth observation. The urgent need to efficiently organize and manage the huge volume of HRRS images is self-evident, therefore, HRRS image retrieval (HRRSIR), which aims to find images having a similar visual content with respect to a given query from a large-scale HRRS image archive [1], has attracted more and more research interest.

In HRRSIR, there are mainly two groups of methods: the traditional ones, which are based on hand-crafted features, and the untraditional ones, i.e., deep learning methods, which can adaptively learn a hierarchical representation from data [2]. The former include Bayesian inference [3], Gibbs–Markov random fields [4], support vector machine based relevance feedback [5], active learning [6], scene semantic matching [7], local binary pattern [8], gray-level co-occurrence matrix [8], scale invariant feature transform [8]–[10], morphological texture descriptors [11], [12], bag of visual words (BoVW) [10], [12], [13], bag of spectral values [10], kernel techniques [14], graph-based models [15], and so on. And the latter's most typical example is convolutional neural network (CNN). In recent years, CNNs have had an overwhelming performance advantage over the traditional techniques, and have become the predominant method for HRRSIR.

However, in our opinion, there are some deficiencies in the existing CNN-based methods.

A. State-of-the-Art and Motivation

CNN-based HRRSIR approaches broadly fall into three categories.

1) Directly Extracting Information From Pretrained CNNs: Frequently, activations of fully connected (FC) layers of pretrained CNNs are directly used as features [16]–[19]. Sometimes, information is extracted from the convolutional layers and is reprocessed to form a holistic feature vector [18], [20]–[22]. Furthermore, multi-CNN feature fusion is carried out sometimes [18].

However, these CNNs were trained over everyday image sets instead of over HRRS datasets, and, thus, may not discover the highly intricate structures of HRRS images. What is more, they were trained for classification purposes, and, thus, similarity measurement that is of great importance for image retrieval was not considered at all during training.

2) Retraining Pretrained CNNs: Some pretrained CNNs are retrained over HRRS image sets [16], [17], [21], [23]–[27], taking the characteristics of HRRS images into account and, thus, leading to a more promising performance than the first group of methods.

However, retraining is also conducted in a multiclass classification scenario. Its training objective is different from the testing procedure of image retrieval, completely ignoring the similarity between two images. We argue that the features learned for classification may not best suit retrieval.

Manuscript received February 8, 2020; accepted March 11, 2020. Date of publication March 17, 2020. This work was supported in part by National Natural Science Foundation of China under Grant 61673184 and in part by the China Scholarship Council under Grant 201806755003. *(Corresponding author: Yishu Liu.)*

The authors are with the School of Geography, South China Normal University, Guangzhou 510631, China (e-mail: yishuliu_gz@hotmail.com; 1509298744@qq.com; 495657773@qq.com; 1021470364@qq.com; 2604245496@qq.com).

Digital Object Identifier 10.1109/JSTARS.2020.2981372

3) Integrating Similarity Learning With CNNs: Some studies [28]–[32] incorporate similarity learning into CNNs. In this case, CNNs are optimized under similarity constraints, and, thus, are more direct and natural for HRRSIR. However, the training procedure only requires weak labels (i.e., the prior knowledge whether two images come from the same class), and does not care about which classes the images belong to. Therefore, these kinds of models cannot take full advantage of the annotated information.

In a nutshell, the existing methods for HRRSIR either omit to explicitly take similarity measurement into consideration, or only utilize limited information about labels, therefore, they leave room for performance enhancement. In fact, studies in nonretrieval fields [33], [34] (see Section I-B for more details) have shown that class membership prediction and similarity learning can complement each other, and combining them will produce more discriminative features and improve CNNs' performance. Encouraged by these studies, in this article, we propose CNN models that aim for both classification prediction and similarity estimation. We call them *classification-similarity networks (CSNs)*.

Furthermore, because different CNNs carry complementary information, we build two CSNs, each of which outputs class probability predictions and similarity scores at the same time. In order to further enhance performance, we combine information from the two CSNs.

B. Related Work

1) Contrastive Loss and Metric Similarity Learning: In face recognition, Sun *et al.* [33] proposed learning CNN features at the same time for face identification, which aims to label a face image according to a set of given identities, and face verification, which aims to determine whether two face images belong to the same identity. They achieved a better performance than when face identification and face verification were dealt with separately. Moreover, in scene classification, Cheng *et al.* [34] incorporated RS scene classification and metric learning into one CNN model to improve classification performance.

Both face verification in [33] and metric learning in [34] are handled through contrastive loss. For two given images I^c and $I'^{c'}$, which pertain to the cth and c'th classes, respectively, the contrastive loss is defined as

$$\mathcal{L}_{\mathrm{c}}\left(\omega; I^c, I'^{c'}\right) = \tau(c,c')\|f - f'\|_2^2 + [1-\tau(c,c')] \times \left[\max\left(0, T - \|f - f'\|_2\right)\right]^2 \quad (1)$$

where ω denotes the network parameters (including weights and biases), f and f' are the learned deep features of I^c and $I'^{c'}$, respectively, $T > 0$ is a given margin threshold, and

$$\tau(c,c') = \begin{cases} 1, & \text{if } c = c' \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The contrastive loss (1) aims to learn a feature space in which two similar images (throughout this article, "similar" means that two images come from the same class) have a small Euclidean distance (i.e., L_2 distance), whereas two dissimilar images (throughout this article, "dissimilar" means that two images come from different classes) have a large Euclidean distance.

Noticing the relationship between Euclidean distance and similarity (a larger Euclidean distance means a smaller similarity; in fact, they can be transformed into each other over a given image set), we can regard face verification in [33] and metric learning in [34] as a similarity learning problem. More specifically, they both aim to learn metric similarity, since Euclidean distance is a metric distance. However, metric distance is subject to the rigid constraint of metric axioms (i.e., self-similarity, symmetry, and triangle inequality) [35], and several studies [36], [37] have shown that these metric axioms are epistemologically invalid for perceptual distance of human beings. As indicated by Jacobs *et al.* [38], the changeful face images cannot be matched into a metric feature space without large distortions in the distances between them, and metric similarity is less competent in robust visual recognition than nonmetric similarity.

In the light of these findings, we propose nonmetric similarity learning in this article, this is the key difference between our models and those in [33] and [34], which simultaneously take into consideration classification and metric similarity learning. Besides, we use Siamese networks and, hence, can carefully contrive a much wider variety of image pairs beforehand; whereas other researchers [33] and [34] harnessed single-branch CNNs to accomplish similarity learning, and randomly chose image pairs from among each mini-batch. Finally, our CSNs can predict and output similarity scores for image pairs and, hence, are more congenial to image retrieval; whereas the models in [33] and [34] are classification-oriented, they only predict class membership.

2) Feature Fusion and Score Fusion: Integration of information from various CNNs is an effective means of improving performance, this has been validated by many studies [18], [39]–[42].

The most common practice for multi-CNN information fusion is to consolidate features extracted from different CNNs. For example, Penatti *et al.* [39] concatenated the information from two pretrained CNNs' FC layers, Hu *et al.* [40] and Ge *et al.* [18] first encoded the activations of convolutional layers using BoVW, then combined the encoded representations. They all reported a distinct performance improvement due to feature fusion.

Stimulated by these promising results, in this article, we also propose consolidating deep features learned by individual CSNs.

We can go further than that—bear in mind that our CSNs can estimate similarity scores for any two images. In the RS community, there has been little research on score fusion. However, it has been thoroughly studied in biometric authentication [43]–[46], in which a model aiming to predict scores is called an *expert*, and score fusion is also called *expert fusion*. Some researchers [47], [48], from theoretical as well as empirical perspectives, have proved that score fusion can definitely have a positive effect on performance as long as fusion is conducted in a specific way. They found the optimal fusion theorem, which guarantees that expert fusion will necessarily lead to a better performance than any individual expert.

To further improve performance, we leverage these theoretical findings to carry out score fusion.

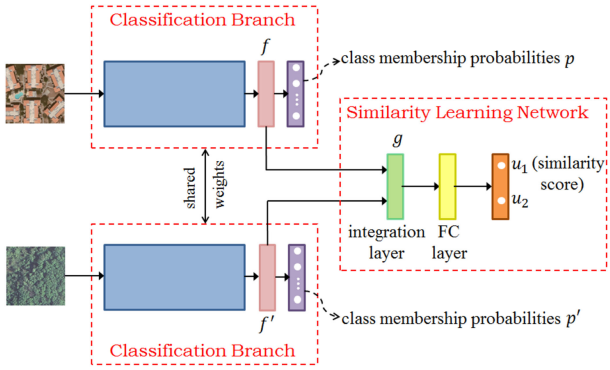

Fig. 1. Architecture sketch of CSN (figure adapted from [49]). CSN is composed of two classification branches and a similarity learning network. Two branches have the same architecture and share the same weights, and they output the class membership probabilities, with p_i indicating the probability that the corresponding input image belongs to the ith category. The similarity learning network aims to predict similarity between two input images.

In short, in this article, feature fusion and score fusion are simultaneously dealt with. Hereinafter, the phrase "double fusion" is used to indicate "feature fusion + score fusion."

C. Contributions

Overall, our contributions are fourfolded.

1) We propose a novel model called CSN, which aims to classify images and learn nonmetric similarity at the same time.
2) To make full use of network outputs, we train two CSNs and perform feature fusion as well as score fusion. As far as we know, this is the first time that two kinds of information, i.e., deep features and similarity scores, have been consolidated.
3) The theoretical discoveries in biometric authentication, which guarantee a performance improvement resulting from expert fusion, are borrowed to carry out score fusion.
4) We conduct extensive experiments over publicly available HRRS datasets, and achieve a state-of-the-art retrieval performance.

II. Proposed Method

This section presents our proposed method in detail. Architecture sketch of CSNs is introduced first; then, training procedure is elucidated; subsequently, we shed light on double fusion and investigate how to compute score fusion weights and normalization parameters; finally, we explain the retrieval process.

A. Network Framework

Fig. 1 shows the framework of our proposed CSN model. CSN is composed of two classification branches, which share exactly the same architecture as well as the same weights, and one similarity learning network.

The inputs to CSN must be a pair of images, each classification branch aims to classify the corresponding image. Suppose there are n classes, then the last layer of the first (or second) classification branch outputs an n-length vector p (or p'), which indicates the predicted probabilities that the input image pertains to each class. Moreover, we refer to the activation vector "locating in" a classification branch's penultimate layer [colored pink in Fig. 1] as the *feature vector* of the input image.

The similarity learning network consists of one integration layer, one FC layer, and one output layer. Suppose the feature vectors of two input images are f and f', respectively, then we formulate the activation vector "locating in" the integration layer as

$$g = (f - f').*(f - f') \tag{3}$$

where ".*" means element-by-element multiplication. In this way, swapping two input images will not change g, and, hence, will not change the predicted similarity between them.

Furthermore, the last layer outputs a vector $u = (u_1, u_2)^{\mathbf{T}}$ (the superscript $\mathbf{T}$ means transpose throughout the article), whose two entries predict the probabilities that two input images are similar and dissimilar, respectively. Therefore, $u_1 + u_2 = 1$, and the ground truth probability distribution is $(1\,0)^{\mathbf{T}}$ if two input images belong to the same class, and $(0\,1)^{\mathbf{T}}$ otherwise.

We use cross-entropy loss to penalize incorrect predictions, including class membership predictions and "similar-dissimilar" predictions. Correspondingly, we define two loss functions as follows:

$$\mathcal{L}_{\text{cl}}\left(\omega; I^c, I'^{c'}\right) = -\log p_c - \log p'_{c'} \tag{4}$$

and

$$\mathcal{L}_{\text{s}}\left(\omega; I^c, I'^{c'}\right) = -\tau(c, c')\log u_1 - [1 - \tau(c, c')]\log u_2. \tag{5}$$

$\mathcal{L}_{\text{s}}$ in (5) tries to "pull" u_1 toward 1 (0) if I^c and $I'^{c'}$ are similar (dissimilar). Since a larger (smaller) value of u_1 means that two input images are more (less) likely to come from the same class, u_1 indicates the extent to which I^c and $I'^{c'}$ are similar, and can be used to measure the similarity between them. Therefore, we call u_1 the *similarity score* of I^c and $I'^{c'}$.

Furthermore, from (5), it can be seen that $\mathcal{L}_{\text{s}}$ has nothing to do with metric distance, in contrast with the contrastive loss (1). So what the similarity learning network learns is "nonmetric similarity" (please see the discussion in Section I-B).

Finally, we combine (4) with (5) to define the loss for CSNs as

$$\mathcal{L}_{\text{CSN}}\left(\omega; I^c, I'^{c'}\right) = \mathcal{L}_{\text{cl}}\left(\omega; I^c, I'^{c'}\right) + \lambda\mathcal{L}_{\text{s}}\left(\omega; I^c, I'^{c'}\right) \tag{6}$$

where $\lambda \in \mathbb{R}^+$ is a tunable parameter, which determines the tradeoff between classification and similarity learning and, thus, is called the *tradeoff parameter*.

B. Training Procedure

1) Converting and Pretraining: We convert GoogLeNet [50] and ResNet50 [51] to form the classification branches of two CSNs. The purpose of converting is to reduce the dimensions of feature vectors and, thus, accelerate retrieval. More specifically, we insert four FC layers between GoogLeNet's last pooling layer and output layer, with the numbers of neurons being 512, 256, 128, and 32, respectively [see Fig. 2(a); in this way, feature

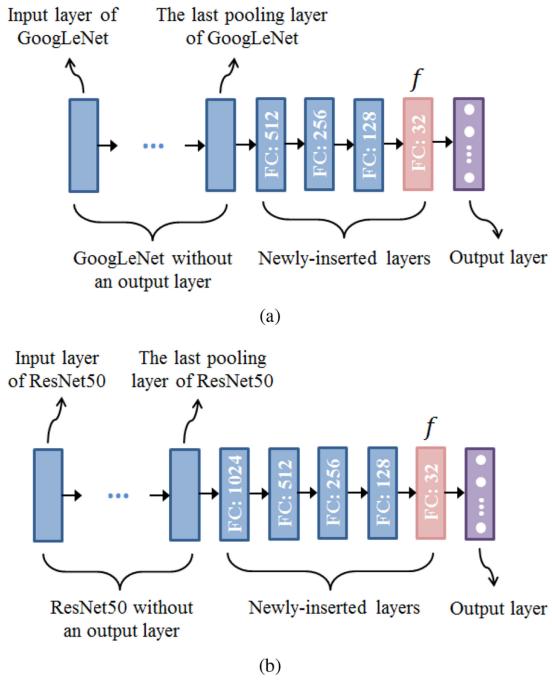

Fig. 2. Classification branch architecture. Branch networks are "adapted" from the well-known GoogLeNet and ResNet50. (a) Classification branch architecture of CSN 1. GoogLeNet is converted into the classification branch of CSN 1 to reduce feature dimensions. (b) Classification branch architecture of CSN 2. ResNet50 is converted into the classification branch of CSN 2 to reduce feature dimensions.

dimension is reduced from 1024 to 32]; and we insert five FC layers between ResNet50's last pooling layer and output layer, with the numbers of neurons being 1024, 512, 256, 128, and 32, respectively [see Fig. 2(b); in this way, feature dimension is reduced from 2048 to 32].

Hereinafter, the number "1" always relates to the CSN whose classification branch is adapted from GoogLeNet, whereas "2" always relates to the other CSN involving ResNet50.

Furthermore, for both CSNs, the FC layer in the similarity learning network has 256 neurons.

It should be stressed that batch normalization and ReLu, following each newly inserted layer, are always performed to regularize and accelerate the learning.

After new layers are added, an HRRS dataset that has sufficient images is used to pretrain our CSNs coupled with the loss (6), making the network parameters (especially the new ones) basically fit for HRRS images. Fig. 3 illustrates this process.

2) Fine-Tuning: After pretraining, two CSNs are fine-tuned. Then, the classification branches and similarity learning network are detached from each CSN, the former will be used to extract features, and the latter will be used to calculate similarity scores.

Moreover, all images in the HRRS image archive pass through classification branches 1 and 2 [note that classification branch j ($j \in \{1, 2\}$) can be either branch of CSN j] to generate feature archives 1 and 2, which are stored for later use at the retrieval stage. Besides, in order to speed up the subsequent online retrieval, we carry out feature fusion (see Section II-C for more details) to create the final feature vector archive.

The process of fine-tuning as well as feature archive generation is shown in Fig. 4.

C. Double Fusion

As mentioned above, CSNs not only learn feature representations, but also predict similarity scores. We conduct both feature fusion and score fusion, i.e., double fusion.

For a given image I (when no class labels are involved, we drop the superscript "c" to simplify notation), feature fusion involves two steps: 1) L_2-normalizing I's two feature vectors learned by both CSNs; and 2) concatenating the normalized feature vectors. We call the resulting vector the *final feature vector* of I, and denote it as $\ddot{f}(I)$.

In fact, feature fusion is quite simple. In this section, we concentrate on describing how score fusion can be performed.

Naturally, the similarity scores of similar images are unlike those of dissimilar images. We call the former *S-type scores* ("S" means "similar"), and the latter *D-type scores* ("D" means "dissimilar").

We regard the jth expert's each type of scores as the realizations of a random variable X_j^t ($j = 1, 2;\ t \in \{\mathcal{S}, \mathcal{D}\}$), with μ_j^t and σ_j^t as the expected value and standard deviation of X_j^t, respectively.

We denote by ρ^t the correlation coefficient between X_1^t and X_2^t. Let $\delta = (\delta_1\ \delta_2)^{\mathbf{T}}$, where

$$\delta_j = \mu_j^{\mathcal{S}} - \mu_j^{\mathcal{D}}, \quad j = 1, 2. \tag{7}$$

Let

$$\xi^t = (X_1^t\ X_2^t)^{\mathbf{T}}, \quad t \in \{\mathcal{S}, \mathcal{D}\} \tag{8}$$

be a two-dimensional random vector, and denote by Σ^t the covariance matrix of ξ^t.

Suppose that

$$\rho^{\mathcal{D}} = \rho^{\mathcal{S}} \tag{9}$$

and

$$\sigma_j^{\mathcal{D}} = K\sigma_j^{\mathcal{S}}, \quad j = 1, 2 \tag{10}$$

where $K \in \mathbb{R}^+$ is a constant, then

$$w = L(\Sigma^{\mathcal{S}})^{-1}\delta, \quad 0 \neq L \in \mathbb{R} \tag{11}$$

gives the optimal weights of score fusion [48].

Liu *et al.* [48] showed that only if fusion coefficients are chosen according to (11), will score fusion lead to a better performance than both the experts. This theoretical finding is important because it guarantees performance improvement.

When the condition (9) or (10) is not satisfied, we can employ the following fusion scheme instead:

$$\tilde{w} = L(\Sigma^{\mathcal{S}} + \Sigma^{\mathcal{D}})^{-1}\delta, \quad 0 \neq L \in \mathbb{R}. \tag{12}$$

Liu *et al.* [48] demonstrated that $\tilde{w}$ in (12) is an effective substitute for w in (11), yielding near-optimal fusion results.

In practice, we can estimate w and $\tilde{w}$ using validation samples.

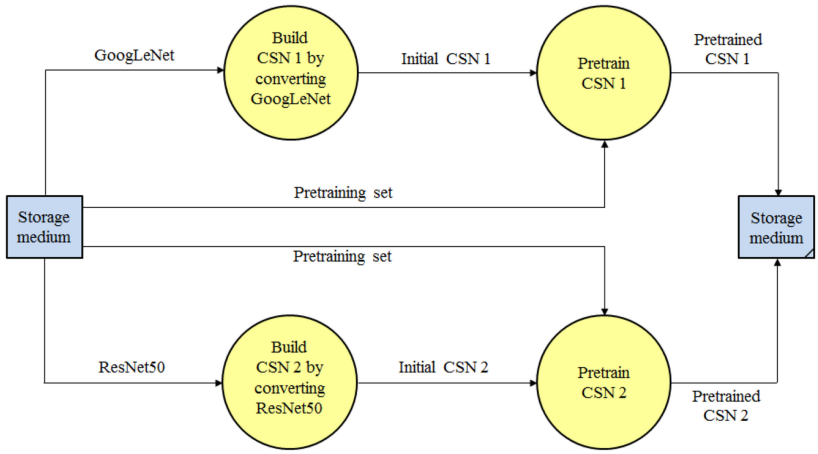

Fig. 3. Process of converting and pretraining. After converting, two CSNs are pretrained to make the network parameters (especially the new ones) basically fit for HRRS images.

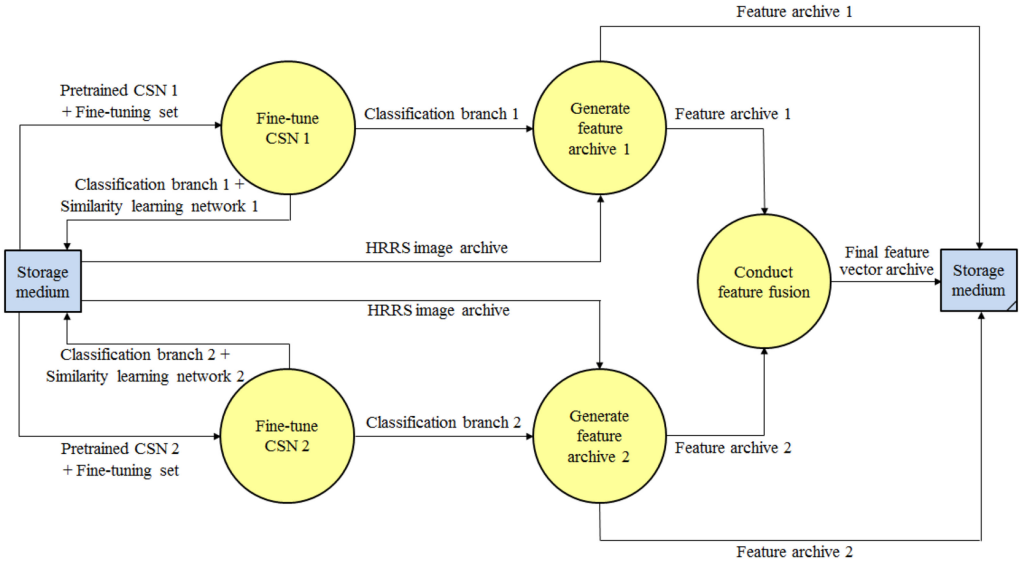

Fig. 4. Process of fine-tuning CSNs and generating feature archives. CSNs 1 and 2 are fine-tuned over the HRRS image set; then the classification branches and similarity learning network are detached from CSN j ($j = 1, 2$); subsequently, either branch of CSN j is used to extract features for the HRRS image archive, creating feature archive j. After that, feature fusion is conducted over feature archives 1 and 2, yielding the final feature vector archive. Finally, these three feature sets are stored for later use.

D. Computing Score Fusion Weights and Normalization Parameters

After fine-tuning has been finished, we can estimate fusion weights and calculate normalization parameters, preparing for double fusion.

For a given validation set $\mathfrak{V}$, let

$$A = \max_{I,J\in\mathfrak{V}} \|\ddot{f}(I) - \ddot{f}(J)\|_2. \tag{13}$$

Obviously, for any $I, J \in \mathfrak{V}$, $1 - \frac{\|\ddot{f}(I)-\ddot{f}(J)\|_2}{A} \in [0, 1]$. In this way, the distance between I and J is transformed into a similarity measure ranging from 0 to 1. We refer to A as *the normalization parameter based on feature fusion* (*NPBoFF*).

Moreover, suppose the similarity score observations over $\mathfrak{V}$ are $x_j^t = (x_{1j}^t, x_{2j}^t, \ldots, x_{m^t j}^t)^{\mathbf{T}}$ ($j = 1, 2;\ t \in \{\mathcal{S}, \mathcal{D}\}$), where $m^t \in \mathbb{N}$ is the number of similar (if $t = \mathcal{S}$) or dissimilar (if $t = \mathcal{D}$) image pairs over $\mathfrak{V}$, then we can estimate μ_j^t as follows:

$$\hat{\mu}_j^t = \frac{1}{m^t} \sum_{i=1}^{m^t} x_{ij}^t, \quad j = 1, 2;\ t \in \{\mathcal{S}, \mathcal{D}\}. \tag{14}$$

Correspondingly, we have

$$\hat{\delta} = (\hat{\delta}_1\ \hat{\delta}_2)^{\mathbf{T}} \tag{15}$$

where

$$\hat{\delta}_j = \hat{\mu}_j^{\mathcal{S}} - \hat{\mu}_j^{\mathcal{D}}, \quad j = 1, 2. \tag{16}$$

Let

$$\Phi^t = (\phi_{ij}^t)_{2\times 2}, \quad t \in \{\mathcal{S}, \mathcal{D}\} \tag{17}$$

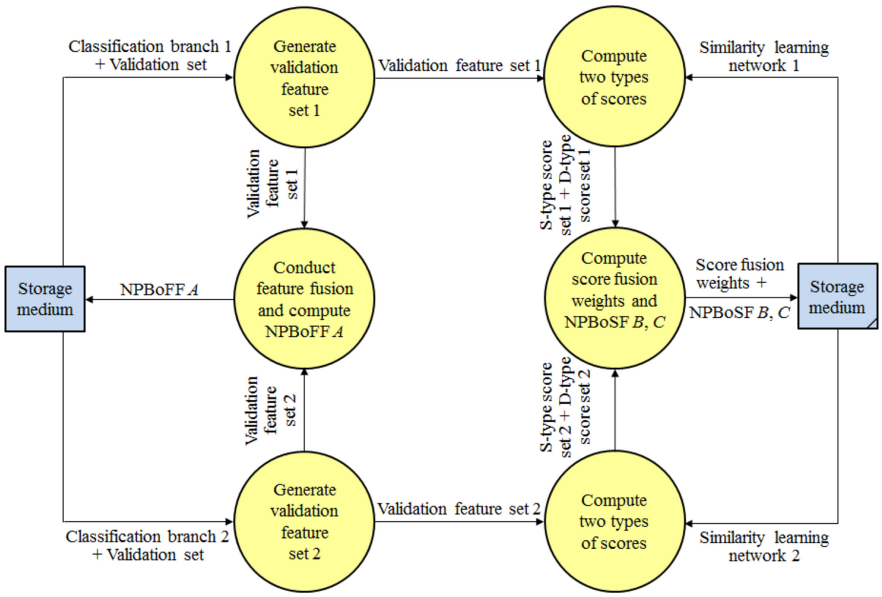

Fig. 5. Process of computing score fusion weights and normalization parameters. Classification branch j is used to extract features for the HRRS validation set, producing validation feature set j ($j = 1, 2$); then, over validation feature sets 1 and 2, feature fusion is conducted to compute the normalization parameter A. In addition, feature vector pairs from validation feature set j are fed to similarity learning network j, generating S-type score set j and D-type score set j; then, score fusion weights are computed using these four score sets. Finally, over S-type score sets 1 and 2 and D-type score sets 1 and 2, score fusion is performed to calculate the normalization parameters B and C.

where

$$\phi_{ij}^t = \sum_{k=1}^{m^t} (x_{ki}^t - \hat{\mu}_i^t)(x_{kj}^t - \hat{\mu}_j^t), \quad i, j = 1, 2;\ t \in \{\mathcal{S}, \mathcal{D}\}. \tag{18}$$

Then, an unbiased estimate of Σ^t is given by [52]

$$\hat{\Sigma}^t = \frac{1}{m^t - 1}\Phi^t, \quad t \in \{\mathcal{S}, \mathcal{D}\}. \tag{19}$$

Besides, the correlation coefficients and standard deviations are estimated, respectively, as

$$\hat{\rho}^t = \frac{\phi_{12}^t}{\sqrt{\phi_{11}^t \phi_{22}^t}}, \quad t \in \{\mathcal{S}, \mathcal{D}\} \tag{20}$$

and

$$\hat{\sigma}_j^t = \sqrt{\frac{\phi_{jj}^t}{m^t - 1}}, \quad j = 1, 2;\ t \in \{\mathcal{S}, \mathcal{D}\}. \tag{21}$$

We choose $L = 1$ in (11) and (12), then an estimate of optimal (or near-optimal) fusion weights can be expressed as

$$\hat{w} = (\hat{\Sigma}^{\mathcal{S}})^{-1}\hat{\delta} \tag{22}$$

or

$$\hat{\tilde{w}} = (\hat{\Sigma}^{\mathcal{S}} + \hat{\Sigma}^{\mathcal{D}})^{-1}\hat{\delta}. \tag{23}$$

In other words, if the two requirements

$$\hat{\rho}^{\mathcal{D}} = \hat{\rho}^{\mathcal{S}} \tag{24}$$

and

$$\hat{\sigma}_j^{\mathcal{D}} = K\hat{\sigma}_j^{\mathcal{S}}, \quad j = 1, 2 \tag{25}$$

are met, $\hat{w}$ in (22) is used as the fusion weight vector; if not, $\hat{\tilde{w}}$ in (23) is chosen instead. We refer to $\hat{w}$ and $\hat{\tilde{w}}$ as *score fusion weights*.

For convenience, we assume that $\hat{w}$ is used for score fusion. Let

$$x_j = x_j^{\mathcal{S}} \oplus x_j^{\mathcal{D}}, \quad j = 1, 2 \tag{26}$$

where $\oplus$ means vector concatenation, then score fusion over $\mathfrak{V}$ can be formulated as

$$y = (x_1\ x_2)\hat{w}. \tag{27}$$

Let

$$B = \frac{1}{\max\limits_i y_i - \min\limits_i y_i} \tag{28}$$

$$C = \frac{\min\limits_i y_i}{\min\limits_i y_i - \max\limits_i y_i} \tag{29}$$

where y_i is y's ith entry, which represents the combined similarity score of the ith image pair over $\mathfrak{V}$. Obviously, for any $i \in \{1, 2, \ldots, m^{\mathcal{S}} + m^{\mathcal{D}}\}$, $By_i + C \in [0, 1]$. We refer to B and C as *the normalization parameters based on score fusion* (*NPBoSF*).

The process of weight estimation and parameter computation is illustrated in Fig. 5. It should be stressed that this is an offline process.

E. Retrieval

Denote by $\mathfrak{A}$ the HRRS image archive, and by $|\mathfrak{A}|$ the size of $\mathfrak{A}$; denote the ith image in $\mathfrak{A}$ as I_i (since many images are involved now, we add a subscript "i" to differentiate them).

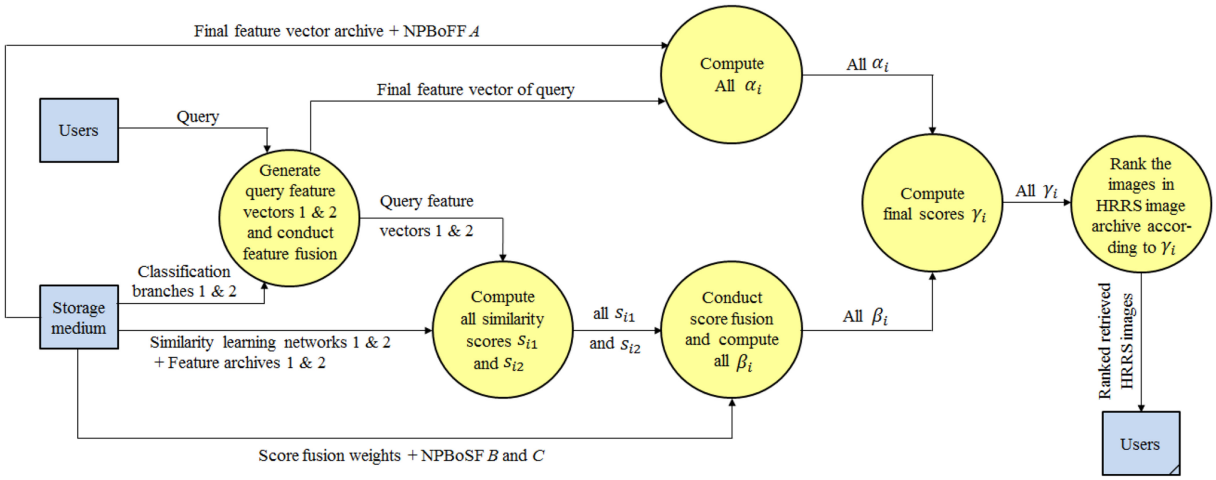

Fig. 6. Retrieval process. When a user provides a query q, classification branch j is used to extract features for q, yielding query feature vector j ($j = 1, 2$). Then, these two feature vectors are combined to generate $\ddot{f}(q)$. After that, $\ddot{f}(q)$ is compared in sequence with each vector in the final feature vector archive to calculate α_i ($i = 1, 2, \ldots, |\mathfrak{A}|$; $|\mathfrak{A}|$ is the number of HRRS images against which q queries). On the other hand, query feature vector j is paired in sequence with each vector in feature archive j and fed to similarity learning network j to calculate similarity scores s_{ij}. Then, s_{i1} and s_{i2} are combined to yield β_i. Subsequently, α_i and β_i are summed to give γ_i, which is the final score. Finally, images in the HRRS image archive are sorted according to γ_i, and several top-ranking images are presented to the user.

A query q is first fed to two classification branches to extract corresponding feature vectors f_1 and f_2 (as multiple feature vectors are involved, we also differentiate between them by adding subscripts). Then, f_1 and f_2 are consolidated to yield the final feature vector $\ddot{f}(q)$.

Let

$$\alpha_i = 1 - \frac{\|\ddot{f}(q) - \ddot{f}(I_i)\|_2}{A}, \quad i = 1, 2, \ldots, |\mathfrak{A}|. \tag{30}$$

Note that all $\ddot{f}(I_i)$ have been created beforehand and saved in the final feature vector archive. By means of deep features learned by two CSNs, α_i measures the similarity between q and I_i.

Besides, f_j is paired with each feature vector in the jth feature archive to pass through the jth similarity learning network, generating similarity scores s_{ij} ($i = 1, 2, \ldots, |\mathfrak{A}|; j = 1, 2$). Let

$$\beta_i = B(s_{i1}\ s_{i2})\hat{w} + C, \quad i = 1, 2, \ldots, |\mathfrak{A}|. \tag{31}$$

β_i measures the similarity between q and I_i by combining similarity scores output by two CSNs.

From the definitions of normalization parameters A, B, and C, it follows that both α_i and β_i roughly range from 0 to 1. Then, we define the final score of image pair $\langle q, I_i \rangle$ as

$$\gamma_i = \alpha_i + \beta_i, \quad i = 1, 2, \ldots, |\mathfrak{A}|. \tag{32}$$

Integrating two feature representations with two similarity predictions, γ_i gives a more reliable estimate of similarity than when only one CSN is employed and/or only one kind of "product" made by CSNs is utilized.

Finally, all images in the HRRS image archive are sorted in descending order according to their final scores, and those with a higher rank will be presented to users.

The aforementioned steps are summarized in Fig. 6.

III. Experimental Results

In this section, we present experimental setup first, then analyze experimental results and discuss our findings at length.

A. Experimental Setup

1) Dataset: We use three HRRS datasets: NWPU-RESISC45 (N-R) [53], PatternNet [19], and UC-Merced (UCM) [54]. N-R has 31 500 images covering 45 classes; the image size is 256×256, and the spatial resolution varies from about 30–0.2 m per pixel. The newly released PatternNet is the first publicly available image set created exclusively for HRRSIR, it has 38 classes and 30 400 images, with 800 images per class; these images are of 256×256 pixels, and have a resolution as high as 0.062–4.693 m per pixel. In UCM, there are 21 categories, each containing 100 images; the images are with the size of 256×256 pixels and 0.3 m spatial resolution. Some images from these three datasets are shown in Figs. 7–9.

The largest dataset N-R is used for pretraining, since many new network parameters are added. Both PatternNet and UCM are randomly split into subsets for fine-tuning, validation (i.e., estimating score fusion weights and normalization parameters), and test (test sets serve as HRRS image archives). Over PatternNet, the ratios of fine-tuning, validation, and test are 10%, 10%, and 80%, respectively. Over UCM, the division strategy is 40%/10%/50%. Besides, we use random horizontal and vertical image flip to expand both fine-tuning sets.

It should be stressed that some evaluation measures, such as $P@k$ (see below), depend on the size of test set. Most empirical studies [1], [18], [19], [28], [31] use 20% of the dataset PatternNet for test, therefore, for purpose of a fair comparison, we randomly divide our PatternNet test set into four equal parts, compute evaluation measures over each part separately, and report the average values.

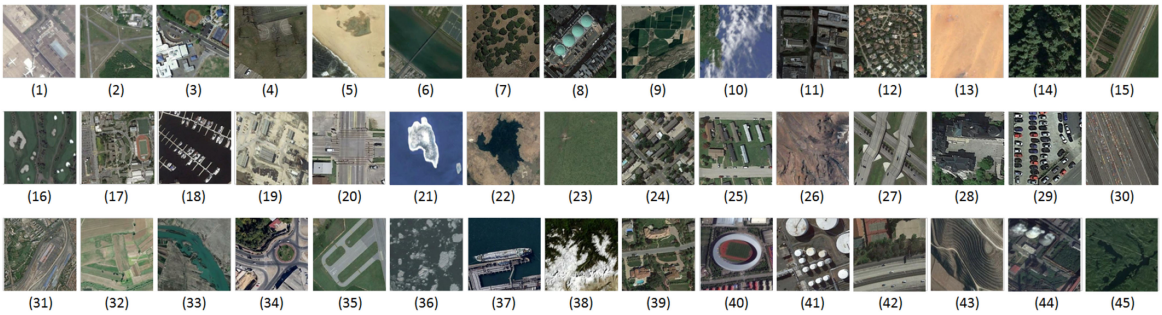

Fig. 7. Example images from the N-R dataset: (1) airplane; (2) airport; (3) baseball diamond; (4) basketball court; (5) beach; (6) bridge; (7) chaparral; (8) church; (9) circular farmland; (10) cloud; (11) commercial area; (12) dense residential; (13) desert; (14) forest; (15) freeway; (16) golf course; (17) ground track field; (18) harbor; (19) industrial area; (20) intersection; (21) island; (22) lake; (23) meadow; (24) medium residential; (25) mobile home park; (26) mountain; (27) overpass; (28) palace; (29) parking lot; (30) railway; (31) railway station; (32) rectangular farmland; (33) river; (34) roundabout; (35) runway; (36) sea ice; (37) ship; (38) snowberg; (39) sparse residential; (40) stadium; (41) storage tank; (42) tennis court; (43) terrace; (44) thermal power station; and (45) wetland.

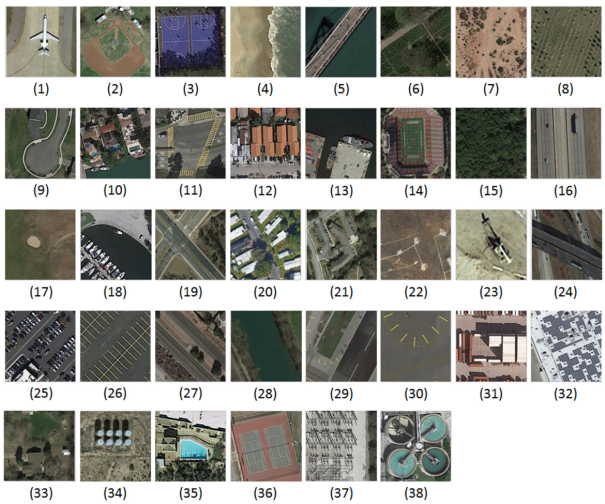

Fig. 8. Example images from the PatternNet dataset: (1) airplane; (2) baseball field; (3) basketball court; (4) beach; (5) bridge; (6) cemetery; (7) chaparral; (8) Christmas tree farm; (9) closed road; (10) coastal mansion; (11) crosswalk; (12) dense residential; (13) ferry terminal; (14) football field; (15) forest; (16) freeway; (17) golf course; (18) harbor; (19) intersection; (20) mobile home park; (21) nursing home; (22) oil gas field; (23) oil well; (24) overpass; (25) parking lot; (26) parking space; (27) railway; (28) river; (29) runway; (30) runway marking; (31) shipping yard; (32) solar panel; (33) sparse residential; (34) storage tank; (35) swimming pool; (36) tennis court; (37) transformer station; and (38) waste water treatment plant.

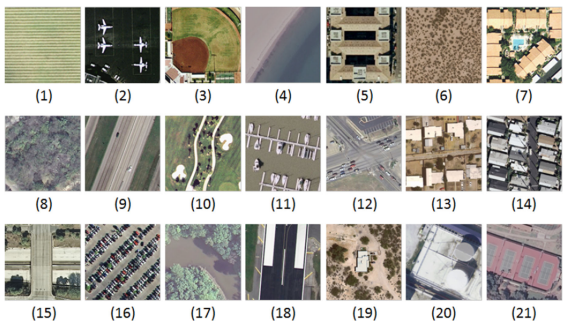

Fig. 9. Example images from the UCM dataset: (1) agricultural; (2) airplane; (3) baseball diamond; (4) beach; (5) building; (6) chaparral; (7) dense residential; (8) forest; (9) freeway; (10) golf course; (11) harbor; (12) intersection; (13) medium residential; (14) mobile home park; (15) overpass; (16) parking lot; (17) river; (18) runway; (19) sparse residential; (20) storage tank; and (21) tennis court.

Furthermore, we create image pairs in such a way that each class provides the same number of similar image pairs and each class combination provides the same number of dissimilar image pairs. Similar (dissimilar) image pairs are randomly selected within each class (class combination). Since there are a large variety of class combinations, especially when a dataset has many classes, we create much more dissimilar image pairs than similar ones. Besides, we set the ratio between similar and dissimilar image pairs to $1 : [1 + 0.25(i - 1)]$ in the ith epoch.

The details of datasets are summarized in Table I.

2) Development Environment: All the numerical experiments are performed on an AMAX workstation, which has two Intel Xeon E5-2640Wv4 CPUs with ten cores, two NVIDIA Titan X GPUs, and a 128-GB memory. MatConvNet [55] acts as our development platform.

We pretrain/fine-tune CSNs using stochastic gradient descent [56], with a batch size of 64, a weight decay of 0.0005, and a momentum of 0.9. During pretraining, learning rates for new layers and "old" ones are set to 0.01 and 0.001, respectively; during fine-tuning, learning rate for all layers is 0.0001.

3) Evaluation Measures: Four measures are used to evaluate retrieval performance: average normalized modified retrieval rank (ANMRR), mean average precision (mAP), precision at cutoff k ($P@k$), and the interpolated precision-recall (P-R) curve. Their definitions can be found in [1].

Moreover, evaluation is performed by using each test image to query against the rest of the test images and reporting the average.

B. Effect of Tradeoff Parameter on Performance

Different values of the tradeoff parameter λ are tried, and the corresponding results are shown in Tables II and III. The relationship between λ and mAP is also presented in Fig. 10 for a more intuitive understanding. It can been seen that a too large or small value of λ will weaken CSNs, indicating that classification prediction and similarity learning jointly contribute to performance enhancement and there is a proper tradeoff between them. From another perspective, this demonstrates the rationality of taking classification and similarity learning into account at the same time.

TABLE I
DETAILS OF DATASETS

Dataset	# Classes	# Images per Class	# Total Images	# Training Images	# Validation Images	# Test Images	# Image Pairs for Training	# Image Pairs for Validation
N-R	45	700	31.5 K	31.5 K	-	-	600 K[1], 3 M[2]	-
PatternNet	38	800	30.4 K	3.04 K	3.04 K	24.32 K	100 K, 500 K	5 K , 10 K
UCM	21	100	2.1 K	0.84 K	0.21 K	1.05 K	50 K, 250 K	420, 840

[1]The number of similar image pairs.
[2]The number of dissimilar image pairs.

TABLE II
VALUES OF λ VERSUS PERFORMANCE (OVER PATTERNNET)

	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
ANMMR	0.0389	0.0260	0.0121	0.0133	**0.0027**	0.0036	0.0138	0.0255	0.0127
mAP (%)	94.89	96.48	98.31	97.85	**99.56**	99.39	97.79	96.51	98.15

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP.

TABLE III
VALUES OF λ VERSUS PERFORMANCE (OVER UCM)

	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
ANMMR	0.0621	0.0476	0.0614	0.0347	0.0338	**0.0211**	0.0282	0.0282	0.0584
mAP (%)	91.21	93.26	91.66	94.79	94.90	**96.71**	95.75	95.81	92.01

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP.

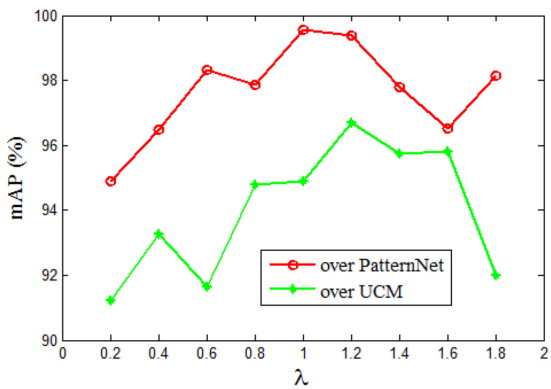

Fig. 10. Tradeoff parameter λ versus performance. If a proper tradeoff between image classification and similarity learning can be found, we will achieve the best performance from CSNs. It turns out that the optimal values of λ are 1.0 and 1.2 over PatternNet and UCM, respectively.

It turns out that when $\lambda = 1.0$ and $\lambda = 1.2$, retrieval performance reaches its optimum over PatternNet and UCM, respectively. In the next several sections, the results reported are all based on the best choices of λ.

C. Comparison Among Models

We compare the following nine models.

1) fine-tuning GoogLeNet and extracting features from the last pooling layer to compute similarity (we denote this model as "Fine-tuned GoogLeNet");
2) extracting feature vectors from CSN 1 to compute similarity (we denote this model as "1: FV," with "1" meaning the first CSN and "FV" meaning "feature vectors");
3) using the similarity scores output by CSN 1 as a similarity measure (we denote this model as "1: SS," with "SS" meaning "similarity scores");
4) fine-tuning ResNet50 and extracting features from the last pooling layer to compute similarity (we denote this model as "Fine-tuned ResNet50");
5) extracting feature vectors from CSN 2 to compute similarity (we denote this model as "2: FV," with "2" meaning the second CSN);
6) using the similarity scores output by CSN 2 as a similarity measure (we denote this model as "2: SS");
7) conducting feature fusion over two CSNs and, then, using the combined features to compute similarity (we denote this model as "1&2: FF," with "FF" meaning "feature fusion");
8) conducting score fusion over two CSNs and, then, using the combined scores as a similarity measure (we denote this model as "1&2: SF," with "SF" meaning "score fusion");
9) conducting double fusion over two CSNs and, then, using the combined features and scores to compute similarity (we denote this model as "1&2: DF," with "DF" meaning "double fusion").

The retrieval performance in terms of ANMMR, mAP, and $P@k$ ($k = 5, 10, 50, 100, 1000$) is shown in Tables IV and V. Besides, Fig. 11 presents the interpolated P-R curves. Based on these results, our observations and analyses are listed as follows.

1) Our proposed method, i.e., model 9, apparently outperforms the other eight models over both HRRS datasets, regardless of which evaluation measure is used. This

TABLE IV
PERFORMANCE COMPARISON AMONG MODELS OVER PATTERNNET

	Model 1 Fine-tuned GoogLeNet	Model 2 1: FV	Model 3 1: SS	Model 4 Fine-tuned ResNet50	Model 5 2: FV	Model 6 2: SS	Model 7 1&2: FF	Model 8 1&2: SF	Model 9 1&2: DF
ANMMR	0.1271	0.1102	0.1263	0.0923	0.0718	0.0959	0.0245	0.0513	**0.0027**
mAP (%)	84.28	86.28	84.33	88.14	90.77	88.02	96.62	93.36	**99.56**
P@5	95.51	95.82	95.57	96.33	97.05	96.05	98.99	97.78	**99.61**
P@10	94.69	95.11	94.75	95.89	96.63	95.42	98.88	97.40	**99.58**
P@50	91.29	92.33	91.37	93.63	94.88	93.18	98.35	96.08	**99.57**
P@100	87.35	88.92	87.42	90.73	92.81	90.25	97.67	94.75	**99.55**
P@1000	15.37	15.49	15.38	15.61	15.69	15.60	15.85	15.76	**15.90**

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP and $P@k$.

TABLE V
PERFORMANCE COMPARISON AMONG MODELS OVER UCM

	Model 1 Fine-tuned GoogLeNet	Model 2 1: FV	Model 3 1: SS	Model 4 Fine-tuned ResNet50	Model 5 2: FV	Model 6 2: SS	Model 7 1&2: FF	Model 8 1&2: SF	Model 9 1&2: DF
ANMMR	0.2645	0.1667	0.2399	0.1657	0.1466	0.1994	0.0347	0.0618	**0.0211**
mAP (%)	67.59	77.71	70.13	77.85	82.19	74.33	94.76	91.61	**96.71**
P@5	85.22	88.85	86.13	88.92	90.85	87.63	97.35	95.82	**97.79**
P@10	81.56	86.49	82.80	86.61	88.96	84.84	96.81	95.01	**97.60**
P@50	62.30	71.93	64.72	72.05	77.21	68.71	90.52	86.55	**93.27**
P@100	38.83	43.73	40.06	43.79	44.96	42.09	48.23	47.21	**48.63**
P@1000	4.90	4.90	4.90	4.90	4.90	4.90	4.90	4.90	**4.90**

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP and $P@k$.

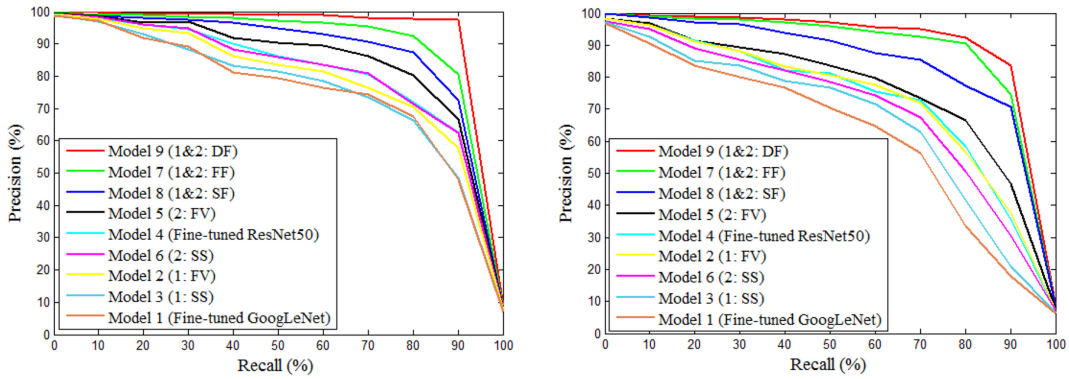

Fig. 11. Interpolated P-R curves. A curve farther away from the origin indicates a better performance. (a) Over PatternNet. (b) Over UCM.

demonstrates that our "CSNs + double fusion" approach is effective.

2) Whichever evaluation measure we choose, model 7 (i.e., conducting feature fusion over two CSNs) and model 8 (i.e., conducting score fusion over two CSNs) invariably rank second and third, respectively, falling behind our proposed method but staying ahead of nonfusion models 1 to 6. This indicates that multi-CNN information fusion is indeed advantageous and double fusion is superior to single fusion.
3) Specially, in terms of performance, model 8 (i.e., conducting score fusion over two CSNs) is always better than models 3 and 6, which use the similarity scores predicted by only one CSN. This validates the claim that the optimal score fusion scheme guarantees performance improvement.
4) Model 2 whose classification branches are "adapted" from GoogLeNet surpasses model 1, the fine-tuned GoogLeNet; similarly, model 5 whose classification branches are "adapted" from ResNet50 surpasses model 4, the fine-tuned ResNet50. Noting that these four models are all stand-alone and they concern deep features rather than similarity scores, we reach the conclusion that our proposed CSNs can learn stronger and more

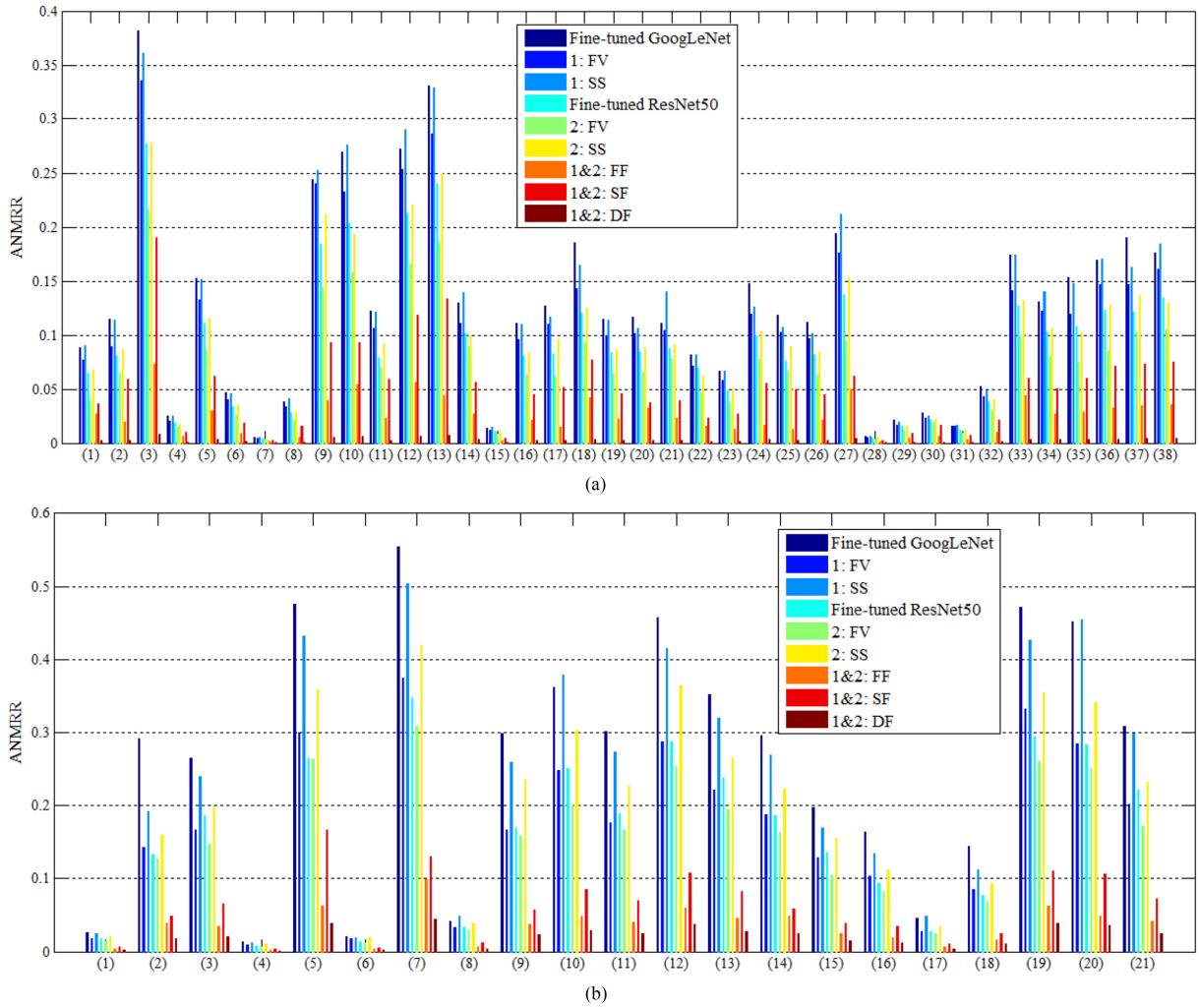

Fig. 12. ANMRR per class. From a class-level perspective, we also find that CSNs are frequently superior to usual CNNs and our proposed "two CSNs + double fusion" model performs best. Note that only class numbers are given here due to limited space, the class names can be found in Figs. 8 and 9. (a) Over PatternNet. (b) Over UCM.

discriminative features than usual CNNs. In other words, giving simultaneous consideration to classification and similarity learning makes CSNs powerful.

5) Model 2 outperforms model 3; similarly, model 5 outperforms model 6. These results show that deep features learned by a CSN provide more essential and useful information for image retrieval than similarity scores predicted by the CSN. A possible explanation may be that only two neurons are used in the output layer of CSNs, lacking fine-grained ranges of similarity and, thus, making a too rough prediction.
6) The fine-tuned ResNet50 is superior to the fine-tuned GoogLeNet, correspondingly, the models based on ResNet50 are also better than their counterparts based on GoogLeNet (model 5 versus model 2, model 6 versus model 3). Therefore, a well-performed branch will make a positive contribution to CSNs.

Furthermore, the retrieval time our proposed method (i.e., model 9) needs is acceptable, although it involves two CSNs and carries out not only feature fusion but also score fusion. In fact, our model's final feature vectors are of lower dimension (64-D) compared to the original GoogLeNet (1024-D) and ResNet50 (2048-D). Besides, feature vector pairs can be fed in batch to a similarly learning network, which is quite shallow, and GPUs can be used to accelerate computation. So predicting similarity scores does not cost much time. As for score fusion and computation of final scores, the time they consume is negligible.

The average retrieval time of our proposed model is 0.24 and 0.05 s per query over PatternNet and UCM, respectively, in contrast to 0.20 and 0.04 s for the fine-tuned ResNet50. Certainly, feature indexing techniques such as hashing [14], [29], [30], [57] and vocabulary trees [58], [59] can be used to speed up online retrieval. This, however, lies beyond the scope of this article.

Finally, the pretraining time for CSNs 1 and 2 is about 7 and 9 h, respectively; over PatternNet, the fine-tuning time for CSNs 1 and 2 is about 4 and 6 h, respectively; and over UCM, the fine-tuning time for CSNs 1 and 2 is about 2 and 3 h, respectively.

TABLE VI
COMPARISON WITH STATE-OF-THE-ART METHODS OVER PATTERNNET

Method	ANMRR	mAP (%)	Feature Dimension	Size of Training Set
VGG-F [19]	0.2995	63.37	4096	**0%**
VGG-S [19]	0.2961	63.74	4096	**0%**
GoogLeNet-m [18]	0.2784	65.98	832	**0%**
ResNet50 [19]	0.2584	68.23	2048	**0%**
LDCNN [19]	0.2416	69.17	**38**	80%
SBS-CNN [1]	0.2185	72.24	256	70%
Siamese graph CNN [32]	0.21	81.79	256	75%
Re-ranking [27]	0.0118	98.49	2048	82%
Distribution consistency [31]	-	99.43	2048	80%
DML [28]	0.0030	99.55	2048	80%
CSNs + Double fusion [this work]	**0.0027**	**99.56**	64	10%

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP.

TABLE VII
COMPARISON WITH STATE-OF-THE-ART METHODS OVER UCM

Method	ANMRR	mAP (%)	Feature Dimension	Size of Training Set
ResNet101 [19]	0.356	-	2048	**0%**
Fine-tuned VGG-M [17]	0.329	-	4096	**0%**
Siamese graph CNN [32]	0.30	69.89	256	75%
Multiple feature combinations [18]	0.2915	-	35800	**0%**
GoogLeNet [21]	0.285	-	1024	**0%**
SBS-CNN [1]	0.2683	67.12	256	**0%**
SatResNet-50 [16]	0.239	69.94	2048	**0%**
Discriminator NW [61]	0.09	81.20	512	80%
CNN features + Weighted distance [23]	0.0404	-	2048	80%
Adversarial Hah-code learning [62]	-	93.33	4096	80%
Aggregated deep local features [26]	-	94.90	256	80%
Re-ranking based on CNNs [27]	0.0359	95.61	2048	90%
DML [28]	0.0223	96.63	2048	50%
DHNN [29]	-	**97.62**	4096	88%
CSNs + Double fusion [this work]	**0.0211**	96.71	**64**	40%

A smaller value of ANMRR indicates a better performance, and the opposite is true for mAP.

D. ANMRR Per Class

Fig. 12 presents ANMRR per class. Our observations and analyses are given as follows.

1) Broadly speaking, class-level ANMRR conforms to our findings discussed in Section III-C, such as that our proposed model performs best, and double fusion prevails over single fusion, whereas single fusion prevails over nonfusion.
2) Values of ANMRR vary greatly across classes, indicating that images from different categories may be "close together" (i.e., highly similar), at the same time, some classes are far away from others.
3) Generally, natural scenes have a better retrieval performance than man-made ones. For example, all the five natural scene classes over PatternNet (beach, chaparral, Christmas tree farm, forest, and river) have small ANMRR values; the five groups of lowest ANMRR over UCM are all yielded by natural scene classes (agricultural, beach, chaparral, forest, and river). This is consistent with the previous finding that natural scene images are easily distinguishable and, hence, have a high classification accuracy [34], [41], [60].

E. Comparison With State-of-the-Art Methods

In this section, we make a comparative study. Since deep features frequently have a huge performance advantage over the traditional hand-crafted features, we only make a comparison with CNN-based methods. Moreover, besides ANMRR and mAP, feature dimension and the size of training set are also listed for the sake of fairness and objectivity.

The numerical results in Table VI reveal that over PatternNet, our proposed method outperforms all existing models in terms of retrieval performance, even in the case where CSNs use a much smaller training set (note that the size of our training set is only 10%, whereas many sizes in Table VI are bigger than 70%). Generally, our features are more compact, requiring less storage space and feature comparison time.

The numerical results about UCM are summarized in Table VII. It turns out that in terms of retrieval performance, our model is superior to all existing methods with the exception

of DHNN [29]—our mAP is lower than DHNN's by 0.91%. However, our training set is much smaller (40% versus 88%), and our feature vectors are much shorter (64-D versus 4096-D).

To sum up, our proposed "CSNs + double fusion" model is more competitive than the state-of-the-art approaches.

IV. Summary and Future Work

Most of the existing CNN-based HRRSIR models are classification-oriented, and give little consideration to similarity learning, which is very important for image retrieval. In contrast, others only "concern" themselves with similarity learning, failing to make full use of information about class labels. To address these issue, we propose a novel CNN model called CSN, which aims to classify images and learn similarity simultaneously. Moreover, two kinds of information from multiple CSNs, i.e., deep features and similarity scores, are combined. Besides, the optimal fusion theorem in biometric authentication, which will definitely lead to a better performance, is used to conduct score fusion.

To validate our models, we have conducted extensive experiments, and the experimental results reveal that 1) our proposed CSNs, which take into consideration both image classification and similarity learning, can learn more powerful and discriminative features than usual CNNs; 2) multi-CNN information fusion boosts retrieval performance, double fusion outperforms single fusion, and single fusion outperforms nonfusion; 3) the optimal score fusion scheme indeed guarantees performance improvement; and 4) our "multiple CSNs + feature fusion + score fusion" approach has achieved a better performance than the existing models, and, generally, the resulting features are much more compact.

However, our present strategies leave room for further improvement. For example, it is of interest to incorporate hard example mining [63] into the training process to leverage hard image pairs. Besides, the tradeoff parameter λ has been tentatively selected, consuming much time; it might be automatically learned instead. Furthermore, because the classification branches of CSNs are based on GoogLeNet and ResNet50, which were both pretrained on everyday RGB images, our proposed methods are validated on optical HRRS images that have three channels, just like everyday images. We will extend our methods to other types of RS data, such as hyperspectral images, SAR images, and time series. Finally, hashing techniques can be integrated with CSNs to expedite the retrieval process. These form the focus of our future research.

Acknowledgment

Y. Liu would like to express her gratitude to CENPARMI for its warm hospitality. All the authors would like to thank the editors and anonymous reviewers for their valuable and constructive comments.

References

[1] 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.*, to be published, doi: 10.1109/TGRS.2020.2984703.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Sep. 2015.

[3] M. Datcu, K. Seidel, and M. Walessa, "Spatial information retrieval from remote-sensing images—Part I: Information theoretical perspective," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 5, pp. 1431–1445, May 1998.

[4] M. Schröder, H. Rehrauer, K. Seidel, and M. Datcu, "Spatial information retrieval from remote-sensing images—Part II: Gibbs-Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 5, pp. 1446–1455, May 1998.

[5] M. Ferecatu and N. Boujemaau, "Interactive remote-sensing image retrieval using active relevance feedback," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 818–826, Apr. 2007.

[6] B. Demir and L. Bruzzone, "A novel active learning method in relevance feedback for content-based remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2323–2334, May 2015.

[7] M. Wang and T. Song, "Remote sensing image retrieval by scene semantic matching," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2874–2886, May 2013.

[8] 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.

[9] 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. 2013.

[10] 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.

[11] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 3023–3034, May 2014.

[12] E. Aptoula, "Bag of morphological words for content-based geographical retrieval," in *Proc. IEEE 12th Int. Workshop Content-Based Multimedia Indexing*, Jun. 2014, pp. 1–5.

[13] 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, Apr. 2015.

[14] B. Demir and L. Bruzzone, "Hashing-based scalable remote sensing image search and retrieval in large archives," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 892–904, Feb. 2016.

[15] Y. Wang *et al.*, "A three-layered graph-based learning approach for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6020–6034, Oct. 2016.

[16] P. Napoletano, "Visual descriptors for content-based retrieval of remote sensing images," *Int. J. Remote Sens.*, vol. 39, no. 5, pp. 1343–1376, May 2018.

[17] 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.

[18] Y. Ge, S. Jiang, Q. Xu, C. Jiang, and F. Ye, "Exploiting representations from pre-trained convolutional neural networks for high-resolution remote sensing image retrieval," *Multimed. Tools Appl.*, vol. 77, no. 13, pp. 17 489–17 515, Jul. 2018.

[19] 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.

[20] 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.

[21] G.-S. Xia, X.-Y. Tong, F. Hu, Y. Zhong, M. Datcu, and L. Zhang, "Exploiting deep features for remote sensing image retrieval: A systematic investigation," Jul. 2017. [Online]. Available: http://arxiv.org/abs/1707.07321

[22] P. Li, P. Ren, X. Zhang, Q. Wang, X. Zhu, and L. Wang, "Region-wise deep feature representation for remote sensing images," *Remote Sens.*, vol. 10, no. 6, Jun. 2018, Art. no. 871.

[23] 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.

[24] Z. Shao, K. Yang, and W. Zhou, "Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset," *Remote Sens.*, vol. 10, no. 6, Jun. 2018, Art. no. 964.

[25] W. Zhou, X. Deng, and Z. Shao, "Region convolutional features for multi-label remote sensing image retrieval," Jul. 2018. [Online]. Available: http://arxiv.org/abs/1807.08634
[26] 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, p. 493, Mar. 2019.
[27] F. Ye, M. Dong, W. Luo, X. Chen, and W. Min, "A new re-ranking method based on convolutional neural network and two image-to-class distances for remote sensing image retrieval," *IEEE Access*, vol. 7, pp. 141 498–141 507, Sep. 2019.
[28] R. Cao, Q. Zhang, J. Zhu, Q. Li, Q. Li, B. Liu, and G. Qiu, "Enhancing remote sensing image retrieval with triplet deep metric learning network," *CoRR*, vol. abs/1902.05818, Feb. 2019. [Online]. Available: http://arxiv.org/abs/1902.05818
[29] 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.
[30] Y. Li, Y. Zhang, X. Huang, S. Member, and J. Ma, "Learning source-invariant deep Hashing convolutional neural networks for cross-source remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6521–6536, Nov. 2018.
[31] L. Fan, H. Zhao, and H. Zhao, "Distribution consistency loss for large-scale remote sensing image retrieval," *Remote Sens.*, vol. 12, no. 1, Jan. 2020, Art. no. 175.
[32] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Comput. Vis. Image Und.*, vol. 184, pp. 22–30, Jul. 2019.
[33] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.
[34] 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.
[35] E. Kreyszig, *Introductory Functional Analysis With Applications*, vol. 1. New York, NY, USA: Wiley, Jun. 1978.
[36] X. Tan, S. Chen, Z.-H. Zhou, and J. Liu, "Learning non-metric partial similarity based on maximal margin criterion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2006, vol. 1, pp. 145–168.
[37] N. Garcia and G. Vogiatzis, "Learning non-metric visual similarity for image retrieval," *Image Vis. Comput.*, vol. 82, pp. 18–25, Feb. 2019.
[38] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu, "Classification with non-metric distances: Image retrieval and class representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 583–600, Jun. 2000.
[39] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2015, pp. 44–51.
[40] F. Hu, G. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015.
[41] Y. Liu, C. Y. Suen, Y. Liu, and L. Ding, "Scene classification using hierarchical Wasserstein CNN," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2494–2509, May 2019.
[42] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 183–186, Feb. 2018.
[43] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.
[44] N. Poh and S. Bengio, "How do correlation and variance of base-experts affect fusion in biometric authentication tasks?" *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4384–4396, Nov. 2005.
[45] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, "Likelihood ratio based biometric score fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 342–347, Feb. 2008.
[46] Q. Tao and R. N. J. Veldhuis, "Robust biometric score fusion by naive likelihood ratio via receiver operating characteristics," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 2, pp. 305–313, Feb. 2013.
[47] Y. Liu, L. Yang, and C. Y. Suen, "The effect of correlation and performances of base-experts on score fusion," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 510–517, Apr. 2014.
[48] Y. Liu, Z. Yang, C. Y. Suen, and L. Yang, "A study on performance improvement due to linear fusion in biometric authentication tasks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 9, pp. 1252–1264, Sep. 2016.
[49] Y. Liu, Z. Han, C. Chen, L. Ding, and Y. Liu, "Eagle-eyed multitask CNNs for aerial image retrieval and scene classification," *IEEE Trans. Geosci. Remote Sens.*, Mar. 2020, doi: 10.1109/TGRS.2020.2979011.
[50] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1–9.
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2016, pp. 770–778.
[52] B. G. Tabachnick and L. S. Fidell, *Using Multivariate Statistics*, 5th ed. Boston, MA, USA: Allyn & Bacon/Pearson Education, 2007.
[53] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
[54] 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.*, San Jose, CA, USA, Nov. 2010, pp. 270–279.
[55] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 689–692.
[56] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, 2nd ed. Berlin, Germany: Springer, 2012, vol. 7700, pp. 421–436.
[57] P. Li, X. Zhang, X. Zhu, and P. Ren, "Online hashing for scalable remote sensing image retrieval," *Remote Sens.*, vol. 10, no. 5, May 2018, Art. no. 709.
[58] D. Nister and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2161–2168.
[59] J. Wang, J. Xiao, W. Lin, and C. Luo, "Discriminative and generative vocabulary tree: With application to vein image authentication and recognition," *Image Vis. Comput.*, vol. 34, pp. 51–62, Feb. 2015.
[60] Y. Liu and C. Huang, "Scene classification via triplet networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 220–237, Jan. 2018.
[61] M. Tharani, N. Khurshid, and M. Taj, "Unsupervised deep features for remote sensing image matching via discriminator network," Oct. 2018. [Online]. Available: http://arxiv.org/abs/1810.06470
[62] C. Liu, J. Ma, X. Tang, X. Zhang, and L. Jiao, "Adversarial hash-code learning for remote sensing image retrieval," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Aug. 2019, pp. 4324–4327.
[63] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 761–769.

Yishu Liu (Member, IEEE) received the B.S. degree from Nanjing University, Nanjing, China, in 1996, and the M.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2002 and 2013, respectively, all in computational mathematics.

From 1996 to 1999, she was a Research Assistant with the Institute of Mathematics, Shantou University, Shantou, China. In 2004, she joined as a Lecturer with the School of Geography, South China Normal University, Guangzhou, where she is currently a Professor. Her research interests include machine learning, pattern recognition, remote-sensing image interpretation, remote-sensing image retrieval, and remote-sensing information fusion.

Conghui Chen received the B.S. degree in geographic information system from Fujian Normal University, Fuzhou, China, in 2018. She is currently working toward the M.S. degree with the School of Geography, South China Normal University, Guangzhou, China.

Her research interests include machine learning, remote-sensing image retrieval, and remote-sensing information fusion.

Zhengzhuo Han received the B.S. degree in information management and information system from the Hunan City University, Yiyang, China, in 2019. He is currently working toward the M.S. degree with the School of Geography, South China Normal University, Guangzhou, China.

His research interests include deep learning, remote-sensing image retrieval, and scene classification.

Liwang Ding received the B.S. degree in geographic information system from Nanyang Normal University, Nanyang, China, in 2017. He is currently working toward the M.S. degree with the School of Geography, South China Normal University, Guangzhou, China.

His research interests include deep learning, remote-sensing image retrieval, and aerial scene classification.

Yingbin Liu received the B.S. degree in geographic information system from the Southwest University of Science and Technology, Mianyang, China, in 2017. He is currently working toward the M.S. degree with the School of Geography, South China Normal University, Guangzhou, China.

His research interests include deep learning, remote-sensing image retrieval, and aerial scene classification.