International Journal of Geo-Information

MDPI

Article

Similarity Retention Loss (SRL) Based on Deep Metric Learning for Remote Sensing Image Retrieval

Hongwei Zhao [1,2], Lin Yuan [1,2] and Haoyu Zhao [3,*]

1 College of Computer Science and Technology, Jilin University, Changchun 130012, China; zhaohw@jlu.edu.cn (H.Z.); yuanlin19@mails.jlu.edu.cn (L.Y.)
2 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
3 Editorial Department of Journal (Engineering and Technology Edition), Jilin University, Changchun 130012, China
* Correspondence: zhaohaoyu@jlu.edu.cn; Tel.: +86-1594-809-9990

Received: 25 November 2019; Accepted: 19 January 2020; Published: 21 January 2020

Abstract: Recently, with the rapid growth of the number of datasets with remote sensing images, it is urgent to propose an effective image retrieval method to manage and use such image data. In this paper, we propose a deep metric learning strategy based on Similarity Retention Loss (SRL) for content-based remote sensing image retrieval. We have improved the current metric learning methods from the following aspects—sample mining, network model structure and metric loss function. On the basis of redefining the hard samples and easy samples, we mine the positive and negative samples according to the size and spatial distribution of the dataset classes. At the same time, Similarity Retention Loss is proposed and the ratio of easy samples to hard samples in the class is used to assign dynamic weights to the hard samples selected in the experiment to learn the sample structure characteristics within the class. For negative samples, different weights are set based on the spatial distribution of the surrounding samples to maintain the consistency of similar structures among classes. Finally, we conduct a large number of comprehensive experiments on two remote sensing datasets with the fine-tuning network. The experiment results show that the method used in this paper achieves the state-of-the-art performance.

Keywords: content-based remote sensing image retrieval (CBRSIR); deep metric learning (DML); structural ranking consistency

1. Introduction

Due to the wide use of satellite sensors with short revisit time, various forms of remote sensing images have been accumulated in an unprecedented number. The large amount of generated data that is nowadays available makes it necessary to be able to extract complex information from these images. Image retrieval is a popular information extraction mechanism. Its principle is to retrieve visually consistent images from a predefined database, given a query concept [1,2].

Content-Based Remote Sensing Image Retrieval (CBRSIR) is a specific application of image retrieval on remote sensing image datasets. The working mode of the CBRSIR system can be summarized as two basic processes, namely feature extraction and image matching. The purpose of feature extraction is to find and extract some representative and robust features from the images. The traditional feature extraction methods rely on artificial descriptors (such as SIFT) [3], which is also a widely used remote sensing image representation method in RSIR (Remote Sensing Image Retrieval) work [4,5]. The extraction of artificial features mainly depends on the artificial tags associated with the scene. However, the design of tags requires sufficient professional knowledge and is time-consuming. At the same time, the quality and

availability of the tags directly affect the performance of search engines. Therefore, this feature extraction method has certain defects. On the other hand, some characteristics of the remote sensing images also hinder the direct application of some commonly used image retrieval techniques (such as geometric verification, query expansion, etc.). The remote sensing image contains not only one specific target but also one or more targets and it also has rich geographic information, such as man-made buildings and large-scale natural landscapes, such as trees, farmland, grassland and so forth. Specifically, the remote sensing image covers a relatively large geographical area and can contain different numbers of different semantic objects at the same time, which can be captured by the region at different scales. Although some common remote sensing datasets contain many images that belong to the same semantic category, these images are quite different. For instance, they may differ significantly in appearance or originate from different geographic areas. In addition, the resolution level of remote sensing image and the height of image acquisition will directly affect the size of the target object and some details. In summary, these characteristics have led to certain difficulties and challenges in RSIR.

With the further development of deep learning, CBIR has developed from the simple "artificial descriptor" to the complex "convolutional descriptor" which can be extracted from the Convolutional Neural Networks (CNNS) [6–8]. The deep convolutional neural network can establish the mapping relationship between low-level features and high-level semantics. By extracting highly abstract image information with high-level semantics, the accuracy of RSIR after deep neural network training is better than RSIR based on traditional artificial features [9–11]. In addition, the deep features can be automatically learned from the data without human effort, which makes deep learning techniques have extremely important application value in large-scale RSIR research. Among them, Deep metric learning (DML) is a technology that combines deep learning and metric learning [12]. The purpose of DML is to learn the embedding space, which encourages the embedding vectors between similar samples to be closer, while the dissimilar samples are far away from each other [13–15]. Deep metric learning uses the discriminative ability of CNNS to embed images into metric space, where semantic metrics between measured images can be directly calculated by simple metric algorithms such as Euclidean distance, which makes the implementation process of the algorithm simpler. In addition deep metric learning has been applied in many natural image domains, such as face recognition [12], visual tracking [16,17], natural image retrieval [18], cross-model retrieval [19], geometric multi-manifold embedding [20] and so forth. Although remote sensing images are quite different from ordinary natural images, deep metric learning still has a full development prospect in CBRSIR.

In the DML framework, the loss function plays a key role. With the development of research, a number of loss functions have been proposed. Kaya M et al. [21] combined with recent research results, revealed the importance of deep metric learning and summarized the current problems dealt with in this filed. For instance, the contrastive loss [22,23] captures the similarity or dissimilarity between pairwise of samples, while the triplet-based loss [12,24] describes the relationship among the triple samples. Each triplet consists of an anchor sample, a positive sample and a negative sample. In general, the triplet loss is better than contrastive loss due to the increased relationship between positive and negative sample pairs. Inspired by this, recent researches have considered the richer representation of structured information among multiple samples [25–28] and have achieved good performance in many practical applications (such as image retrieval and image clustering). In particular, Wang et al. [29] proposed a metric learning loss function based on the angular relationship of constrained triples in negative samples, which is called "angular loss". However, the most advanced DML methods still have some limitations. First of all, we notice that when selecting samples for some loss functions, only partial sample information is used and differences and permutations between sample classes are ignored. In this case, not only are some non-trivial samples wasted but the relevant information between the classes is not fully utilized. In Reference [30], researcher used all non-trivial samples with non-zero loss (i.e., violating the pair constraint of query) to construct a structure with more information to learn the embedding vectors, so as to avoid wasting the structural information of some non-trivial samples. Although the information obtained by the method is abundant, some of them are redundant,

which would cause a considerable burden on the calculation cost and data storage. Secondly, the spatial distribution of samples within the class is not considered in the above-mentioned losses but only the similar samples are made as close as possible. Moreover, we observe that the previous losses are equal to each positive sample, that is, they do not consider the impact of the quantitative relationship between simple samples and hard samples on loss optimization. Ideally, a larger weight should be given to a hard sample with a larger percentage. In Reference [31], the authors proposed Distribution Structure Learning Loss (DSLL), which considers that the relative spatial structure of the initial state of negative sample classes is maintained by weighting the negative sample classes. However, it does not consider the influence of the relationship among the positive samples and the interaction between the positive and negative samples on the spatial structure. The above methods would lose some similarity structures and useful sample information within the class.

Based on the above issues, this paper proposes a deep metric learning method based on the Similarity Retention Loss (SRL). This method is improved in the following two aspects. The first is to mine samples based on information pairs and the second is to assign different relative weights to all selected samples. Firstly, we set different thresholds and selection strategies for positive and negative samples to ensure that the selected samples are both representative and non-redundant. At the same time, we recommend that attention should be paid to preserve the structural information within the positive sample class during sample mining. Specifically, we just try to narrow the samples of the same class to within a certain distance threshold, without forcing them to a point. Secondly, we assign dynamic weights to selected hard samples according to the ratio of easy samples to hard samples within the class and weight the loss of ranking consistency based on the distribution of negative sample classes. We build an end-to-end fine-tuning network architecture for remote sensing image retrieval, as shown in Figure 1. Our contributions in this paper are listed as follows:

1. We propose the Similarity Retention Loss (SRL) for deep metric learning, which is completed by two iterative steps, samples mining and pair weights, as shown in Figure 1. The SRL considers the maintenance of similarity structures within and between classes, which makes the model more efficient and more accurate in collecting and measuring information pairs, thus improving the performance of image retrieval.
2. We learn a threshold between similar samples to preserve the distribution of data within the class instead of narrowing down each class to a certain point in the embedding space. The efficient information retention within the class is considered so that the spatial structure features of each class are preserved in the feature space.
3. By using an end-to-end fine-tuning network, we have performed extensive and comprehensive experiments on remote sensing datasets of PatternNet [11] and UCMD (UC Merced Land Use Dataset) [32] to validate the SRL theory. The results show that our method is significantly better than the state-of-the-art technology.

2. Related Work

The fine-tuning network for remote sensing image retrieval consists of samples, network model structure and loss function. These three compositions constitute a complete end-to-end image retrieval system through deep metric learning training. In the following, we will discuss the related work on our main contributions around these three aspects.

2.1. Fine-Tunning Network

The fine-tuning of the network is an alternative method applied directly to a pre-trained network. The method is initialized by a pre-trained classification network and then trained for different tasks. Image feature learning on large-scale datasets (i.e., ImageNet) has strong generalization capabilities and can be effectively migrated to other small-scale datasets [33]. In the process of CNN transfer learning, the output value of the fully connected layer should be considered [7]. However, since the

value of the local feature of the convolutional layer expression image is relatively large [34], we usually use convolutional layer features instead of fully connected layers.

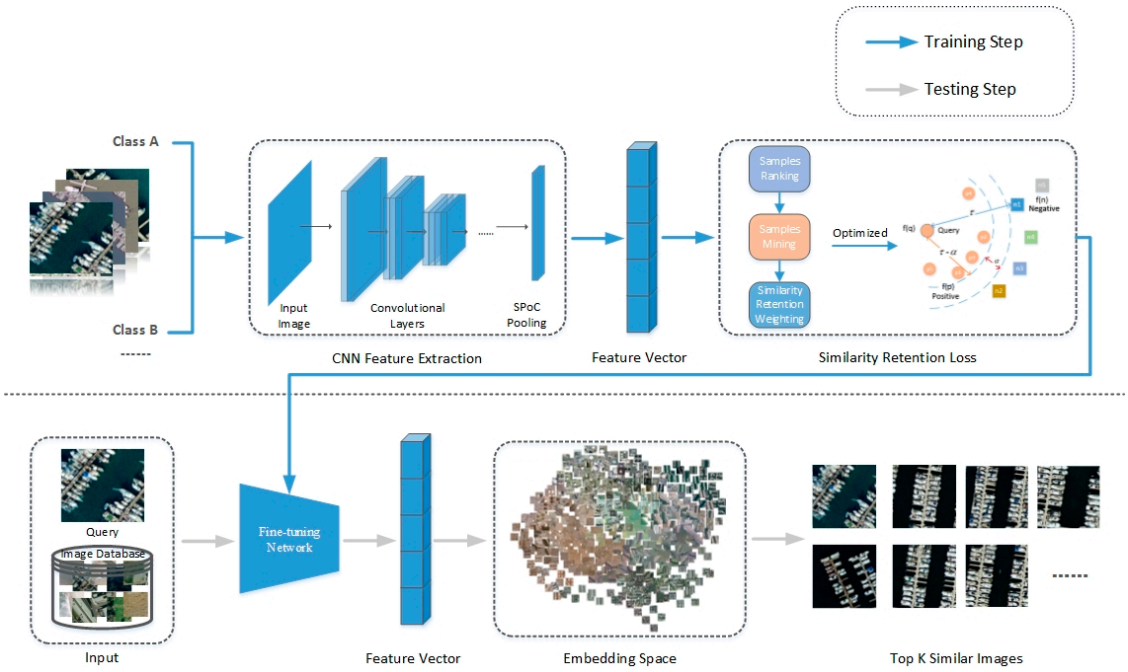


Figure 1. The overall framework of our proposed Similarity Retention Loss algorithm. The top is the training process of the samples in the network and the bottom is the testing process.

Pooling is another major concept in CNNS and is actually a form of down-sampling. Pooling layer imitate the visual input system by reducing the dimension and abstraction of the visual input object. It has the following three functions—feature invariance, feature dimension reduction and avoidance of over-fitting. There are some general pooling models, the most common of which is sum pooling proposed by Babenko and Lempitsky [35] and it performs well in combination with descriptor whitening. Subsequently, Kalantidis et al. proposed weighted sum pooling [36], which can also be seen as a method of transfer learning. The hybrid scheme of linear combination of maximum and sum pooling is the R-Mac [37]. A global hybrid pooling is proposed for image retrieval [38], which is a standard local pooling for object recognition [39].

In this paper, we first use the pre-trained network to fine-tune the network, then select sample pairs from the remote sensing image dataset to train the network and finally optimize our proposed SRL for the final remote sensing image retrieval task. Observing the remote sensing data, we find that the image covers a large geographical area and the area contains rich background information and different numbers of different semantic pairs. We compared several common pooling methods and choose the most appropriate SPoC (Sum-pooled Convolutional Features) pooling layer as the aggregation layer. This convergence layer serves as the last layer of fine-tuning the convolutional neural network to build the system that is best suited for CBRSIR.

2.2. Hard Sample Mining

Sample pair-based metric learning usually use a large number of paired samples but these samples often contain much redundant information. These redundant samples greatly reduce the actual function and convergence speed of the model. Therefore, the sampling strategy plays a particularly critical role in measuring the training speed of the learning model. In contrastive loss, the method of selecting training samples is the simplest, that is, randomly selecting positive and negative sample pairs in the data. Initially, some researches on embedded learning tended to use the simple pairs

training in Siamese network [23,40]. The Siamese network is composed of two computing branches, each of which contains a CNN component. However, this method reduces the convergence speed of the network.

In order to solve this problem, the hard negative mining methods have been proposed and widely used [12,41–43]. Schroff et al. [12]. proposed a hard negative mining scheme by exploring semi-hard triplets. The scheme defines a negative pair father than the positive. However, this negative mining method only generate a small number of valid semi-hard triples and network training usually requires large samples. Harwood et al. [41] proposed a framework called smart mining to collect the samples from the entire dataset. The method will incur high off-line computing costs. Ge et al. [43] proposed the Hierarchical Triplet Loss (HTL), which constructs a hierarchical tree of all categories and collects hard negative pairs through dynamic margin. In Reference [42], the problem of sample mining in deep metric learning was discussed and a distance weighted sample mining was proposed to select pairs of negative samples.

Although all samples within the threshold were mined by the above methods, the differences between the negative sample classes and the influence of surrounding samples on the samples were not considered. In this paper, the diversity and difference of samples are fully considered. Based on this, we select multiple positive samples and negative samples of different classes and set the distance to the samples according to the distribution of negative neighbor samples. We propose a new hard samples mining method, that is, selecting different mining strategies to select positive sample pairs and negative sample pairs by sorting the sample similarity and class information. In this way, the sample selection is both representative and non-redundant, thereby achieving faster convergence and better performance of the model.

2.3. Loss Functions for Deep Metric Learning

The loss function plays a key role in deep metric learning. It is to increase or decrease the distance between samples by adjusting the similarity between samples. In Reference [44], it is recommended to use triplets as training samples to learn the feature space, where the similarity of the positive sample pairs of triples is higher than that of the negative sample pairs. Specifically, the feature space assigns equal weight to the selected sample pairs. In addition, quadruple loss functions have been studied, such as histogram loss [45]. N-pair-mc [23] learns the embedded features by using the structured relationship between multiple samples. The goal is to extract N-1 negative samples from N-1 categories, one negative sample for each category and improve triplet loss by interacting with more negative samples and categories. Concretely, the samples selected in N-pair loss are also assigned the same weight. Movshovitz-Attias et al. proposed Proxy-NCA Loss [42], which uses a proxy instead of the original sample to solve the sampling problem. Static proxy assignment is a proxy for each class and its performance is better than dynamic proxy assignment. However, Proxy-NCA cannot retain the scalability of DML, so the number of classes need to be proposed. Dong et al. proposed a binomial deviance loss [46] and used binomial bias to evaluate the loss between labels and similarity. Binomial deviance cost makes the model mainly train on the hard pairs, that is, the model focuses more on negative samples near the boundary. Unlike the hinge loss, the binomial deviance loss assigns different weights to the sample pairs based on their distance differences. Later, Song et al. proposed Lifted Struct [25], which learns the embedded features by combining all negative samples. The purpose of Lifted Struct is to draw the positive sample pair as close as possible and push all negative samples to a position farther than the margin.

Observing the above loss, triplet loss and N-pair loss give the same weight to the positive and negative sample pairs. Unlike them, binomial Deviance Loss considers self-similarity and Lifted Struct Loss sets weights for positive and negative sample pairs according to negative relative similarity. However, these methods ignore the distribution of samples in the class and the differences between different classes between classes. In this work, we propose the Similarity Retention Loss (SRL). We sort all the samples except the query image according to the learned feature space similarity score with

the query. Then we weight the selected sample pairs according to the feature sorting and label, that is, the degree to which each pair violates the constraint. SRL avoids the limitations of traditional methods by merging a number of hard samples and exploring the inherently structured information. For negative sample pairs, the distance should be as large as possible, so the higher the similarity, the greater the impact and the higher the weight. For positive samples, on the contrary, the lower the similarity, the more attention needs to be paid and the higher the weight. The illustration and comparison of different ranking-motivated losses and our method is presented in Figure 2.

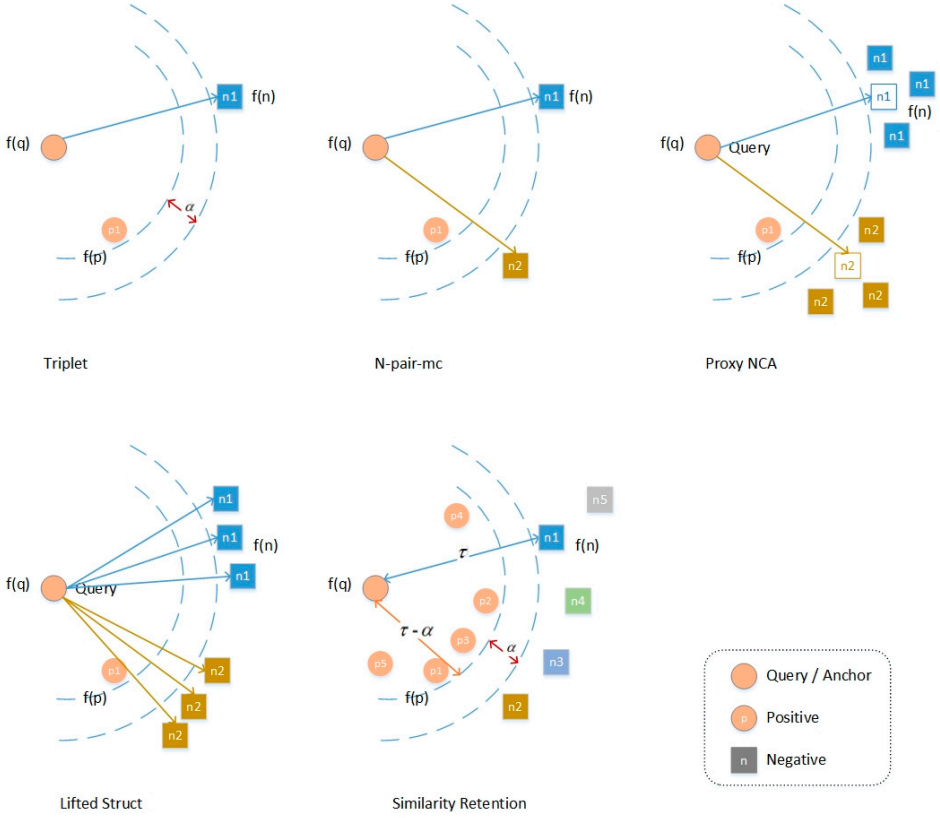


Figure 2. The illustration and comparison of different ranking-motivated losses and our method.

3. The Proposed Approach

Our target is to identify all examples that match this query image from other samples in the dataset, given any query image of any class in the remote sensing dataset. Set $X = \left\{\left(x_i, y_i\right)\right\}_{i=1}^{N}$ as the input data, where $\left(x_i, y_i\right)$ represents the i-th image whose class label is y_i.The number of classes is C, where $y_i \in [1, 2, \ldots, C]$. Let $\left\{X_i^c\right\}_{i=1}^{N_c}$ be the set of images in class c, where the total number of images in class c is N_c.

3.1. Sampling Mining

For the query images, we mine both informative positive and negative samples. Given a query sample X_i^c, we sort all other samples by their similarity to X_i^c. P_i^c is a collection of the same class as the query image which is expressed as $P_i^c = \left\{X_j^c \middle| j \neq i\right\}$, $\left|P_i^c\right| = N_c - 1$. N_i^c is a collection of other images, denoted as $N_i^c = \left\{X_j^k \middle| k \neq c, j \in [1, 2, \ldots, N_k]\right\}$, $\left|N_i^c\right| = \sum_{k \neq c} N_k$. We create a dataset consisting of tuples $(X_i^c, P(X_i^c), N(X_i^c))$, where X_i^c represents the query image, $P\left(X_i^c\right)$ is the positive set that selected from

P_i^c and $N\left(X_i^c\right)$ is the negative set selected from N_i^c. The training image pairs consist of these tuples, where each tuple corresponds to $\left|P\left(X_i^c\right)\right|$ positive sample pairs and $\left|N\left(X_i^c\right)\right|$ negative sample pairs.

Positive sample set $P\left(X_i^c\right)$. Based on the spatial characteristics of the samples, we observe that the positive samples closer to the query not only do not have much useful information to train the network but also increase the cost of samples calculations. Therefore, based on the CNN descriptor distance, we select from P_i^c a fixed number of positive samples that are least similar to the query image as hard positive samples for training iterations. The choice of hard positive samples depends on the current CNN's parameters and is refreshed per epoch.

Negative sample $N\left(X_i^c\right)$. Since the classes are non-overlapping, we select negative samples from classes that are different from the class of the query image. We only select hard negative samples [47,48], that is, mismatched samples with the most similar descriptor to the query image. K-nearest neighbors from all mismatched samples are selected. At the same time, there are multiple similar samples in the same class, which would lead to redundancy of sample information. A fixed number of samples each class is allowed, which provide greater variability in the negative samples. The choice of hard negative samples depends on the parameters of the current CNN and is refreshed multiple times per epoch.

3.2. Loss-Based Sample Weight

Our algorithm aims to bring the positive samples closer to the query image than any negative samples, while pushing the negative samples farther than a predetermined boundary τ. In addition, we try to separate the positive sample boundary from the negative sample boundary by the margin α, that is, the positive samples are within the query sample $\tau - \alpha$ distance. Therefore, α is the margin between the negative and positive samples.

For each query image, the similarity between the selected positive and negative samples and their similarity to the query sample are different. In order to make the most of them, we recommend weighting them according to the loss value of the selected samples, that is, the degree to which each sample pair violates the constraint.

We set a hard positive sample mining threshold between the positive samples and the query according to the spatial distribution features of the samples. Assume that the distance between the sample that is the least similar to the query sample and the query sample is *margin*. The positive samples with a distance from the query image in the range of [0, *threshold*] are defined as easy positive samples with high similarity to the query, while positive samples with a distance in the range of [*threshold*, *margin*] are hard positive samples. The huge impact of hard positive samples in training will weaken the influence of negative samples on gradient changes, which will not only affect the accuracy of the network but also slow down the learning speed. Therefore, in this work, the number of hard positive samples is used to limit the impact of positive samples on loss and to avoid an imbalance in the loss of positive and negative samples during training. The *threshold* is set as $\tau - \alpha$, that is, the feature distance threshold of the positive sample and the query image and we record the number of samples in P_i^c with a distance greater than $\tau - \alpha$ from the query as n_i. Given the selected positive sample X_j^c ($X_j^c \in P\left(X_i^c\right)$), its weight w_{ij}^+ can be calculated as:

$$w_{ij}^+ = \frac{1}{\left|P\left(X_i^c\right)\right|} * \left(1 - \frac{\left|P_i^c\right| - n_i}{\left|P_i^c\right|}\right)^2. \quad (1)$$

For negative sample pairs, we propose a loss weight based on the negative sample order similarity retention. The selection of negative samples is not continuous but is determined by two factors—sample class and similarity with the query. From the perspective of class, the degree of difference between the general characteristics of different negative sample classes and that of the class where the query sample is located is different, so the learning level should also be different. At this time, the fixed margin τ cannot work well. Suppose there are three classes, C, N_1, N_2, where C is the class of the

query image and N_1, N_2 are different negative sample classes. If the difference between the N_1 and C is intuitively smaller than that between N_2 and C, then the distance between N_1 and C should be smaller than that between N_2 and C. However, when the margin value is fixed as set before, if the setting is larger, the model may not be able to distinguish between N_1 and C well. On the contrary, if the margin is set smaller, N_2 and C may not be distinguished well. At the same time, the similarity between the negative samples and the query image is also different, so the impact on the training itself and the required computational cost are also different. We assign different weights to each negative sample class to maintain their relative similarity to the query sample, while ensuring that the characteristics of each class are retained. Specifically, given a selected negative sample X_j^k ($X_j^k \in N(X_i^c)$), its weight w_{ij}^- can be calculated as:

$$w_{ij}^- = 1 - \left(\frac{\left| N(X_i^c) \right| - r_j}{\left| N(X_i^c) \right|} \right)^2, \tag{2}$$

where r_j is the sort position of the negative sample X_j^k in the negative sample list $N(X_i^c)$.

3.3. Similarity Retention Loss

For each query X_i^c, we aim to make it father from the negative sample N_i^c than it is from the positive samples P_i^c, with a minimum difference of α. Therefore, we pull samples from the same class into the margin $\tau - \alpha$. We train the dataset on a two-branch network with the Siamese architecture. Each branch is a clone of another branch, which means that they have the same hyper-parameters.

In order to bring together all positive samples in P_i^c, we minimize:

$$L_p(X_i^c; f) = \sum\nolimits_{X_j^c \in P(X_i^c)} w_{ij}^+ \left(\left[f(X_i^c) - f(X_j^c) - (\tau - \alpha) \right]_+ \right)^2, j \in \left[1, 2, \ldots, \left| P(X_i^c) \right| \right]. \tag{3}$$

Similarly, to push negative samples in N_i^c away from the boundary τ, we minimise:

$$L_N(X_i^c; f) = \sum\nolimits_{X_j^k \in N(X_i^c)} \left(\left[w_{ij}^- * \tau - f(X_i^c) - f(X_j^k) \right]_+ \right)^2, j \in \left[1, 2, \ldots, \left| N(X_i^c) \right| \right], \tag{4}$$

where f is a discriminative function we learned, so that the similarity between the query and the positive samples in the feature space is higher than the similarity between the query and the negative samples.

In SRL, we treat the two minimized objectives equally and optimize them jointly:

$$L_{SRL}(X_i^c; f) = \frac{1}{2}(L_p(X_i^c; f) + L_N(X_i^c; f)). \tag{5}$$

In order to reduce the amount of calculation and calculation time, we randomly select I ($I<N_c$) pictures from each class c as the query image set $Q = \left\{ \left\{ X_q^c \right\}_{q=1}^{I} \right\}_{c=1}^{C}$ and other images act as the library (the selected query picture is also the library of other query pictures). The SRL of the network is represented as:

$$L_{SRL}(X; f) = \frac{1}{I * C} \sum\nolimits_{\forall c, \forall q} L_{SRL}(X_q^c; f). \tag{6}$$

3.4. Learning Fine-Tuning Network Based on SRL

We implement our SRL based on a two-branch network with the Siamese architecture. Each branch is a clone of another branch, which means that they have the same hyper-parameters. The learning of the deep embedding function based on SRL is illustrated in Algorithm 1. Network training and testing process is shown in Figure 1.

Algorithm 1 Similarity Retention Loss on Fine-tuning Network

Parameters Setting: The distance constraint τ on negative examples, the margin between positive and negative examples α, the number of classes C, the number of images per class $N_c (c \in C)$, the total number of images $N = \sum_i^C N_i$, the number of query of per class I.

Input: the discriminative function f, the learning rate lr,

$X = \left\{\left(x_i, y_i\right)\right\}_{i=1}^{N} = \left\{\left\{\left(X_i^c\right)\right\}_{i=1}^{N_c}\right\}_{c=1}^{C}$,the query list $Q = \left\{\left\{X_q^c\right\}_{q=1}^{I}\right\}_{c=1}^{C}$

Output: Updated f.

Step 1: Forward all images into f to obtain the images' embedding feature vector.

Step 2: Online iterative ranking and loss computation.

  for each query X_q^c do

  Rank other images according to the similarity with the X_q^c

  Mine positive samples $P\left(X_q^c\right)$.

  Mine negative samples $N\left(X_q^c\right)$.

  Weigh positive samples using Equation (1).

  Weigh negative samples using Equation (2).

  Compute $L_P\left(X_q^c; f\right)$ using Equation (3).

  Compute $L_N\left(X_q^c; f\right)$ using Equation (4).

  Compute $L_{SRL}\left(X_q^c; f\right)$ using Equation (5).

  end for

  Compute $L_{SRL}(X; f)$ using Equation (6).

Step 3: Gradient computation and back propagation to update the parameters of f.

  $\nabla f = \partial L_{SRL}(X; f) / \partial f$

  $f = f - lr * \nabla f$

4. Experiments

4.1. Datasets

This paper uses two published RSIR datasets, PatternNet [11] and UCMD [32], to evaluate our proposed Similarity Retention Loss (SRL) for deep metric learning. The PatternNet [11] is a large-scale remote sensing dataset with high-resolution collected for RSIR. It includes 38 classes, each of which has 800 images of 256 × 256 pixel size. This dataset is images of US cities collected through Google Map API or Google Earth imagery. PatternNet contains images with different resolutions. The maximum spatial resolution is about 0.062m and the minimum spatial resolution is about 4.693m. The representative image of each class of the PatternNet dataset are shown in Figure 3, visually. The UCMD [32] is a land-cover or land-use dataset used as the RSIR benchmark dataset. It contains 21 classes with 100 images of 256 × 256 pixels per class. These images are segmented from large aerial images downloaded by the USGS (United States Geological Survey), with a spatial resolution of approximately 0.3m. UCMD is a highly challenging dataset with some high overlapping categories such as the sparse, medium and dense residential. A representative image of each class of the UCMD dataset are shown in Figure 4, visually.

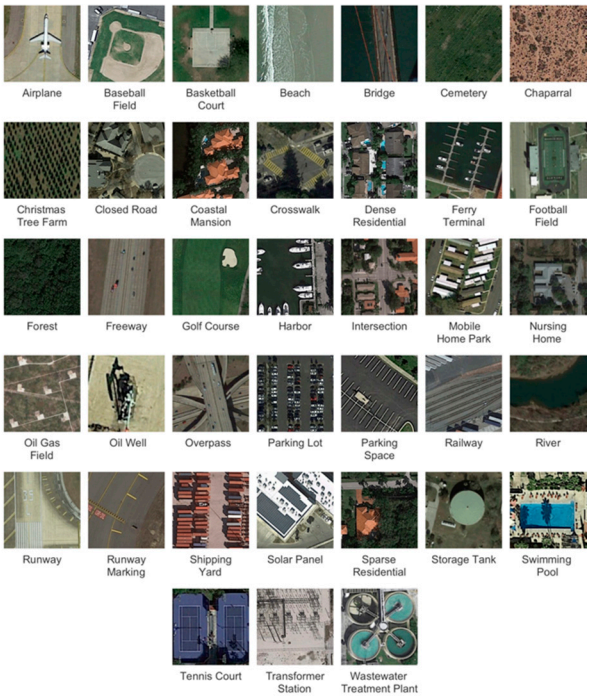


Figure 3. Illustration of the PatternNet. The PatternNet database covers 38 land-cover classes and one image of each class randomly selected from the PatternNet are shown.

Figure 4. Illustration of the UCMD (UC Merced Land Use Dataset). The UCMD database covers 21 land-cover classes and one image of each class randomly selected from the UCMD are shown.

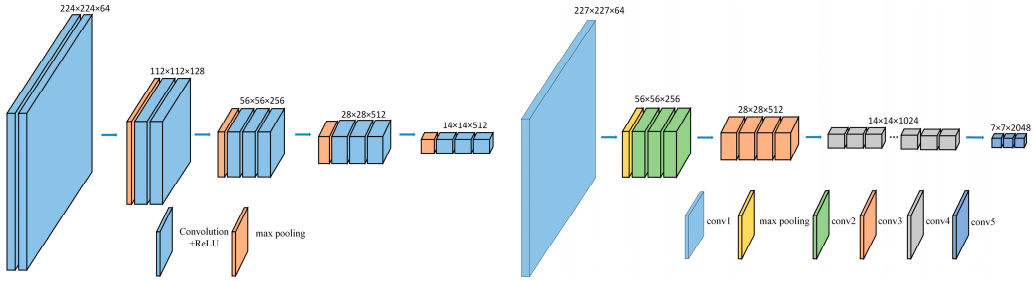
4.2. Performance Evaluation Metrics

In this experiment, we measure the similarity with the Euclidean distance and use the mean Average Precision (mAP), precision of the top-k (P@k) and recall of the top-k (R@k) to evaluate image retrieval performance.

4.3. Training Setup

For UCMD, we adopt the data segmentation strategy that produces the best performance in Reference [10], that is, randomly select 50% samples of each category for training and the remaining 50% for testing. For PatternNet, we use 80%/20% training and testing data segmentation strategy from [11].

Figure 5 presents the two CNNs used by our network (shown in Figure 1), which are used as the basic networks for feature extraction, namely VGG16 [49] and ResNet50 [50]. We use MatConvNet [51] to fine-tune the network. For the CNNs, only the convolutional layers are used to extract features. We remove the last pooling layer of the CNN networks and use the other convolutional layers as our basic CNN structure and then connect the SPoC pooling and L2 regularization to the new network structure. In this experiment, the network is implemented based on the PyTorch framework. Initialize the parameters of each network using the corresponding network weights pre-trained on the ImageNet. We train the network with the Adam optimizer, with weight decay 5×10^{-4}, momentum 0.9, proved by the increase of embedded dimension and the training tuple of batch size 5.



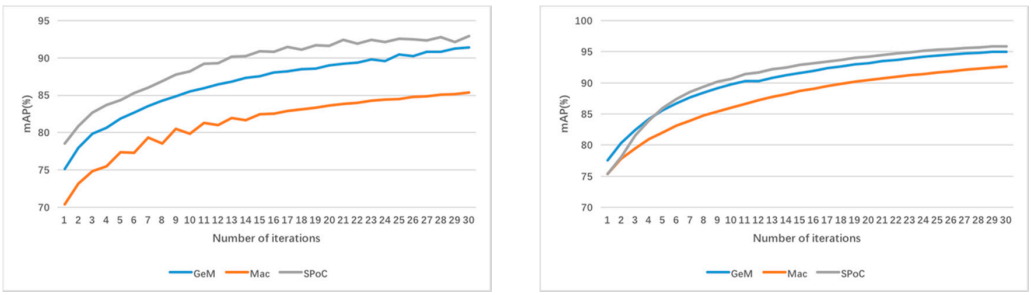
(**a**) (**b**)

Figure 5. Convolutional Neural Network (CNN) network structure: (**a**) VGG16; (**b**) ResNet50.

4.4. Result and Analysis

4.4.1. Pooling Methods

In this section we compare the most advanced pooling methods—max pooling (MAC) [52], average pooling (SPoC) [35] and Generalized Mean pooling (GeM) [33]. We use the SRL loss for network training on the datasets with the learning rate 5e-8. Instead of fine-tuning the pooling layer of the last layer of the convolutional neural network, the above three pooling methods are used. It can be concluded from Figure 6 that SPoC is superior to MAC and GeM on all datasets. In general, there are two main aspects to the error of feature extraction. The first is an increase in the variance of the estimates due to the finite size of the neighborhood. The second reason is that the error of the convolutional layer parameters leads to the offset of the estimated mean. The SPoC pooling can retain more image background information by calculating the average value of the image area, so as to reduce the occurrence of the first type of error. This feature satisfies the large geographic area of the remote sensing images dataset, has rich background information and contains different numbers of different semantic pairs, which makes the effect of SPoC better than other pooling methods in remote sensing image retrieval.



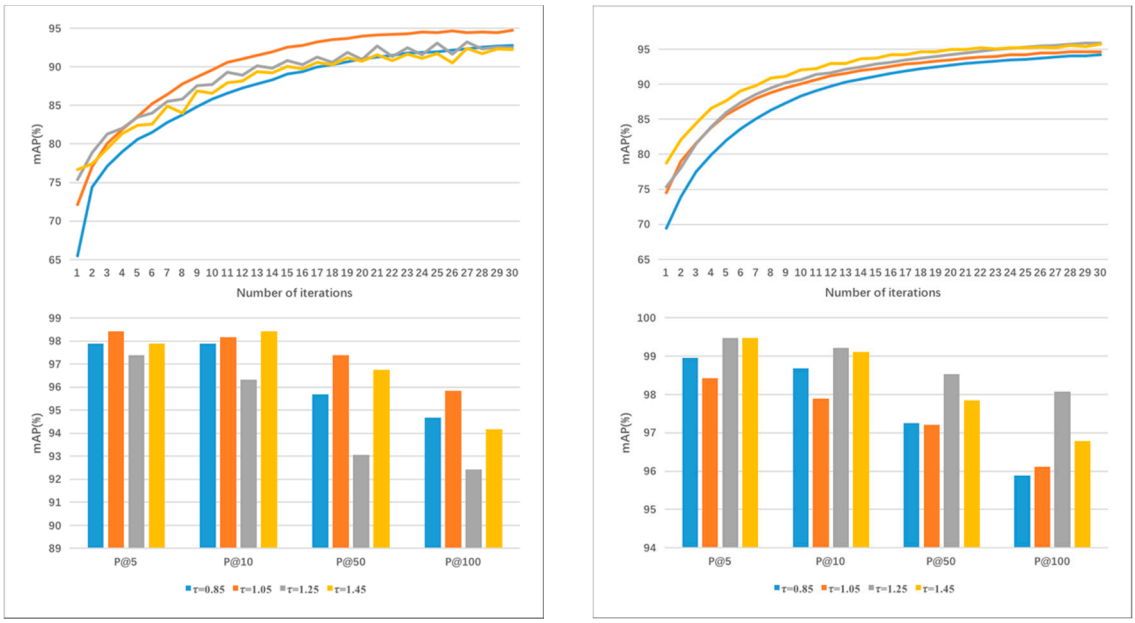
(**a**) (**b**)

Figure 6. The pooling methods. Evaluation is performed with VGG16 (**a**) and ResNet50 (**b**) on PatternNet datasets. The curve represents the evolution of mAP in the training iteration.

4.4.2. Impact of the Negative Margin

As shown in the Section 3.2, for each query sample, SRL ensures the consistency of the structural similarity order of the negative samples by adjusting the size of the negative sample space structure. Since the constraint parameter τ determines the size of the negative space, we performed experiments on the dataset to analyze the impact of the parameter τ.

In order to adapt the threshold τ to the PatternNet dataset and improve the performance of different networks, the experiment selects a value of 0.5–1.5 and trains the network with a learning rate of 0.00001. Finally, the results of τ = 0.85, 1.05, 1.25, 1.45 were selected according to the experiment, as shown in Figure 7. The chart shown in Figure 7a is trained under VGG16, while Figure 7b represents the dataset obtained under ResNet50 training. The results show that the optimal parameter τ for VGG16 network is 1.05, while it is 1.25 for ResNet50. As can be seen from the graph, the performance of the network increases with the increase of the threshold τ but when τ increases to a certain threshold, the value decreases. This is because when the threshold value is small, the distance between the query and negative samples is not enough to distinguish them. As the threshold τ increases, negative samples with high similarity will decrease, which will affect the training effect. The results show that when the thresholds are 1.05 (VGG6) and 1.25 (ResNet50), the difference between the positive and negative samples is the best and the model results are the best. In the next experiment, we chose the threshold τ = 1.05 for the VGG16 network and 1.25 for the ResNet50.



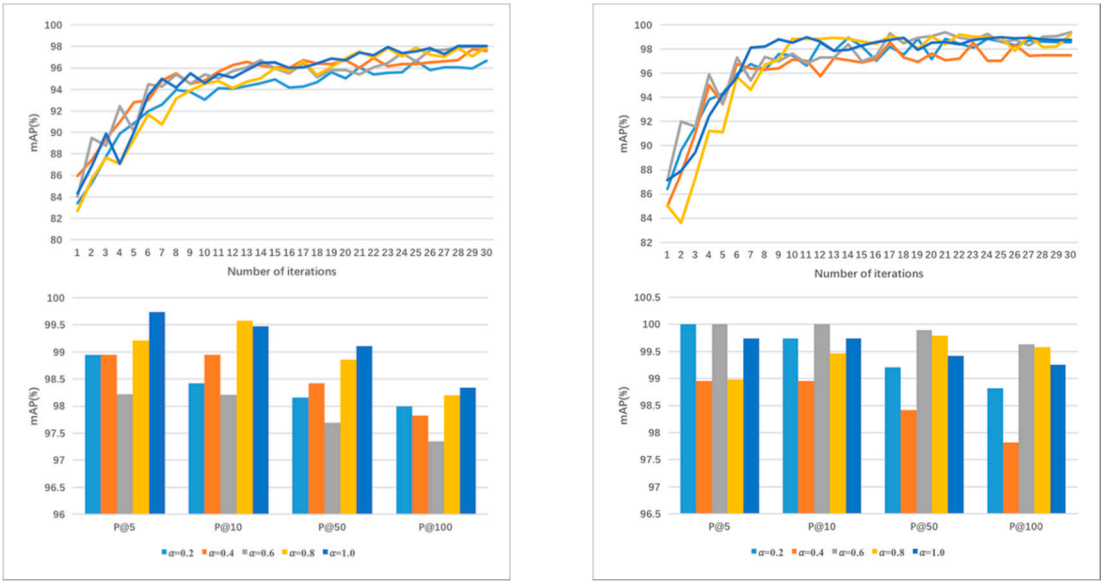
(**a**) (**b**)

Figure 7. The impact of the different threshold τ. Evaluation is performed with VGG16 (**a**) and ResNet50 (**b**) on PatternNet datasets. The curve represents the evolution of mAP in the training iteration. The histogram shows the evaluation of P@K under different thresholds τ.

4.4.3. Impact of the Parameter α

The threshold τ is used to control the distance that the negative samples are pushed away, while the threshold α is used to control the degree of aggregation of the positive samples, that is, the distance between the positive and negative samples. By setting the threshold α, the distance between the positive and negative samples can be pulled while maintaining the spatial structure among the positive samples. As described in 4.4.2, in VGG16, we performed an experiment of threshold α under the condition of τ = 1.05 and in ResNet50, we set τ = 1.25.

In the experiment, the values of the threshold α are 0.2, 0.4, 0.6, 0.8 and 1.0, respectively. The experimental results are shown in Figure 8. The results show that when we set $\alpha = 1.0$, the best result is obtained in VGG16 (a). And in ResNet50 (b), the best result is obtained at $\alpha = 0.6$. That's because when the α is small, the distance between the positive and negative samples is not large enough, so that the network after training cannot clearly distinguish them. Conversely, when α is too large, the spatial structure inside the positive sample cannot be maintained. Therefore, the network can achieve the best effect only when the value of α can distinguish the positive and negative samples and maintain the positive sample space structure.



(**a**) (**b**)

Figure 8. The impact of the different threshold α. Evaluation is performed with VGG16 (**a**) and ResNet50 (**b**) on PatternNet datasets. The curve represents the evolution of mAP in the training iteration. The histogram shows the evaluation of P@K under different thresholds α.

4.4.4. Ceteris Paribus Analysis

In this section, we study in more benefits of using the method Similarity Retention Loss over other structural losses. For this purpose, we replace the proposed SRL in our approach with the Triplet Loss [44], N-pair-mc Loss [23], Proxy-NCA Loss [42], Lifted Struct Loss [25] and Distribution Structure Learning Loss (DSLL) [31]. We then re-train the network, keeping the network structure (ResNet50) identical and separately re-tuning some hyper-parameters, such as the weight decay and the learning rate. In the experiment, we use the mean Average Precision (mAP), precision of the top-k (P@k) and recall of the top-k (R@k) to evaluate image retrieval performance. The UCMD dataset used in the experiment contains 21 classes of 100 images per class. We randomly select 50% of each class for training and the remaining 50% for testing (i.e., 50 images of each class). According to the quantitative characteristic of UCMD dataset, we choose Recall at top 25, 40, 50, 100 as one of the evaluation criteria for the test result. We randomly select 80% of each class of images from the PatternNet dataset (containing 38 classes, 800 images per class) as the training set and the remaining 20% as the test set (i.e., 160 images of each class are used as the test set). So we select Recall at top 80, 100, 160, 200 as the evaluation criteria for the test result of the PatternNet dataset. We evaluate the proposed algorithm on image retrieval tasks in comparison with the advanced metric learning loss algorithms. Performance after training is presented in the Tables 1 and 2. As can be seen from the table, the accuracy of our method is higher than others. When using the ResNet50 network framework, compared with the DSLL, SRL provides a significant improvement of +1.26% in mAP and +1.12% in

R@50 on UCMD dataset. Furthermore, the SRL signatures achieves a gain of +1.07% in mAP and +0.98% in R@160 on PATTERNNET dataset, which surpassed recently published DSLL and achieves mAP of 99.41%, P@10 of 100 and R@180 of 99.96%. In general, our approach is demonstrated to be the most effective. This is because we use a new method of mining samples through spatial distribution and the loss of similarity retention calculation for all selected samples.

Table 1. The evaluation results of mAP and P@K on the PatternNet and UCMD database comparing with the other structure loss.

Dataset	Structural Loss	mAP	P@5	P@10	P@50	P@100	P@1000
UCMD	Triplet Loss	92.94	98.52	96.92	92.13	46.07	4.61
	N-pair-mc Loss	91.11	94.94	91.15	90.33	45.17	4.52
	Proxy-NCA Loss	95.71	97.56	96.69	94.89	47.45	4.74
	Lifted Struct Loss	96.58	98.05	97.62	95.75	47.88	4.79
	DSLL	97.52	98.09	98.03	96.68	48.34	4.83
	SRL	98.78	99.63	99.56	99.33	48.96	4.90
PatternNet	Triplet Loss	94.96	99.04	97.63	96.62	95.16	15.69
	N-pair-mc Loss	94.81	97.04	95.46	94.49	95.08	15.67
	Proxy-NCA Loss	97.72	98.98	98.65	98.23	98.02	15.71
	Lifted Struct Loss	98.09	98.90	98.82	98.78	98.46	15.76
	DSLL	98.34	99.05	98.98	98.93	98.67	15.86
	SRL	99.41	100	100	99.55	99.24	15.90

Table 2. The evaluation results of R@K on the PatternNet and UCMD database comparing with the other structure loss methods.

Dataset	Structural Loss	R@25	R@40	R@50	R@100
UCMD	Triplet Loss	47.75	76.99	91.23	96.21
	N-pair-mc Loss	45.39	75.57	90.19	95.65
	Proxy-NCA Loss	48.56	77.47	96.92	99.14
	Lifted Struct Loss	49.04	77.11	97.13	99.26
	DSLL	49.63	78.06	97.31	99.28
	Similarity Retention Loss	49.71	78.48	98.43	99.95
Dataset	**Structural Loss**	**R@100**	**R@130**	**R@160**	**R@180**
PatternNet	Triplet Loss	48.85	77.52	96.32	98.61
	N-pair-mc Loss	48.80	77.38	95.97	98.36
	Proxy-NCA Loss	48.97	78.60	97.31	99.17
	Lifted Struct Loss	49.01	78.64	97.51	99.28
	DSLL	49.16	79.03	98.30	99.33
	Similarity Retention Loss	49.96	79.78	99.28	99.96

4.4.5. Overall Results and Per-Class Results

We present experiments on the PatternNet and UCMD datasets, with margin $\tau = 1.05$ for VGG16 and 1.25 for ResNet50. In this experiments we set margin $\alpha = 1.0$ for VGG16 and 0.6 for ResNet50. The final results of the PatternNet and UCMD datasets are shown in Table 3. It can be seen that, compared to the state-of-the-art performance, the SRL-based features can achieve optimal performance. When using the VGG16 network framework, compared with the MiLaN, SRL provides a significant improvement of +7.38% in mAP on UCMD dataset. Furthermore, the SRL achieves a gain of +24.92% in mAP and +3.67% in P@10 on PATTERNNET dataset, which surpassed recently published GCN (Graph Convolutional Networks). When using the ResNet50 network framework, on the UCMD dataset, the experimental results achieve +8.38% growth compared to MiLaN in mAP and achieve mAP of 99.41%, P@10 of 100 and offer over 73.11%, 95.53% gain over the GCN on PATTERNNET dataset. At the same time, we find that although the effect of the EDML (Enhancing Remote Sensing Image Retrieval with Triplet Deep Metric Learning Network) [53] on the PatternNet dataset is slightly

higher than our SRL, for example, the EDML achieves a gain of +1.40% and +0.14% in mAP on PatternNet database, which trained respectively on the VGG16 network and ResNet50. But based on comprehensive experimental results, our SRL is the best. First, from the results (Table 3), our method can effectively improve the accuracy of the network on the UCMD dataset (the number of images in the dataset is smaller). Specific example—the SRL method gains +2.91% and +2.15% on the mAP obtained after training on the VGG16 network and the ResNet50 network, respectively, which exceeds the result of EDML. This shows that our method is more friendly to the dataset with insufficient images, which is very meaningful in image retrieval. Second, we find that the sample mining strategy adopted by the EDML is to randomly pick positive samples from the same class as the anchor (except the anchor) and negative samples from any other classes. This strategy has some disadvantages. (1) The representativeness of the samples is difficult to guarantee; (2) The work of obtaining samples is heavy; (3) It makes the training convergence time longer. In order to verify the advantages of the proposed SRL algorithm model in terms of training speed, we reproduce the EDML and compare the training time of the model with our model. We conduct experiments on Intel® i7-8700, 11 GB memory CPU, Ubuntu 18.04LTS operating system and use VGG16 and ResNet50 as the basic network to calculate the training time. The results show that the training time of 70 epochs of UCMD database using EDML algorithm in VGG16 and ResNet50 network is 9.8 h and 27.8 h respectively, while the training time of PatternNet dataset is 11.6 h and 30.9 h respectively. Training with SRL took 8.2 h (VGG16, UCMD), 24.4 h (ResNet50, UCMD), 9.9 h (VGG16, PatternNet) and 27.6 h (ResNet50, PatternNet). In general, our approach is demonstrated to be the most effective. To summarize, on both remote sensing datasets like UCMD dataset and PatternNet dataset, our method achieves new state-of-the-art or comparable performance.

Interestingly, the best performance on PatternNet is significantly better than the UCMD. One possible reason is that data-driven is a major feature of deep metric learning and the learning performance of representative features is affected by the amount of training data. PatternNet has a larger amount of data than UCMD, so the network for the former is better trained than the latter. The image retrieval visualized results of PatternNet and UCMD trained under the ResNet50 network are shown in Figure 9.

Table 3. Evaluation results on the PatternNet and UCMD database comparing with the state-of-the-art methods.

Dataset	Feature	mAP	P@5	P@10	P@50	P@100	P@1000
PatternNet	Gabor Texture [11]	27.73	68.55	62.78	44.61	35.52	8.99
	VLAD [11]	34.10	58.25	55.70	47.57	41.11	11.04
	UFL [11]	25.35	52.09	48.82	38.11	31.92	9.79
	VGGF Fc1 [11]	61.95	92.46	90.37	79.26	69.05	14.25
	VGGF Fc2 [11]	63.37	91.52	89.64	79.99	70.47	14.52
	VGGS Fc1 [11]	63.28	92.74	90.70	80.03	70.13	14.36
	VGGS Fc2 [11]	63.74	91.92	90.09	80.31	70.73	14.55
	ResNet50 [11]	68.23	94.13	92.41	83.71	74.93	14.64
	LDCNN [11]	69.17	66.81	66.11	67.47	68.80	14.08
	G-KNN [54]	12.35	-	13.24	-	-	-
	RAN-KNN [54]	22.56	-	37.70	-	-	-
	VGG-VD16 [54]	59.86	-	92.04	-	-	-
	VGG-VD19 [54]	57.89	-	91.13	-	-	-
	GoogLeNet [54]	63.11	-	93.31	-	-	-
	GCN [54]	73.11	-	95.53	-	-	-
	SGCN [54]	71.79	-	97.14	-	-	-
	EDML (VGG16) [53]	99.43	99.53	99.50	99.47	99.46	15.90
	EDML (ResNet50) [53]	99.55	99.58	99.57	99.57	99.54	15.90
	SRL (VGG16)	98.03	99.86	99.20	98.41	98.26	15.90
	SRL (ResNet50)	99.41	100	100	99.55	99.24	15.90

Table 3. *Cont.*

Dataset	Feature	mAP	P@5	P@10	P@50	P@100	P@1000
UCMD	KSLSH [55]	63.0	-	-	-	-	-
	G-KNN [54]	7.5	-	10.12	-	-	-
	RAN-KNN [54]	26.74	-	24.90	-	-	-
	VGG-VD16 [54]	53.71	-	78.34	-	-	-
	VGG-VD19 [54]	53.19	-	77.60	-	-	-
	GoogLeNet [54]	53.13	-	80.96	-	-	-
	GCN [54]	64.81	-	87.12	-	-	-
	SGCN [54]	69.89	-	93.63	-	-	-
	MiLaN [54]	90.4					
	EDML (VGG16) [53]	94.87	97.41	96.87	90.57	48.28	4.90
	EDML (ResNet50) [53]	96.63	97.75	97.57	93.20	48.55	4.90
	SRL (VGG16)	97.78	98.97	98.14	96.78	48.74	4.90
	SRL (ResNet50)	98.78	99.63	99.56	99.33	48.96	4.90

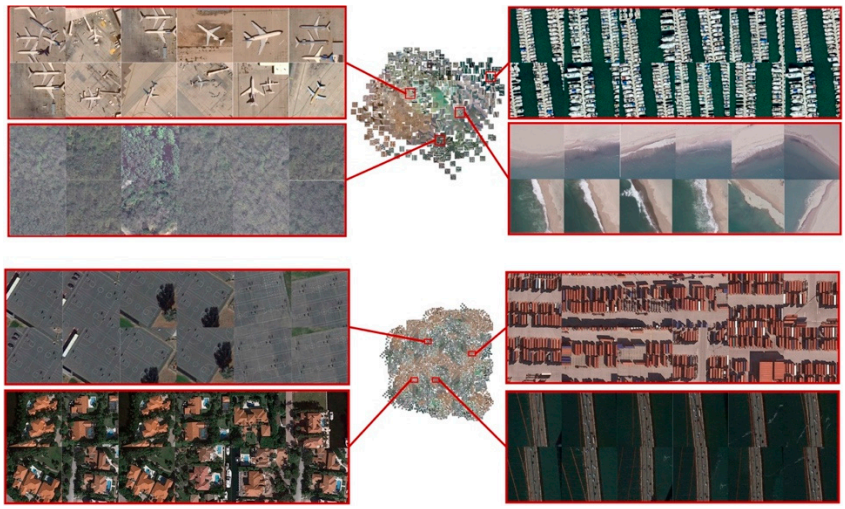


Figure 9. The image retrieval visualized results of PatternNet (bottom) and UCMD (top) trained under the ResNet50.

For per-class results, specific results of mAP evaluation which is performed with VGG16 and ResNet50 in the PatternNet and UCMD datasets comparing with the pre-trained CNNs are shown in Tables 4 and 5. It can be seen that the results of each class of training based on SRL are improved compared to pre-training. The performances of mAP based on the different deep features of VGG16 (top) and ResNet50 (bottom) are visually shown in Figures 10 and 11, respectively. Overall, for every class, the SRL-based features are superior to the pre-trained features on both datasets. As presented in Table 4, in general, for almost every class, SRL-based features outperform pre-trained features. Pre-trained VGG16-based features have particular difficulty in retrieving images of building, intersection and sparser residential, with an average mAP of 25.68%, much lower than that of its counterpart, with 87.4% for the SRL-based features on the UCMD dataset. Meanwhile, on the PatternNet dataset, Pre-trained VGG-based features have particular difficulty in dense residential, intersection and parking lot, with an average mAP of 29.35%, reach up to 93.8% for SRL-based features. Pre-trained ResNet50-based features perform poorly on classes like bridge, nursing home intersection and runway, with an average mAP of 26%. This value for SRL-based feature is 98.38% on the UCMD dataset. While on the PatternNet dataset, pre-trained features are not performance well in bridge, tennis court and ferry terminal, with an average mAP of 28.27%, while 95.11% for SRL-based features, which further demonstrates the superior performance of SRL-based features for CBRSIR. As can be seen from Figures 10 and 11 that SRL-based features outperform much better than pre-trained features for all the classes. At the same

time, the results on the PatternNet are better than UCMD in both networks and ResNet50 outperformed VGG16 for both datasets.

Table 4. Evaluation of mAP which is performed with VGG16 and ResNet50 on per-class in the PatternNet datasets comparing with the pre-trained CNNs.

	VGG16		ResNet50	
	Pre-trained	**SRL-based**	**Pre-trained**	**SRL-based**
Airplane	95.23	100	92.99	100
Baseball Field	97.01	99.91	96.82	100
Basketball Court	50.67	97.24	45.32	98.63
Beach	100	100	99.92	100
Bridge	24.34	98.97	13.50	99.43
Cemetery	93.74	100	93.87	100
Chaparral	99.94	100	100	100
Christmas Tree Farm	98.23	100	83.88	100
Closed Road	93.16	99.99	91.26	99.99
Coastal Mansion	99.65	97.21	98.02	99.90
Crosswalk	96.63	100	93.57	100
Dense Residential	52.99	82.00	46.84	99.70
Ferry Terminal	58.19	83.67	40.14	87.97
Football Field	97.61	99.99	89.17	100
Forest	99.84	100	100	100
Freeway	99.82	100	99.62	100
Golf Course	95.18	99.53	95.13	99.93
Harbor	89.84	96.23	92.12	96.76
Intersection	52.38	98.75	51.39	99.93
Mobile Home Park	86.20	99.55	81.47	100
Nursing Home	23.87	96.72	59.68	98.15
Oil Gas Field	99.99	100	99.99	100
Oil Well	100	100	100	100
Overpass	77.56	99.82	90.00	99.98
Parking Lot	99.96	99.99	98.30	100
Parking Space	52.53	100	47.60	100
Railway	83.15	99.63	78.05	100
River	99.75	100	99.82	100
Runway	29.86	99.46	36.26	99.98
Runway Marking	99.34	99.99	99.88	100
Shipping Yard	97.11	99.76	99.91	99.99
Solar Panel	99.01	99.43	99.57	100
Sparse Residential	64.32	91.98	47.74	99.75
Storage Tank	42.85	99.68	55.23	99.57
Swimming Pool	18.29	96.15	43.95	99.13
Tennis Court	59.74	91.65	31.18	97.92
Transformer Station	69.75	99.97	63.97	99.97
Wastewater Treatment Plant	91.73	98.49	90.99	99.92
Average	78.66	98.03	77.56	99.41

Table 5. Evaluation of mAP which is performed with VGG16 and ResNet50 on per-class in the UCMD datasets comparing with the pre-trained CNNs.

	VGG16		ResNet50	
	Pre-trained	**SRL-based**	**Pre-trained**	**SRL-based**
Agriculture	94.48	99.8	99.74	100
Airplane	66.49	100	99.73	99.98
Baseball Diamond	60.82	99.90	59.27	99.96
Beach	99.25	100	99.03	100
Buildings	33.53	74.21	37.12	99.07
Chaparral	99.80	100	100	100
Dense Residential	36.83	94.47	24.49	97.63
Forest	88.30	100	99.82	100
Freeway	55.65	99.16	87.55	99.57
Golf Course	42.08	99.60	83.02	99.77
Harbor	59.00	100	68.00	100
Intersection	31.76	98.37	31.26	98.81
Medium Residential	48.77	93.24	61.19	99.00
Mobile Home Park	58.78	100	72.27	99.94
Overpass	37.55	97.52	51.57	99.50
Parking Lot	79.00	100	32.30	82.80
River	67.59	98.96	60.50	99.17
Runway	57.05	100	89.27	99.98
Sparse Residential	11.76	89.64	55.88	99.18
Storage Tanks	77.72	99.17	88.40	99.49
Tennis Court	39.01	99.99	78.47	100
Average	59.32	97.78	70.35	98.77

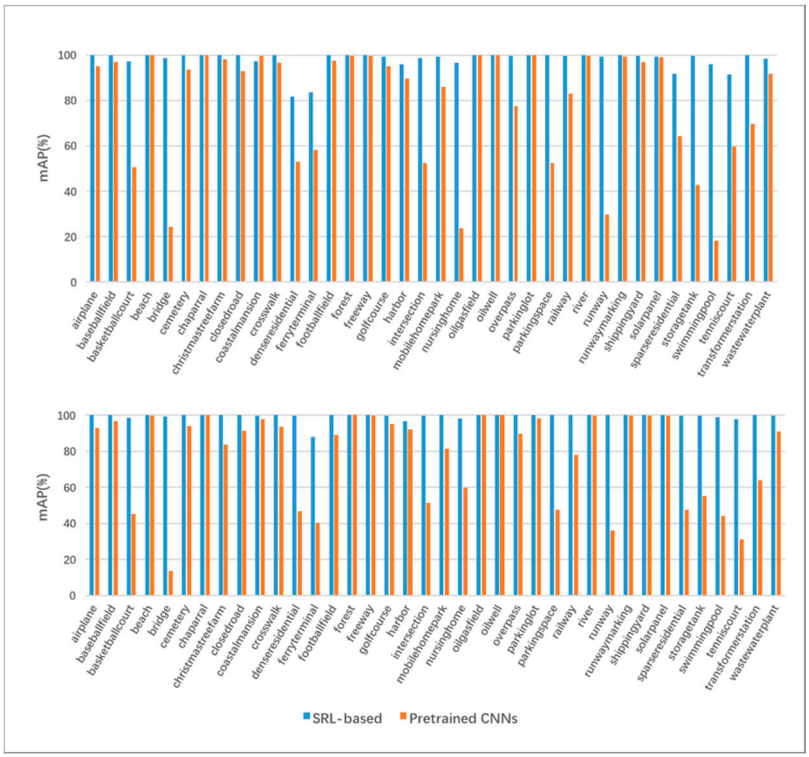


Figure 10. Evaluation of mAP is performed with VGG16 (top) and ResNet50 (bottom) on per-class in the PatternNet datasets trained with different features.

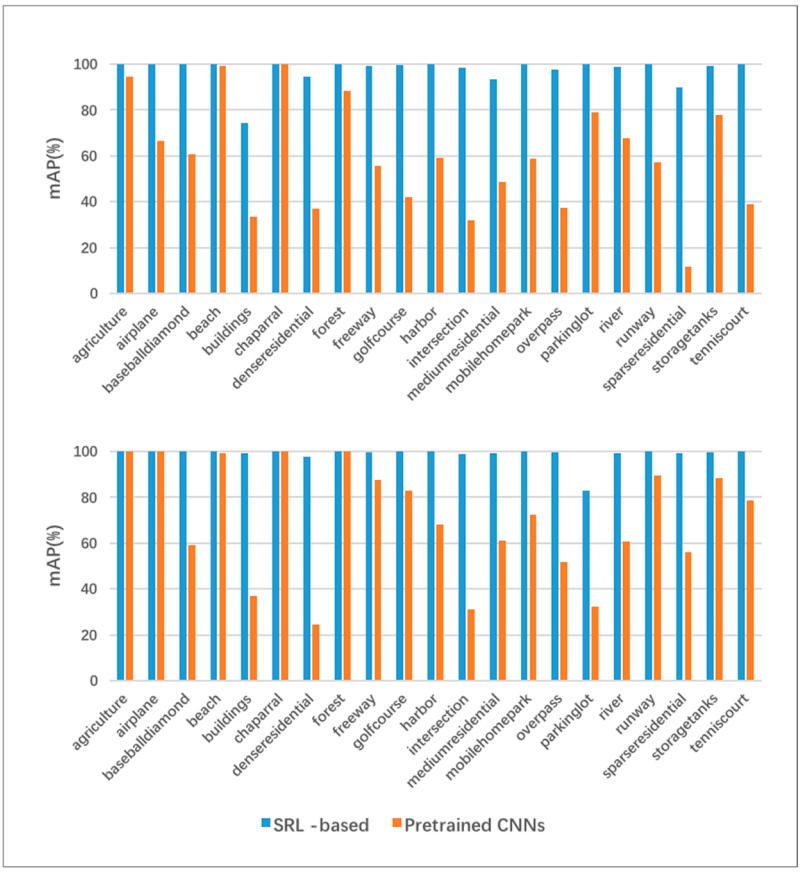


Figure 11. Evaluation of mAP is performed with VGG16 (top) and ResNet50 (bottom) on per-class in the UCMD datasets trained with different features.

5. Conclusions

In this work, we propose a deep metric learning based on Similarity Retention Loss for image retrieval and apply it to the CBRSIR, which is a key technology to effectively utilize the growing quality of remote sensing images. The SRL combines the image features of remote sensing images (only RGB composite images) and improves the algorithm from the following three aspects—the feature aggregation method, sample mining strategy based on information pairs selection and the relative weight calculation of different sample pairs, so as to achieve accurate image retrieval. First, we propose to use the SPoC pooling method to aggregate the convolutional features extracted by the network to adapt to remote sensing images with a large geographic area and rich background information. Second, we propose the concept of similarity retention. By learning the sample distribution around each sample, we separate the negative pairs from the query image into different distances. At the same time, we learn an intra-class threshold for each class to avoid compressing the features of the positive samples to a point and guarantee the structure of the positive samples. Third, we use the similarity as the benchmark and set different thresholds and selection strategies to select positive and negative samples. In this way, the algorithm can ensure that the sample selection is both representative and not redundant.

We test the method on two publicly available datasets and achieve the best performance on both datasets. It is sufficient to prove the effectiveness of the Similarity Retention Loss for deep metric learning in image retrieval. More importantly, our method can also be applied to geographic information research, such as urban road traffic intelligence, environmental testing, natural disaster detection, vegetation mapping, urban planning and research on high-resolution remote sensing image retrieval.

Author Contributions: All the authors contributed to this study. Conceptualization, Hongwei Zhao and Lin Yuan; methodology, Lin Yuan; software, Haoyu Zhao and Lin Yuan; writing, Lin Yuan; writing—review, Hongwei Zhao and Haoyu Zhao. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61841602, the Provincial Science and Technology Innovation Special Fund Project of Jilin Province, grant number 20190302026GX, the Jilin Province Development and Reform Commission Industrial Technology Research and Development Project, grant number 2019C054-4, the Higher Education Research Project of Jilin Association for Higher Education, grant number JGJX2018D10 and the Fundamental Research Funds for the Central Universities for JLU.

Acknowledgments: We would like to thank Peng Wang for his suggestions for coding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Y.; Zhang, D.; Lu, G.; Ma, W.Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognit.* **2007**, *40*, 262–282. [CrossRef]
2. Dharani, T.; Aroquiaraj, I.L. A survey on content based image retrieval. In Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), Periyar University, Tamilnadu, India, 21–22 February 2013; pp. 485–490.
3. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kerkyra, Corfu, Greece, 20–25 September 1999; pp. 1150–1157.
4. Yang, Y.; Newsam, S.J.I.T.o.G.; Sensing, R. Geographic image retrieval using local invariant features. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 818–832. [CrossRef]
5. Özkan, S.; Ateş, T.; Tola, E.; Soysal, M.; Esen, E. Performance analysis of state-of-the-art representation methods for geographical image retrieval and categorization. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1996–2000. [CrossRef]
6. Sünderhauf, N.; Shirazi, S.; Jacobson, A.; Dayoub, F.; Pepperell, E.; Upcroft, B.; Milford, M. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In Proceedings of the Robotics: Science and Systems XII, Rome, Italy, 13–17 July 2015.
7. Babenko, A.; Slesarev, A.; Chigorin, A.; Lempitsky, V. Neural codes for image retrieval. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 584–599.
8. Noh, H.; Araujo, A.; Sim, J.; Weyand, T.; Han, B. Large-scale image retrieval with attentive deep local features. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3456–3465.
9. Napoletano, P. Visual descriptors for content-based retrieval of remote-sensing images. *Int. J. Remote Sens.* **2018**, *39*, 1343–1376. [CrossRef]
10. Ye, F.; Xiao, H.; Zhao, X.; Dong, M.; Luo, W.; Min, W.J.I.G.; Letters, R.S. Remote sensing image retrieval using convolutional neural network features and weighted distance. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1535–1539. [CrossRef]
11. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 197–209. [CrossRef]
12. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
13. Lowe, D.G. Similarity metric learning for a variable-kernel classifier. *Neural Comput.* **1995**, *7*, 72–85. [CrossRef]
14. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K.-R. Fisher discriminant analysis with kernels. In Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468), Copenhagen, Denmark, 13–15 September 1999; pp. 41–48.
15. Xing, E.P.; Jordan, M.I.; Russell, S.J.; Ng, A.Y. Distance metric learning with application to clustering with side-information. In Proceedings of the Advances in neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 8–13 December 2003; pp. 521–528.
16. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by tracking: Siamese CNN for robust target association. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 1–26 June 2016; pp. 33–40.

17. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 1–26 June 2016; pp. 1420–1429.
18. Gordo, A.; Almazan, J.; Revaud, J.; Larlus, D. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vision* **2017**, *124*, 237–254. [CrossRef]
19. Xu, X.; He, L.; Lu, H.; Gao, L.; Ji, Y. Deep adversarial metric learning for cross-modal retrieval. *Wide Web* **2019**, *22*, 657–672. [CrossRef]
20. Xing, Y.; Wang, M.; Yang, S.; Jiao, L.; Sensing, R. Pan-sharpening via deep metric learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 165–183. [CrossRef]
21. Kaya, M.; Bilge, H.Ş. Deep metric learning: A survey. *Symmetry* **2019**, *11*, 1066. [CrossRef]
22. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Toronto, ON, Canada, 20 June 2005; pp. 539–546.
23. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17 June 2006; pp. 1735–1742.
24. Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wang, J.; Philbin, J.; Chen, B.; Wu, Y. Learning fine-grained image similarity with deep ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1386–1393.
25. Oh Song, H.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep metric learning via lifted structured feature embedding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 1–26 June 2016; pp. 4004–4012.
26. Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 1857–1865.
27. Oh Song, H.; Jegelka, S.; Rathod, V.; Murphy, K. Deep metric learning via facility location. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5382–5390.
28. Law, M.T.; Urtasun, R.; Zemel, R.S. Deep spectral clustering learning. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1985–1994.
29. Wang, J.; Zhou, F.; Wen, S.; Liu, X.; Lin, Y. Deep metric learning with angular loss. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2593–2601.
30. Wang, X.; Hua, Y.; Kodirov, E.; Hu, G.; Garnier, R.; Robertson, N.M. Ranked list loss for deep metric learning. *arXiv* **2019**, arXiv:1903.03238.
31. Fan, L.; Zhao, H.; Zhao, H.; Liu, P.; Hu, H. Distribution structure learning loss (DSLL) based on deep metric learning for image retrieval. *Entropy* **2019**, *21*, 1121. [CrossRef]
32. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS), San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
33. Radenović, F.; Tolias, G.; Chum, O. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1655–1668. [CrossRef] [PubMed]
34. Yue-Hei Ng, J.; Yang, F.; Davis, L.S. Exploiting local features from deep networks for image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 24–27 June 2015; pp. 53–61.
35. Babenko, A.; Lempitsky, V. Aggregating deep convolutional features for image retrieval. *arXiv* **2015**, arXiv:1510.07493.
36. Kalantidis, Y.; Mellina, C.; Osindero, S. Cross-dimensional weighting for aggregated deep convolutional features. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, the Netherlands, 8–16 October 2016; pp. 685–701.
37. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
38. Mousavian, A.; Kosecka, J. Deep convolutional features for image based retrieval and scene categorization. *arXiv* **2015**, arXiv:1509.06033.

39. Lee, C.-Y.; Gallagher, P.W.; Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated and tree. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain, 9–11 May 2016; pp. 464–472.
40. Bell, S.; Bala, K. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph. TOG* **2015**, *34*, 98. [CrossRef]
41. Harwood, B.; Kumar, B.; Carneiro, G.; Reid, I.; Drummond, T. Smart mining for deep metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2821–2829.
42. Wu, C.-Y.; Manmatha, R.; Smola, A.J.; Krahenbuhl, P. Sampling matters in deep embedding learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2840–2848.
43. Ge, W. Deep metric learning with hierarchical triplet loss. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 269–285.
44. Hoffer, E.; Ailon, N. Deep metric learning using triplet network. In Proceedings of the International Workshop on Similarity-Based Pattern Recognition, Copenhagen, Denmark, 12–14 October 2015; pp. 84–92.
45. Ustinova, E.; Lempitsky, V. Learning deep embeddings with histogram loss. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 4170–4178.
46. Yi, D.; Lei, Z.; Li, S.Z. Deep metric learning for practical person re-identification. *arXiv* **2014**.
47. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
48. Simo-Serra, E.; Trulls, E.; Ferraz, L.; Kokkinos, I.; Moreno-Noguer, F. Fracking deep convolutional image descriptors. *arXiv* **2014**, arXiv:1412.6537.
49. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 1–26 June 2016; pp. 770–778.
51. Vedaldi, A.; Lenc, K. Matconvnet: Convolutional neural networks for matlab. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 689–692.
52. Razavian, A.S.; Sullivan, J.; Carlsson, S.; Maki, A. Applications. Visual instance retrieval with deep convolutional networks. *ITE Trans. Media Technol. Appl.* **2016**, *4*, 251–258. [CrossRef]
53. Cao, R.; Zhang, Q.; Zhu, J.; Li, Q.; Li, Q.; Liu, B.; Qiu, G.J.a.p.a. Enhancing Remote Sensing Image Retrieval with Triplet Deep Metric Learning Network. *arXiv* **2019**, arXiv:1902.05818. [CrossRef]
54. Chaudhuri, U.; Banerjee, B.; Bhattacharya, A. Siamese graph convolutional network for content based remote sensing image retrieval. *Comput. Vision Image Underst.* **2019**, *184*, 22–30. [CrossRef]
55. Demir, B.; Bruzzone, L. Hashing-based scalable remote sensing image search and retrieval in large archives. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 892–904. [CrossRef]