International Journal of Geo-Information

MDPI

Article

Image Retrieval Based on Learning to Rank and Multiple Loss

Lili Fan [1], Hongwei Zhao [1,2,3], Haoyu Zhao [4], Pingping Liu [1,2,*] and Huangshui Hu [5]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[3] State Key Laboratory of Applied Optics, Changchun 130033, China
[4] Editorial Department of Journal (Engineering and Technology Edition), Jilin University, Changchun 130012, China
[5] School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China
* Correspondence: liupp@jlu.edu.cn

Received: 26 June 2019; Accepted: 26 August 2019; Published: 4 September 2019

Abstract: Image retrieval applying deep convolutional features has achieved the most advanced performance in most standard benchmark tests. In image retrieval, deep metric learning (DML) plays a key role and aims to capture semantic similarity information carried by data points. However, two factors may impede the accuracy of image retrieval. First, when learning the similarity of negative examples, current methods separate negative pairs into equal distance in the embedding space. Thus, the intraclass data distribution might be missed. Second, given a query, either a fraction of data points, or all of them, are incorporated to build up the similarity structure, which makes it rather complex to calculate similarity or to choose example pairs. In this study, in order to achieve more accurate image retrieval, we proposed a method based on learning to rank and multiple loss (LRML). To address the first problem, through learning the ranking sequence, we separate the negative pairs from the query image into different distance. To tackle the second problem, we used a positive example in the gallery and negative sets from the bottom five ranked by similarity, thereby enhancing training efficiency. Our significant experimental results demonstrate that the proposed method achieves state-of-the-art performance on three widely used benchmarks.

Keywords: multiple loss function; computer vision; deep image retrieval; learning to rank; deep learning

1. Introduction

The goal of instance-image retrieval is to quickly and automatically search images that are same or similar, with the query image from a large but unordered database, and return the results to users, according to related ranking. Convolutional Neural Network (CNN), as a tool for deep learning [1], has made great breakthroughs in terms of computer vision. As a category of CNN model, a pre-trained CNN model is one pass where the success lies in feature extraction and encoding steps. ResNet and GoogleNet in pre-trained networks have won the challenges of ImageNet Large Scale Visual Recognition (ILSVRC) in 2014 and 2015. Though the pre-trained CNN model has achieved remarkable retrieval performance, the fine-tuning of current CNN models on the specified training sets still needs to improve. When using a fine-tuned CNN model, image-level descriptors are typically generated end-to-end, and the network will produce a final visual representation without the need for additional explicit encoding or merging steps. Fine-tuning the CNN model includes fine-tuning for datasets

and network-oriented fine-tuning. The datasets used to fine-tune the network are crucial for learning high-resolution CNN features. This is because ImageNet merely provides class labels for images, which can be classified by pre-trained CNN models. However, it is difficult to distinguish images of the same classification. So, it is necessary to train task-oriented datasets by fine-tuning the CNN model. The fine-tuned structures of CNN model fall into two main categories: classification-based networks and verification-based networks. Classification-based networks are trained to classify landmarks into predefined categories. A verification-based fine-tuning network applies a Siamese network combined with a pairwise loss function or a triple loss function, which can significantly improve the adaptability of the network [2], but the training data needs to be further annotated. The first fine-tuning method requires much manual work to collect images and mark them as specific architectural categories, which improves the accuracy of retrieval. However, its formula is closer to the image classification rather than the expected attributes of instance retrieval. Another method uses a geotagged image database to perform training by separating matching and non-matching pairs, and directly optimizes the similarity metric to be applied in the final task [3]. In this process, metric learning plays a key role.

Within the frameworks of metric learning, loss functions form the key component and current works have proposed various loss functions in which contrastive loss masters the interactional relationships among pairwise data points, with similarity and dissimilarity as a case in point [4]. Triplet losses have also been intensively studied [5,6], with an anchor point, a similar (positive) data point and dissimilar (negative) data point constituting a triplet. Triplet losses aim to learn a distance metric, where the anchor point closes to the similar data point, rather than the dissimilar ones through a margin. The model of triplet loss training has significant randomness when selecting samples, and it takes a long time, which leads to relatively large intra-class spacing, and it also has weak generalization ability from training to testing. Therefore, the quadruplet network [7], the triplet loss with batch hard mining (TriHard loss) [8] and the mining network of marginal sample [9] came into existence. However, these conjoined networks usually rely on a simple network structure. The network architecture involves the collection and aggregation of regions. Its accuracy and robustness of image retrieval is low, and more importantly, the existing metric learning network is characterized by shortening the distance between the query image and positive samples, meanwhile widening the distance towards negative samples. The same value is adopted in the distance setting between the sample and the query image. However, not all negative samples have the same dissimilarity with the query image. Thus, designing a new CNN network and learning strategies are important for making the most of training data.

In this work, we focus on learning to rank (L2R) problems that have well-matched image retrieval tasks. We found that deep learning methods lag behind existing techniques. This is due to the insufficient supervised learning of particular tasks for processing instance-level images [10]. The CNN-based retrieval method usually distinguishes different semantic categories by learning local features extracted by pre-trained network on ImageNet, ready for classification tasks. However, the disadvantage is that the intra-class differences are relatively large. We propose a solution tailored to such problems to retrieve the heavy training process.

Our main contributions are as follows:

1. We propose a novel multiple loss-based ranking to learn discriminative embeddings. In contrast with current losses, we firstly incorporate learning to rank, which learns image features according to their true ranking; besides, we propose a learning of sorted information of every sample, in order to preserve the similarity structure inside it. This avoids the shortcomings that the intra-class data distribution might be dropped due to separate negative pairs into equal distance in the embedding space.
2. We use both hard non-matching examples, also referring to negative examples, and hard-matching examples, namely positive examples prepared for the training of CNN model, and also select samples according to their similarity between the query image and examples, so as to enhance training efficiency.

3. We achieve state-of-the-art performance on three benchmarks, i.e., Oxford Buildings [11], Paris [12], and Holiday [13].

The following sections of this paper are organized as follows. Section 2 discusses related works. Section 3 describes the algorithm framework and its details. Section 4 summarizes the main contributions and evaluates the proposed method of learning to rank and multiple loss (LRML) by a series of experiments. Section 5 reviews and concludes the major points.

2. Related Work

We discuss related works revolving around our main contributions, i.e., the image retrieval based on CNN model, the fine-tuned network and the deep Metric Learning algorithm.

A. CNN-Based Image Retrieval.

More recent approaches to image retrieval replace the low-level hand-crafted features with deep convolutional descriptors obtained from convolutional neural networks (CNNs), typically pre-trained on large-scale datasets such as the ImageNet, forming compact image representations, so as to present a promising direction. Early approaches to applying CNNs for image retrieval included methods that set the fully-connected layer activations to be the global image descriptors [1,14]. Husain proposes a global descriptor REMAP based on CNN, which learns and aggregates the hierarchical structure of deep features from multiple CNN layers, and learns discriminative features which are mutually-supportive and complementary at various semantic levels of visual abstraction [15]. Perronnin and Jegou et al. aggregated deep convolutional descriptors to form image signatures using Fisher Vectors (FV) [16], Vector of Locally Aggregated Descriptors (VLAD) [17] and alternatives [18–20]. Tolias et al. [21] have proposed R-MAC to produce a global descriptor by aggregating the activation features of a CNN. One step further is the weighted sum pooling of Kalantidis et al. [22], which can also be seen as a way to perform transfer learning. Popular encodings such as BoW, BoW-CNN and Fisher vectors are adapted in the context of CNN activations in the work of Kalantidis et al. [22], Mohedano et al. [23] and Ong et al. [24], respectively. In image retrieval, the query expansion is used to enhance the image retrieval efficiency [25–27]. Shen [28] proposed using a small number of training images to learn low-dimensional subspaces, by preserving neighbor-reversibility (NR) correlation to enhance training efficiency.

In this work, we propose that by adding large visual code books [20,29] and spatial verification [28,29], CNN can control the image retrieval tasks by outweighing state-of-the-art methods and have achieved a higher maturity level.

B. Finetuning for Retrieval.

In this study, we treat instance retrieval as a metric learning problem, i.e., the Euclidean distance well captures the similarity on the basis that an image embedding is learned. Metric learning by representative architectures employ matching and non-matching pairs to perform training and be applicable for the task. Here, the annotations have become striking, e.g., for classification, only an object category label is needed, while labels per image pair are needed for particular objects. There may exist huge differences when comparing two images of the same object, e.g., images of buildings from different viewpoints. So, it is necessary to fine-tune the CNN model, based on task-oriented datasets. Presently, through minimizing classification errors, ImageNet datasets [30] are used by employed networks trained for image classification. Babenko et al. [14] have further retrained such networks by using a dataset closer to the target task. They conducted training with object classes that match particular landmarks/buildings. They achieved significant improvement with the performance on standard benchmark tests. Although obtaining achievements, there are still differences in terms of utilized layers and the truly optimized ones during learning. Much manual effort is required in building up such training datasets. Recently, geotagged datasets with time stamps have equipped weakly-supervised finetuning with a triplet network [3]. Two images are easily classified as non-matching if they are

taken far from each other, while the most similar images are considered as matching examples. In the latter, the current representation of the CNN basically defines similarity. The above presents a new approach, trying end-to-end finetuning for image retrieval especially for the task of geo-localization. The used training images are currently more concerned with the final task. Our point of difference is finding matching and non-matching image pairs in an automatic way. Moreover, matching examples are derived on the basis of 3D reconstructions, allowing for more challenging examples. We also select positive examples by using local features and geometric verification [31]. We have a fully automatic method which starts from manually cleaned datasets of landmarks, so as to avoid exhaustive evaluation by using landmarks of the datasets, instead of the geometry.

C. Deep Metric Learning.

The goal of deep metric learning is to learn an optimal metric to minimize the distance between similar images. The typical architecture used for metric learning is two-branch Siamese [32,33] and the triplet network [5,34]. These methods can complete the image retrieval by shortening the absolute distance of the matching pair and widening the relative distance of the non-matching pair. The traditional triplet network randomly extracts three pictures from the training data. Although this method is simple, most of the extracted images pairs are simple and easy to distinguish. If a large number of trained image pairs are simple, then it is not conducive to better representation of the network [31]. Based on the triplet networks, a training-based batch hard mining-TriHard Loss comes into being. Its core idea is to select a positive sample farthest from the query image for each training batch, and a closest negative sample and query picture to form a triple. TriHard's loss effect is better than the traditional triple loss function [31], which only considers the relative distance between positive and negative samples. In comparison, Chen introduced a quadruplet loss, which only considers the absolute distance between positive and negative samples. The quadruplet loss has increased inter-class variations and reduced intra-class variations, which allows the model to achieve better representations [7]. Xiao designed the margin sample mining loss (MSML)-LSTM architecture with contrastive loss, which is a metric learning method that introduces batch hard mining. However, Varior ignores the influence of parameters in quadruple loss and adopts a more generalized form to represent the quadruplet loss. In summary, the TriHard loss prepares a triple for each image in the batch, and the MSML loss only picks the hardest positive sample pair and the hardest negative pair to carry out loss calculation, which is a hard-positive sample that is more difficult than TriHard. It considers both the relative distance and the absolute distance, and introduces a metric learning method of batch hard mining [22].

3. Methods

This section describes the network we have proposed and the methods of training.

3.1. Network Architecture

In this paper, we propose a network architecture based on L2R and multiple loss (LRML), which associates image retrieval problems with L2R. The model mainly includes the following main steps:

- Step 1: Extract feature vectors. Extract the underlying features of the query image and those from the training database. Calculate the Euclidean distance between the extracted query image and the underlying features of all the images in the training database. The training data is divided into positive images and negative images. The query image, positive images and negative images are input into networks for features extraction.
- Step 2: Obtain the real sequence. Calculate the Euclidean distance between the query image and negative images, and obtain the real ranking sequence of negative examples.
- Step 3: Acquire initial parameters. Randomly select multiple images and extract underlying features of the query image and currently selected multiple images. Obtain the initial parameters of the deep networks.

- Step 4: Model training. Assign the real sequence number of the training data to the negative examples, and combine the sequence number with its threshold value. Calculate the loss value by using the multiple Siamese loss. Adjust the distance between the negative examples and feature vectors of the query picture, as well as adjusting the initial parameters of the deep convolutional network by back-propagation and sharing weights to obtain the final parameters of the deep convolutional network.

Next, we show a detailed design and analysis of each key component in the model.

3.1.1. End-to-End Multivariate Ranking Learning Network

In this section, we will describe how our network works during the training and test phases.

Figure 1 presents a possible CNN structure for our network Figures 2 and 3. Here we take AlexNet [35], VGG16 [36] and ResNet101 [3] as examples. We remove the last pooling layer and all-connected layer of the network as our CNN basic structure, and then connect our GeM pooling, Lw whitening and L2 regularization to form new network structure.

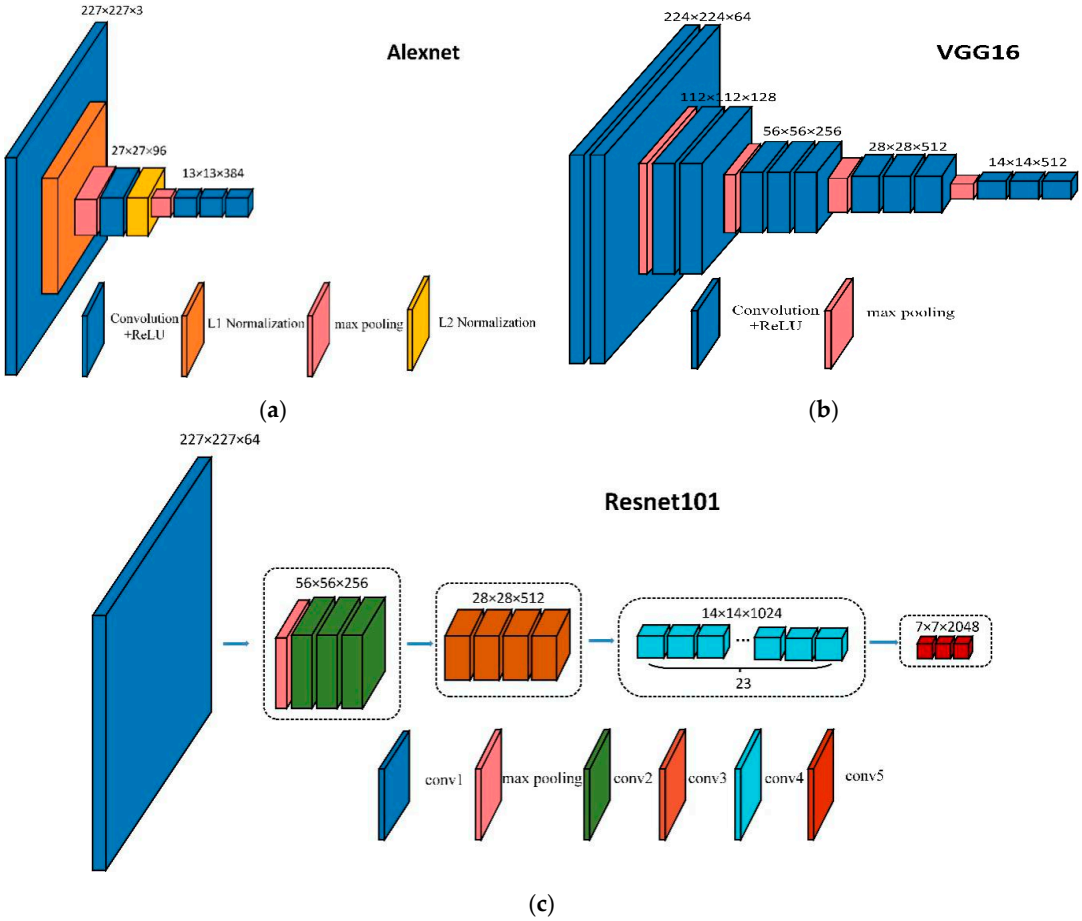

Figure 1. Convolutional Neural Network (CNN) network structure: (**a**) AlexNet; (**b**) VGG16; (**c**) ResNet101.

Training phase: As shown in Figure 2, in the training phase, the model starts with processing multiple original images, where we represent the images as q and i, respectively. In detail, q is the query image, and i consists of a positive image and five negative images. The goal of the training phase is to learn effective feature representations. We input the image into the CNN network. The network structure of CNN is shown in Figure 1. We use the CNN network, which has removed the last pooling layer and has fully connected layers to extract relevant features. The feature is then connected to

the GeM pooling, and the feature after GeM pooling is followed by the L2 regularization operation. Finally, we obtain the feature vector consisting of different features. We add the previously obtained sorting number based on image similarity to the vector, perform loss calculation, and optimize the loss function by zooming in on the distance between the positive sample and the query image. We also sort the negative samples according to the similarity. In the end, we obtain the parameters of the network, and further the structure of the network.

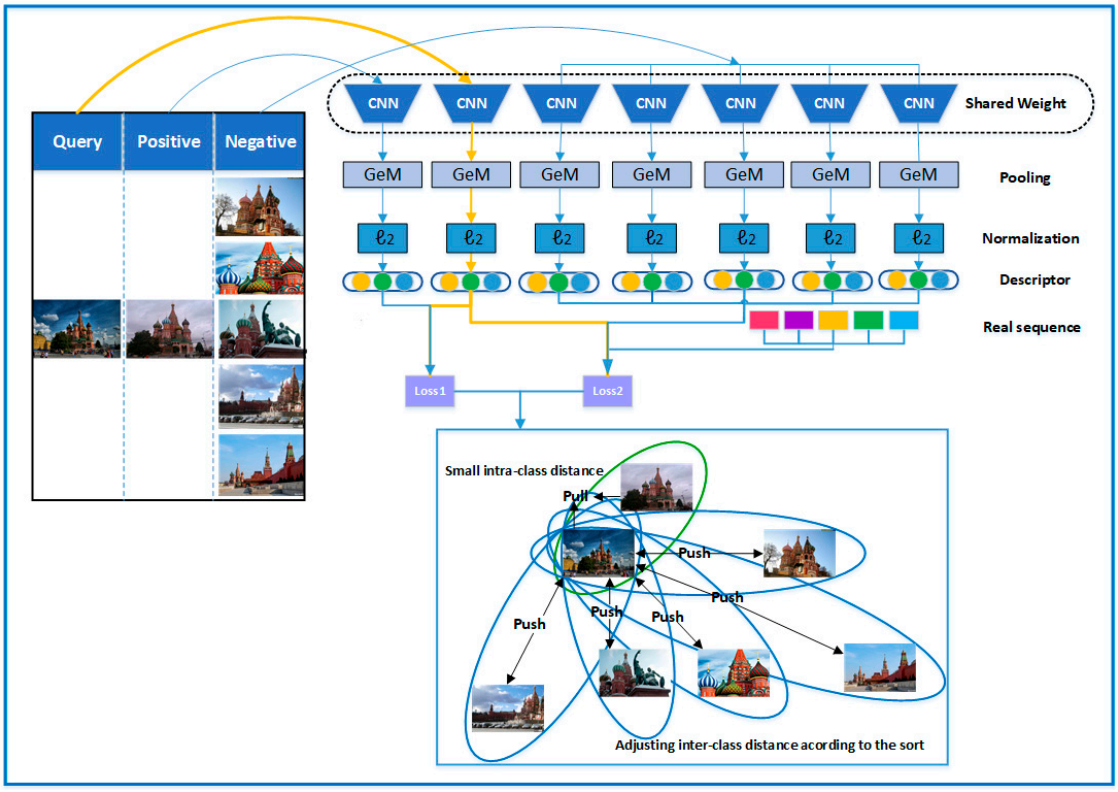

Figure 2. Architecture of the multiple learning-to-rank training network.

Test phase: Figure 3 presents the testing phase of the CNN model, we will conduct multiple processing of the query image and images in the database. Then the images are input into the CNN for feature extraction. The CNN at this time is the same as the CNN used in the training process. The parameters are identical to those trained by the CNN model. The output of features from the CNN will undergo GeM pooling operation, LwPAC operation and L2 regularization processing. We finally get the feature vector that can represent image features. After obtaining the feature vector, we firstly calculate the Euclidean distance of feature vectors between the query image and images in the database. We then firstly rank the images, perform the query expansion operation and obtain the target image for retrieval.

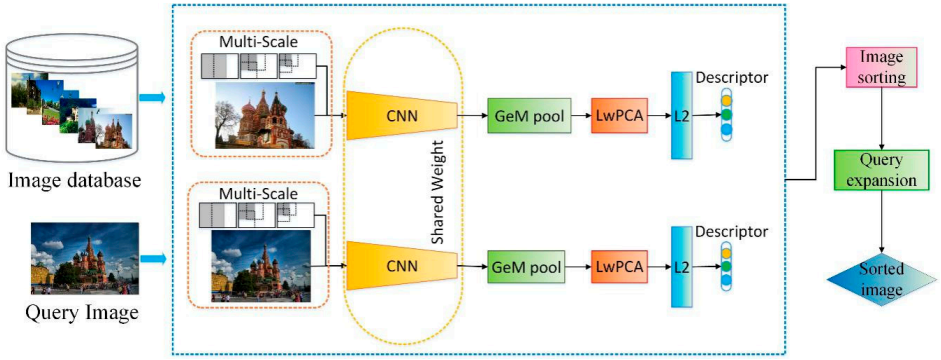

Figure 3. Architecture of the test network.

Ranking-Based Multiple Loss Function: As mentioned above, we define multiple loss as the final loss function. First, we need to train a two-branch Siamese network. This network is identical, except for the loss function. The two branches of the network not only share the same network structure, but also network parameters. The multiple loss function based on ranking includes two parts, q as the query image, i as input image, and Y is the label to distinguish matching or non-matching. Each query image q corresponds to i, with Y(q,i) belong to [1]. If i is a positive image compared to q, then the value of Y(q,i) is equal to 1; if i is a negative image compared to q, then the value of Y(q,i) is equal to 0.

$\overline{X}$(q) represents the visual information feature extracted from a query picture q. $\overline{X}$(i) represents a visual feature information vector extracted from an input picture i. ||X(q) − X(i)|| represents the Euclidean distance between $\overline{X}$(q) and $\overline{X}$(i). n is the number of negative images that do not match the query image q, $e^{\alpha/n}$ is the set threshold, α is the real ranking ordinal number of the image i. If there are five samples, the value of α is 0, 1, 2, 3, 4, and henceforth n is equal to 5. For images that are highly correlated with the query image and have been marked as positive images in the datasets, namely Y(q,i) = 1, the closer Euclidean distance between these images and the query image in feature space, then the loss function also increases.

$$\text{Loss(q, i)} = \begin{cases} \frac{1}{2}\|\overline{X}(q) - \overline{X}(i)\|^2, \ if\ Y(q,i) = 1 \\ \frac{1}{2}\left(max\left\{0, e^{\frac{\alpha}{n}} - \|\overline{X}(q) - \overline{X}(i)\|\right\}\right)^2, \ if\ Y(q,i) = 0 \end{cases} \quad (1)$$

Euclidean distance between the query image and positive images. While images with low correlation with the query image will be marked with $Y(q, i) = 0$ in the training datasets, for these images, if the Euclidean distance between these images and the query image is lower than threshold, then the loss is calculated.

3.1.2. Whitening and Dimensionality Reduction

Current methods [37] adopt PCA, as an independent method to conduct whitening and dimensionality reduction. In this respect, all descriptors and their covariance matrix are analyzed. In this part, Radenović took post-processing of fine-tuned GeM vectors into consideration [27]. It is suggested to leverage the marked and labeled data sourced from 3D models, and adopt the method of linear discriminant projection initially proposed by Radenović [27]. This projection consists of two parts, namely whitening and rotation. The whitening refers to the inversed square-root in terms of intra-class covariance matrix $C_S^{-1/2}$, where:

$$C_S = \sum_{Y(q,i)=1} \left(\overline{X}(q) - \overline{X}(i)\right)\left(\overline{X}(q) - \overline{X}(i)\right)^T \quad (2)$$

The rotation is the PCA of the inter-class (non-matching pairs) covariance matrix in the whitened space eig($C_S^{-1/2}$, C_D, $C_S^{-1/2}$), where:

$$C_D = \sum_{Y(q,i)=0} \left(\overline{X}(q) - \overline{X}(i)\right)\left(\overline{X}(q) - \overline{X}(i)\right)^T \quad (3)$$

The projection $P = C_S^{-\frac{1}{2}} eig(C_S^{-\frac{1}{2}} C_D C_S^{-\frac{1}{2}})$ is then applied as $P^T(f(i) - u)$, where μ is the GeM pooling vector. In order to reduce the descriptor dimension to the D dimension, only the feature vectors in accordance with the D largest eigenvalues are used.

3.2. Network Training

We used a transfer learning policy in the training process of the LRML network. It is unnecessary to prepare any rigid format of the input data in our proposed method, e.g., triplets, n-pair triplets. Instead, it takes random input images with multi-class samples. We conduct online iterative ranking

and loss computation after obtaining images' real sequence. On the same image datasets, we illustrate the comprehensive process to train the model of LRML, and the algorithm is presented in Table 1.

Table 1. Algorithm proposed by us.

Algorithm 1 **Learning to Rank and Multiple Loss**

1: Parameters Setting: The number of negative sample n, the sorting number of negative sample a, learning rate η, the initial weights w, the initial biases b.
2: Input: q (query sample), i (retrieve image), the learning rate β, the embedding.
3: Output: Updated L

(a) Using prior knowledge find the set of samples S, such that $\overline{X}_i$ is deemed similar to $\overline{X}_q$.

(b) Pair the sample $\overline{X}_q$ with all the other training samples and label the pairs so that: $Y(q,i) = 1$ if X_i belongs to S, and $Y(q,i) = 0$ otherwise.

Combine all the pairs to form the labeled training set.
Step 1: Feedforward all images into f to obtain the images' embedding Xi.
Step 2: Calculate the Euclidean distance between all negative samples and the query image, online iterative ranking, and get the images' number of sorting(a) a repeat until convergence:

(a) For each pair (X_q, $\overline{X}_i$) in the training set, do

i. If $Y(q,i) = 1$, then update w to decrease

$$L_{(w,b)} = \frac{1}{2}\|\overline{X}(q) - \overline{X}(i)\|^2$$

ii. If $Y(q,i) = 0$, then update w to increase

$$L_{(w,b)} = \frac{1}{2}\left(max\left\{e^{\frac{a}{n}}\tau - \|\overline{X}(q) - \overline{X}(i)\|\right\}\right)^2$$

Step 3: Gradient computation and back-propagation to update the parameters of w, b.

$$w' = w - \eta\frac{\partial_{L_{(w,b)}}}{\partial_w} \qquad b' = b - \eta\frac{\partial_{L_{(w,b)}}}{\partial_b}$$

4. Experimental Results and Analysis

In this section, we illustrate the training process and a detailed procedure of implementation. We will discuss the implementation details of our training process, assess he different components of our model, and compare the experimental results with the current techniques.

4.1. Data Collection

4.1.1. Acquisition of Training Datasets

We adopt the training dataset constructed by Schonberger et al. [35]. The dataset includes 7.4 million images, which are searched and downloaded by Flickr, with popular landmarks throughout the world. The dataset uses bag-of-word model (BoW) and structure-from-motion (SfM) to reconstruct the 3D model, and uses a method exempt from manual annotation to automatically obtain a large dataset with a query image, a positive image, and a cluster with serial number. There is a total of 91,642 training images in the dataset, and 98 cluster images identical or nearly identical to the test dataset can be excluded through image retrieval based on BoW. About 20,000 images are selected as query images, 181,697 pairs of positive images, and 551 training clusters, including more than 163,000 from original datasets, by the minimum hash and spatial verification methods mentioned in the clustering procedure [27]. The original datasets contain all the images of the Oxford5k and Paris6k datasets.

4.1.2. Selection of Training Datasets

In this paper, we create a tuple dataset (Xq, Xi), where q is the query image, i is a positive image that matches the query image, and these tuples are used to make image pairs for later training.

Selection of positive images: During the training process, several sets of images are randomly selected from positive image pairs. The annotated positive image pairs from the training dataset will be treated as positive images inside the training sets. In [27], the acquisition of a positive sample is through the following three methods.

CNN descriptor distance: The positive image refers to those with the lowest descriptor distance to the query, formally:

$$m_1(q) = \underset{i \in M(q)}{argmin} \|\overline{X}(q) - \overline{X}(i)\| \tag{4}$$

The GPS coordinates of the positive image are the same as the query. Consequently, the images that are chosen will get a small loss since they already have small descriptor distance. Thus, the drawback is that the network is not forced to dramatically change and learn by the matching examples.

Maximum inliers. The positive image is chosen by 3D information, which is independent of the CNN descriptor. Thus, the image has the largest number of co-observed 3D points with the chosen query. That is:

$$m_2(q) = \underset{i \in M(q)}{argmin} \left| P(q) \cap P(i) \right| \tag{5}$$

The number of features with spatial verifications between the two images correspond with this measure, which commonly applies to ranking in BoW-based retrieval. Since this measure is not influenced by the representation of the CNN model, it requires delivering more challenging positive examples.

Relaxed inliers. The positive image pair is randomly selected from a set of images, rather than use a pool of images captured with similar positions of the camera. This image shares a sufficient number of co-observed points with the query image, and does not show extreme scale changes. This positive image is:

$$m_3(q) = rnd\left\{ i \in M(q) : \frac{P(q) \cap P(i)}{P(q)} \geq t_i,\ scale(i,q) \leq t_s \right\} \tag{6}$$

In (6), the scale changes between the two images are reflected from scale (i,q). The harder matching examples selected by this method are guaranteed to ensure the depiction of the same object. We proposed three different methods to exhibit the queries and their corresponding positive ones. The relaxed method is conducive to increase more diversified viewpoints.

Selection of negative images: We select negative examples from clusters that differ from the query image. In the process of training the dataset, we use the training parameters and test methods used by Radenović to extract the image features in the dataset. Here, we use the VGG16 as fine-tuning network, and adopt Generalized-Mean (GeM) pooling to extract salient features, learning whitening to reduce dimensionality and average query expansion to improve retrieval accuracy. By calculating the Euclidean distance between the extracted query image and the feature vector of the images in the dataset, we randomly select a number of negative examples in the training dataset as the pool of low correlation images. In each round of training, we first select the same N image clusters with the smallest Euclidean distance corresponding to the feature vector of the query image. As shown in the Figure 3, q is the query image, and the clusters where A, B, C, D, and E locate are negative clusters that observe far Euclidean distance to the query image. Suppose we choose A, B, C, D, E as negative examples. If we want to select 5 low correlation negative examples, then we first consider image A, and if image A is not in the query cluster q, or in the positive example clusters, then image A is used as a low correlation image listed in the input set of query image q. Similarly, image B is also marked with low correlation in the input set of images. For image C, although there exists large distance of Euclidean distance between its feature vector and the feature vector of the query image, image C and image B belong to one labeled cluster. So, image C, as a low correlation image, is not taken in the

negative image set of the query image q. Images D, E, and F are taken as low correlation images to q. When the number of the required image equals to N, the low correlation image is no longer selected, so image G and other images will no longer be considered.

4.1.3. Acquisition of the Real Sequence

For each selected low correlation image, A, B, C, D, E, F for the query image Q, as shown in Figure 4, we extract the corresponding vectors a′, b′, c′, d′, e′, f′ in a benchmark sort. We calculate the Euclidean distance between them and the query image features, and rank them according to their Euclidean distance to the query image feature vector, the obtained serial number is the ordinal value of the negative correlation image in the loss function, and the obtained sequence is the real ranking sequence of negative examples for the query image. In Figure 5, we present examples of query images and the real ranking sequence. As seen in the figure, the leftmost column is the query image, followed by the negative samples sorted by similarity. From left to right, the similarity decreases. In Figure 6, we show a visual representation of the loss function calculation after the real sequence is obtained.

Figure 4. The selection of negative images: every cluster is formed by similar images; q is the query image; A,B,C,D,E,F are the negative images with low similarity to the query image.

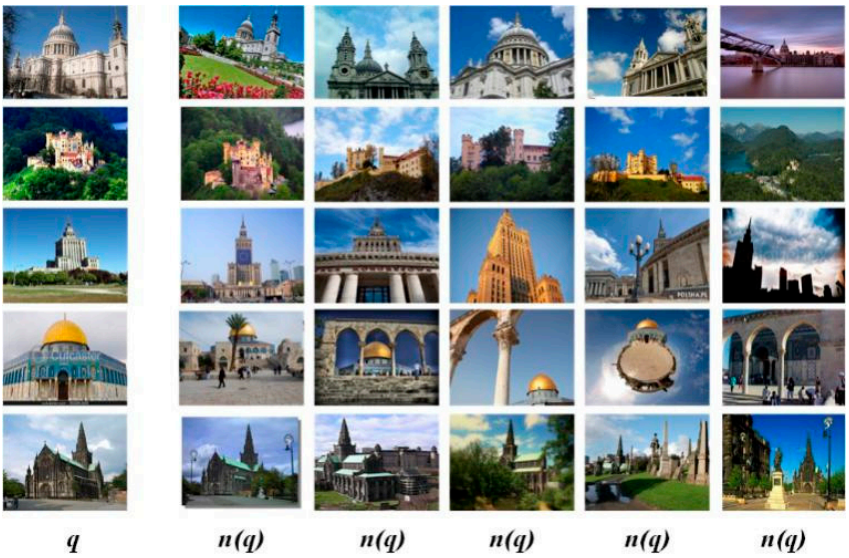

Figure 5. Examples of query images (q) and the real ranking sequence of negative examples (n(q)) for the query image.

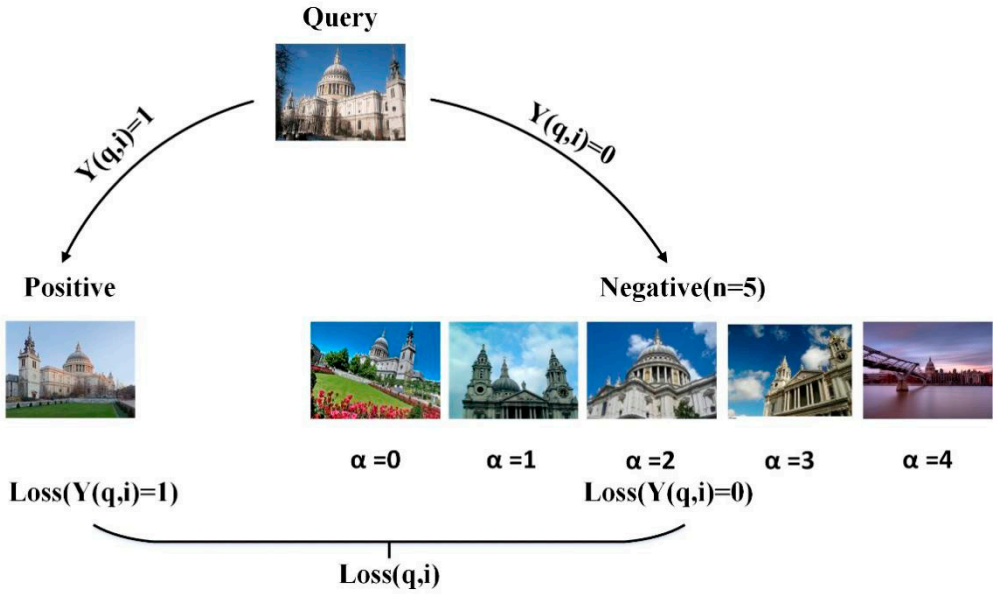

Figure 6. Visual representation of loss calculation.

4.2. Implementation Details

To train the proposed model, we use the pytorch deep learning architecture to train this multiple loss deep network model based on L2R. In order to perform the fine-tuning of networks, we initialize the convolutional layers of AlexNet, VGG 16 and ResNet101, all trained by Adam, and the Adam algorithm is based on the loss function for each parameter. The first moment estimate and the second moment estimate of the gradient are dynamically adjusted for the learning rate of each parameter. Since the network pre-training parameters [36] are used, the learning rate equal to lr = 1×10^{-6}, for the AlexNet network is used during training. The learning rate equal to lr = 7×10^{-7} for the VGG16 and Resnet101 networks, momentum 0.9, margin τ for multiple loss 0.7 for AlexNet, 1.25 for VGG and ResNet, justified by the increase in the dimensionality of the embedding. All of the images in the training set have been resized to a maximum size of 362×362 under the premise of ensuring the original aspect ratio. The training results take the experimental data obtained during the 30 epochs.

The experimental environment is intel(R) i7-8700, GPU with 11GB of memory, NVIDIA(R) 2080Ti graphics card, driver version 419**, operating system Ubuntu 18.04 LTS, pytorch version v1.0.0, CUDA version 10.0, cudnn version 7.5.

4.3. Evaluation Metrics

To evaluate the effect of multiple loss, based on L2R in instance-level image retrieval, we test our method on four standard datasets: the Oxford 5k building dataset [11], the Paris 6k dataset [12], the INRIA Holidays dataset [13]. We combine these datasets with 100k distractors from Oxford 100kd for larger scale evaluation. We crop the query images by adding the bounding box and follow the standard evaluation protocol on the Oxford and Paris datasets. Then we feed the cropped image into the CNN model, and the entire query image for Holiday and UKB is fed as input. To measure the search results, we uniformly use the standards provided on the dataset website, namely calculating mean Average Precision (mAP) of the search results. The mAP value is calculated as shown in Equation (7):

$$\mathrm{mAP} = \frac{1}{|Q_R|} \sum_{q \in Q_R} \mathrm{AP}(q) \tag{7}$$

where AP(q) is mAP of the results of the query image compared with the benchmark annotations in the dataset.

Single-scale evaluation. The image dimensionality input into the CNN is limited to 1024×1024 pixels. In the experiments, no post-processing of vectors is used, if not specifically stated.

Multi-scale evaluation. We only use multi-scale representation during test time. The input images are re-sized into a different size, then multiple inputs are fed into the network, and finally the global descriptors are combined from multiple scales into a single descriptor. Baseline average pooling is compared with GeM pooling, the parameter of which equals the value that was learned in the network of global pooling layer. In this respect, the learning whitening is through final multiple scale image descriptors. In the experiment, we use single scale evaluation, if not specifically stated otherwise.

4.4. Contributions of Add LRML into Contrastive Loss Network

4.4.1. Margin Parameter τ Selection

We trained the model with margin parameter τ at different values to evaluate our network performance. Choosing different margin parameter τ can find the optimal parameters suitable for different datasets. The results are shown in Figure 7. The data shown in Figure 7A,B is trained under AlexNet, while Figure 7C,D presents data trained by the VGG. The results show that whatever is in Oxford5k or in Paris6k, the best results appear when $\tau = 1.25$. Through the image we observe that in our model, the image retrieval performance will be improved to some extent with the increase of τ, but when the value of τ is too large, the distance between the matching pairs with high similarity will be furthered. However, when τ is too small, some negative samples with certain similarities, such as bad samples, will be discarded. This is because fewer samples involved will lead to poor training results. The key to image retrieval based on L2R is to find the range of distance between all samples and query pictures. For the remainder of this article, we use $\tau = 1.25$ in all network and datasets.

4.4.2. Comparison of Different Loss Functions

We use the large number of training pairs generated in [27] as training data. Compared with [3,5,38,39], we find that the contrastive loss has a more powerful convergence function than the triplet loss. Here we compare our multi-loss based on L2R with the current contrastive loss used in [40]. In this experiment, to better observe the impact of ranking accuracy on the proposed model, we use the cosine similarity and the Euclidean distance to measure the similarity between the query image and the images in the training datasets, and obtain two lists with different ranking. The values in the two lists are brought into the function to train the model. We show the results in Figure 8, where

ED represents the similarity measured by Euclidean distance, CS represents that the measurement of trained datasets ranking is cosine similarity, and ED refers to the experimental results of the multiple loss based on L2R. The corresponding τ values are obtained by conducting contrastive loss. It can be seen from the comparison that the performance of image retrieval using our multiple loss function is significantly better than that of contrastive loss function. In addition, we also find that the results in real sequence obtained from Euclidean distance are generally better than those ranked by cosine similarity. This fully demonstrates that multiple loss based on L2R performs better than contrastive loss, and plays a key role in the process of real sequencing of datasets.

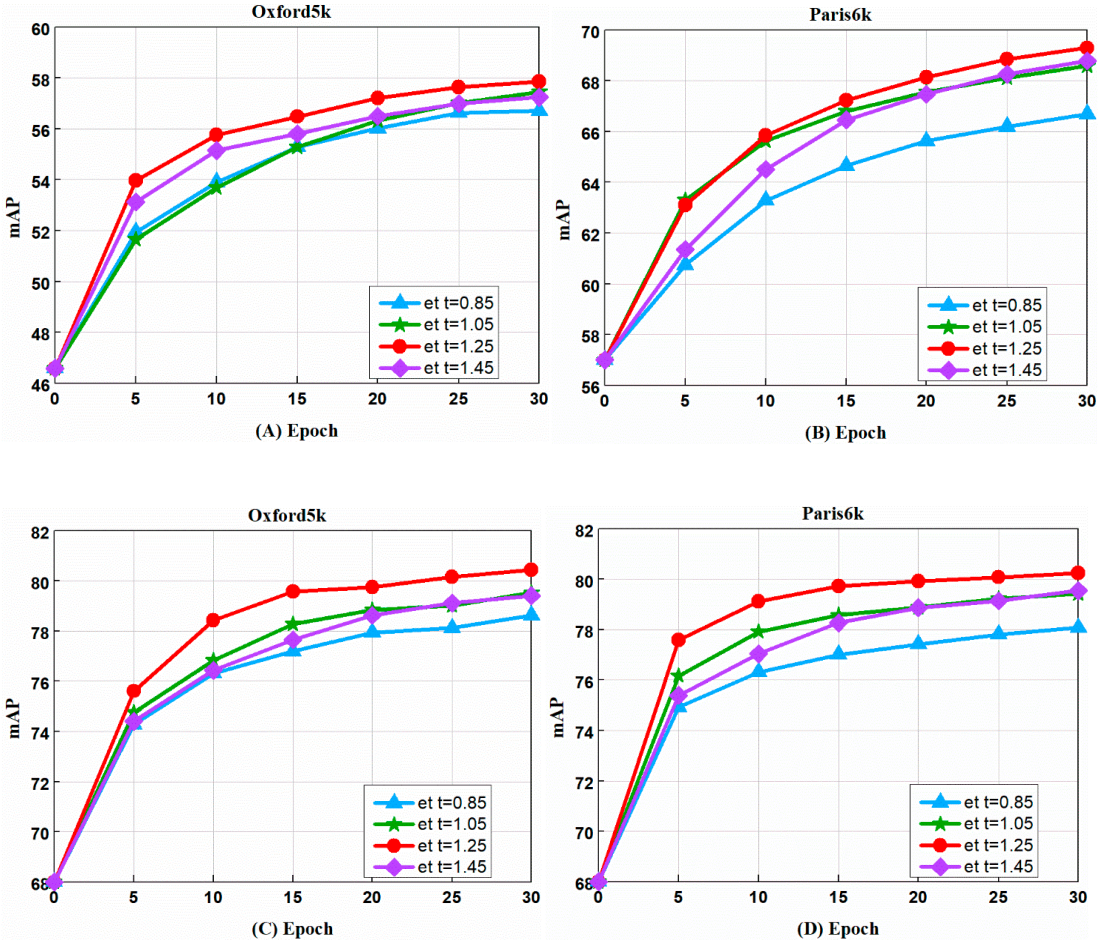

Figure 7. Performance comparison of choice of the different t selection: (**A**) Evaluation is performed with AlexNet with GeM layer on Oxford5K; (**B**) Evaluation is performed with AlexNet with GeM layer on Paris6k; (**C**) Evaluation is performed with VGG with GeM layer on Oxford5k; (**D**) Evaluation is performed with VGG with GeM layer on Paris6k.The curve line presents the evolution of mAP depending on training epochs. Epoch 0 reflects off-the-shelf network. Multiple loss is used in all approaches unless otherwise specified.

4.5. Design Choices of LRML Network

4.5.1. Pooling Methods

We evaluate the impact of different pooling layers on our network and find the pooling method that best fits our network model. Theoretically, when p is 2.32, the performances of GeM is the best. Here we set the value of p to 2.32. We present the results in Table 2. By observing the data in the table, we see that the pooling performance of GeM and the maximum pooling are always better than the current average pooling, whether it is on Oxford5k or Paris6k. So, in the next experiment of this

paper, we combine maximum pooling or GeM with other steps and choose the best combination by comparing the experimental results.

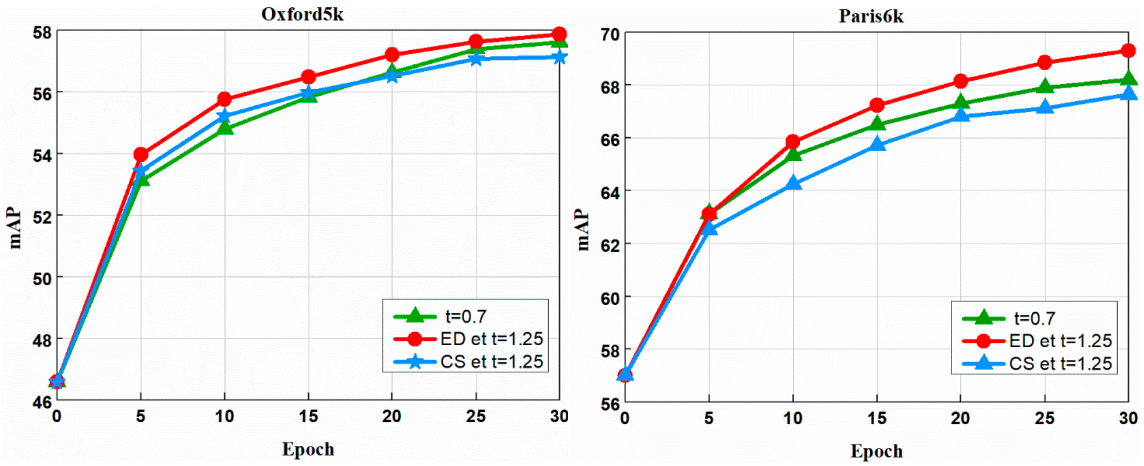

Figure 8. The comparison of performance based on methods of contrastive loss and multiple loss. Evaluation is based on the performance with AlexNet Gem of datasets Oxford5k and uParis6k. The curve line presents the evolution of mean Average Precision (mAP) depending on training epochs.

Table 2. The comparison of performance (mAP) of global max pooling(MAC), average pooling(SPoC) and GeM layers fine-tuned by CNN model. Numbers in bold refers to the best performance.

Pooling	Initial	Learned p	Oxford5k	Paris6k
MAC	inf	-	54.54	66.06
SPoC	1	-	45.62	60.46
GeM	3	2.32	**57.86**	**69.29**

4.5.2. Learned Projections

In the experiment, we compare the previous whitening methods and the newly proposed method of learned discriminative whitening (Lw). The results are shown in Table 3 without post-processing, but with PCAw [40] and Lw [27]. Our experimental results prove that PCAw lowers down the performance, while Lw generally obtains the best performance in the majority of tests and never achieves the worst performance in comparison with others. This complies with the experimental results found by Radenović [27].

Table 3. The comparison of performance (mAP) of post-processing of CNN vector, without post-processing, PCA-whitening [40] (PCAw) and learned whitening (Lw). The reduction of dimensionality is not performed. Fine-tuned AlexNet a 256D vector. Numbers in red and blue refer to the best and worst performances, respectively.

Net	Post	Dim	Oxford5k		Paris6k	
			MAC	GeM	MAC	GeM
AlexNet	-	256	54.54	57.86	66.06	69.29
	PCAw		50.92	62.19	62.89	69.85
	Lw		58.00	67.04	67.74	72.4

During the finetuning process, an additional experiment is conducted to append a whitening layer at the end of the network. By this means, whitening is learned in an end-to-end manner, combined with convolutional filters and the same training data in batch-mode. Specifically, Lw achieves 58 on AlexNet MAC, and 67.74 mAP on both Oxford5k and Paris6k. In contrast, on the same network,

GeM achieves 67.04 mAP on Oxford5k and 72.4 mAP on Paris6k, respectively. In view of these findings, we combined Lw and GeM, because they are trained in a much faster and more effective manner.

4.5.3. Dimensionality of Final Image Vector and Its Impact

We compared the performance of cross-combination of Mac pooling, GeM pooling, PCA whitening and Lw whitening in different dimensions. The performance of image retrieval in image vector from 16 to 256 is shown in Figure 9.

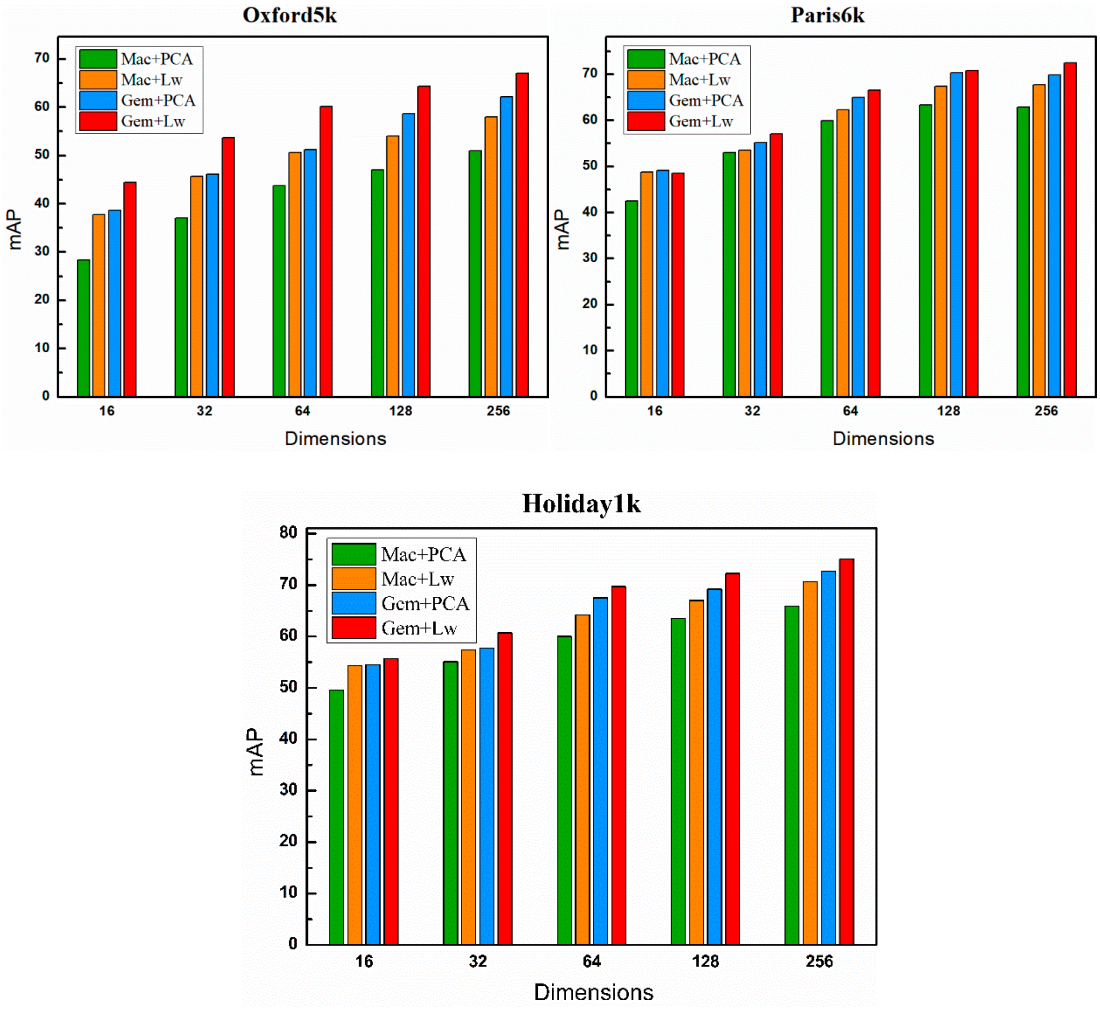

Figure 9. The comparison of performance on the reduction of dimensionality by PCAw and Lw with the fine-tuned AlexNet with global max pooling (MAC) layer and the fine-tuned AlexNet with GeM layer on Oxford5k, Paris6k and Holiday1k datasets.

As can be seen from Figure 9, no matter what the combination, the higher the dimension, the better the performance. In the same dimension, when using the same pooling method, the effect of Lw whitening outweighs the others, and achieves even better results when co-functioning with GeM. Overall, when the dimension is 256, the combination of GeM pooling and Lw whitening has the best performance.

4.5.4. Efficiency

In order to verify the advantages of our proposed LRML algorithm model in terms of training speed, in this part of the experiment, we compare the training time of this model with other classical metric learning models. We run the experiment on intel(R) i7-8700, GPU with 11GB of memory,

operating system Ubuntu 18.04 LTS. In the testing phase, we use VGG16 and ResNet101 as the basic network and calculate training time on the datasets. Table 4 shows the training efficiency of the five methods.

Table 4. The training efficiency of different methods.

Method	Train Time	
	VGG16	ResNet101
Contrastive loss	7.8 h/50 epoch	9.5 h/50 epoch
Triplet loss	8.79 h/50 epoch	10.48 h/50 epoch
Quadruplet loss	8.81 h/50 epoch	10.52 h/50 epoch
Lifted Struct	9.4 h/50 epoch	10.6 h/50 epoch
Ours	8.1 h/50 epoch	10 h/50 epoch

As can be seen from the figure, the training time with our method is shorter than those by Triplet loss and Quadruplet loss, because we use a two-branch network, and the structure is simple. The sample pairs in Lifted Struct are selected in mini-batches, and our method is to choose within the entire data set, so in terms of time, our method has an advantage over Lifted Struct. Compared to contrastive loss, we not only use the same branch structure, but also add a pair of samples based on this sorting operations. This makes our training time slightly longer than contrastive loss, but our outcome is much better. In general, our approach is demonstrated to be the most effective.

Query expansion is a post-processing technology that can effectively improve retrieval performance. It works as follows: in the initial query phase, the feature vector is adopted, and the returned top k results are obtained through the query. The current results will probably experience space verification phase, in which we discard results that do not match the query image. The remaining results are then summed with the original query and experience renormalization. Finally, a second query is performed by combining the descriptors to generate a final list of retrieved images. Query expansion usually leads to a significant increase the level of accuracy. The existing query expansion methods include the average query expansion (AQE) and the weighted (α) query expansion αQE. We used the AlexNet and ResNet to test on the Oxford5k, Paris6k, and Holiday datasets, and the result is shown in Figure 10. The results on AlexNet are shown in Figure 10A–C. The test results are shown in Figure 10D–F on ResNet. We compare the two query expansion methods, and the experimental results are consistent with that found in [27]. The AQE performs very differently on the Oxford5k, Paris6k and Holiday datasets. It performs more prominently on Paris6k, but unstable on the Oxford5k and Holiday datasets. When n = 10, the accuracy rates drop sharply, while αQE is stable on three datasets. We finally set a = 5 and nQE = 50 on Oxford5k and Holiday datasets, and set a = 0 and nQE = 50 on Paris6k datasets.

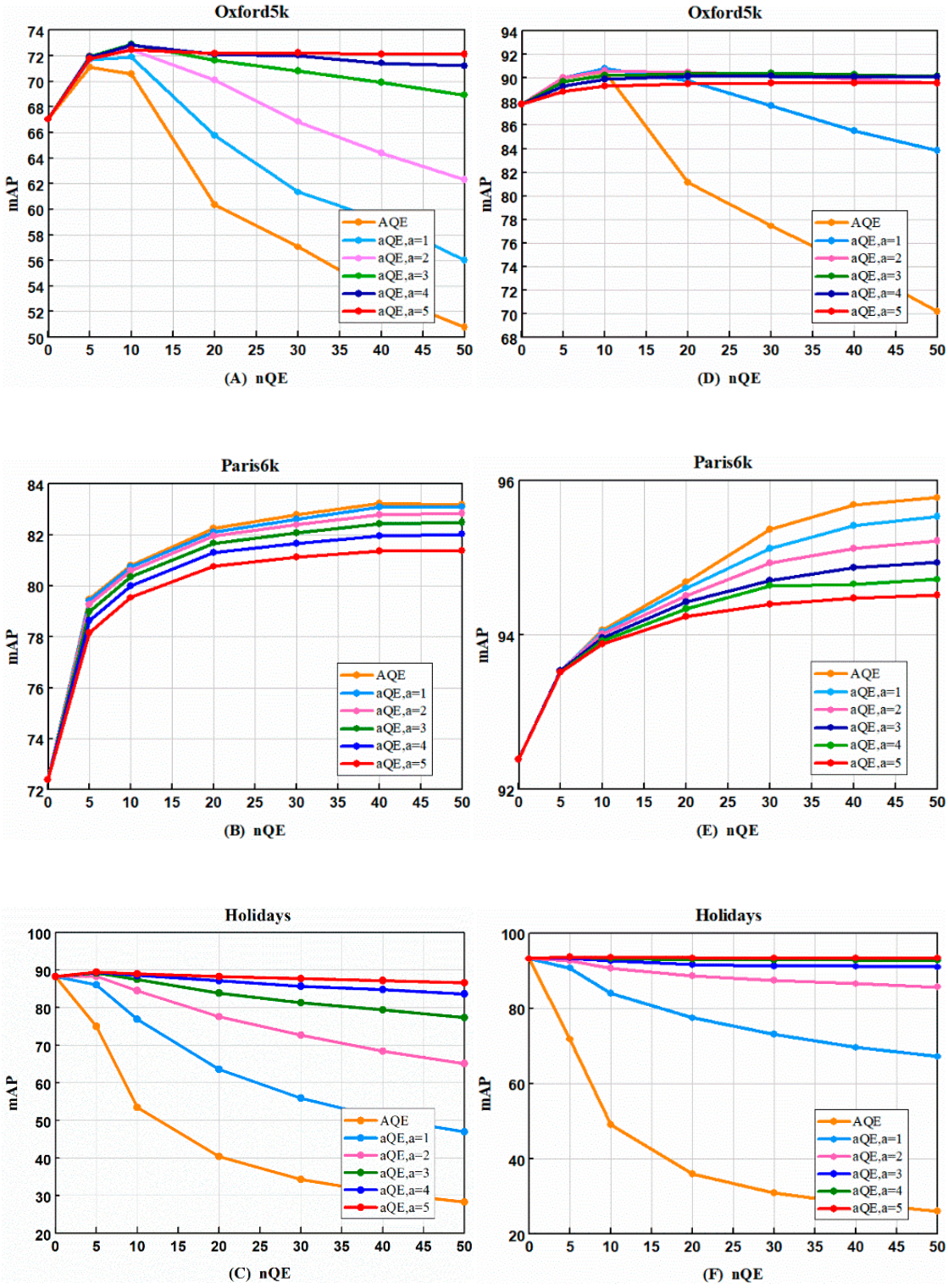

Figure 10. The evaluation of performance on α-weighted query expansion (αQE): (**A**) AlexNet with GeM layer, and Lw on Oxford5k; (**B**) AlexNet with GeM layer, and Lw on Paris6k; (**C**) AlexNet with GeM layer, and Lw on Holiday; (**D**) ResNet with GeM layer, and Lw on Oxford5k; (**E**) ResNet with GeM layer, and Lw on Paris6k; (**F**) ResNet with GeM layer, and Lw on Holiday; The standard average query expansion (AQE) is compared to our αQE for different values of α and the number of images used nQE.

4.6. Experimental Results and Comparison

4.6.1. Comparison with the State of the Art

To comprehensively evaluate the localization performance of our trained LRML model, we have extensively compared the results with the latest performance of compact image representation and methods for query expansion. The results of the fine-tuning network based on the multiple loss are summarized together with the results currently published in Table 5. When using deep networks to compact representation, the better performance on Paris is inherited by the nature of the pre-trained networks; the LRML model with ResNet achieves 87.9 on Oxford5k, 93.3 on Holiday and 93.8 on Paris6k. Using the architectures of VGG and ResNet has achieved most advanced scores on both Paris and Holiday. When using the architecture of ResNet to conduct initialization, the proposed method is superior to the state-of-the-art in all datasets. Notably, the result of GeM+VGG16 [27] on Oxford5k is 87.9, and 83.3 on Oxford105k, better than our results, that is, 83.2 on Oxford5k and 78.6 on Oxford105k. The reason is probably because in training datasets, the differences between the negative samples are limited, with fewer VGG16 network layers. Thus, it is difficult to extract effective features based on ranking sequence. However, when the framework of ResNet is adopted, with the datasets on Oxford5k and Oxford105k, our results are 87.9 and 84.8 respectively, which still outperform those on GeM [27], namely 87.8 and 84.6. When we use VGG19 as the experimental network, the results are better than those on VGG16, which proves one of our hypotheses, that is, the limitation on the number of layers of networks restricts feature extraction.

We have also evaluated how our model fares on the occasion of combining an updated query expansion method by Radenović. This method applies in addition to the current methods and uses information from the closest neighbors in the gallery, so as to improve the ranking results. As shown in Table 5, our proposed method performs well and demonstrates similar improvements to those achieved by Radenović [27]. After adding re-ranking and query expansion, the results on ResNet still achieve the best performance. Specifically, the final LRML model with ResNet achieves 92.0 on Oxford5k, 89.5 on Holiday and 96.7 on Paris6k. In accordance with the current results, the results of VGG16 on Oxford5k and Oxford105k are 90.0 and 86.9, and the results of VGG19 on Oxford5k and Oxford105k are 90.8 and 87.9, respectively, which are lower than the results of GeM+VGG [27], with 91.9 on Oxford5k and 89.6 on Oxford105k. But the counterpart results on Paris and Holiday are 94.2 and 92.1, surpassing the other methods under the same conditions.

Table 5 also indicates that our method is robust in terms of the size of irrelevant training data. We trained our method with different sizes of training data from the Flickr100k distractor images. The experiment was performed on the two large-scale datasets, i.e., Oxford 105k, Paris106k and Holiday101k. The experimental results indicate that the mAP score of our method is almost unchanged with the training size. The results indicate that our method is robust in terms of training size when this is learned from irrelevant data.

4.6.2. Visualization of Image Retrieval

We use the existing ResNet network and the already trained network that joins the multiple loss function to search images on Oxford. As shown in Figure 11, the first column is the query images, and the following two lines behind each query image introduce the Top-20 query results. The first line includes the image retrieval result by using off-the-shelf ResNet architecture, and the second line reflects the retrieval result of the trained network after leveraging multiple loss function. By comparing the first three groups of queries, it can be found that some error images appear in the image search of ResNet, which are circled in red. As observed from the two sets of queries, under the condition that all image search results are correct, ours are uniform and highly consistent with the similarity of the query images. In contrast, the effect of ResNet search is not that integrated. Overall, our model has significantly improved the retrieval performance.

Table 5. The comparison of performance (mAP) between our method and the most advanced image retrieval method under VGG16, VGG19 and ResNet (Res) deep network. F-tuned: 'Yes' means use fine-tuning network, 'no' means use off-the-shelf network, and 'n/a' means local method is not applicable. Dim: The final dimension of the image representation. Marked with * is our method, which is used in combination with learning whitening Lw and multi-scale representation.

Net	Method	F-tuned	Dim	Oxford5k	Oxford105k	Paris6k	Paris106k	Holidays	Hol101k
Compact representation using deep networks									
VGG	MAC [27] †	no	512	56.4	47.8	72.3	58.0	79.0	66.1
	SPoC [41] †	no	512	68.1	61.1	78.2	68.4	83.9	75.1
	CroW [22]	no	512	70.8	65.3	79.7	72.2	85.1	-
	R-MAC [21]	no	512	66.9	61.6	83.0	75.7	86.9	-
	BoW-CNN [23]	no	n/a	73.9	59.3	82.0	64.8	-	-
	NetVLAD [3]	no	4096	66.6	-	77.4	-	88.3	-
	NetVLAD [3]	yes	512	67.6	-	74.9	-	86.1	-
	NetVLAD [3]	yes	4096	71.6	-	79.7	-	87.5	-
	Fisher Vector [24]	yes	512	81.5	76.6	82.4	-	-	-
	R-MAC [31]	yes	512	83.1	78.6	87.1	79.7	89.1	-
	GeM [27]	yes	512	87.9	83.3	87.7	81.3	89.5	79.9
	*ours (VGG16)	yes	512	83.2	78.6	89.4	83.8	90.2	82.7
	*ours (VGG19)	yes	512	85.7	81.4	89.8	82.2	90.7	82.9
Res	R-MAC [21]	no	2048	69.4	63.7	85.2	77.8	91.3	-
	GeM [27]	yes	2048	87.8	84.6	92.7	86.9	93.3	87.9
	*ours	yes	2048	87.9	84.8	93.8	87.5	93.3	88.24
Re-ranking(R) and query expansion (QE)									
n/a	BoW+R+QE [25]	n/a	n/a	82.7	76.7	80.5	71.0	-	-
	BoW-fVocab+R+QE [42]	n/a	n/a	84.9	79.5	82.4	77.3	75.8	-
	HQE [26]	n/a	n/a	88.0	84.0	82.8	-	-	-
VGG	CroW+QE [22]	no	512	74.9	70.6	84.8	71.0	-	-
	R-MAC+R+QE [21]	no	512	77.3	73.2	86.5	79.8	-	-
	BoW-CNN+R+QE [23]	no	n/a	78.8	65.1	84.8	64.1	-	-
	R-MAC+QE [31]	yes	512	89.1	87.3	91.2	86.8	-	-
	GeM+αQE [27]	yes	512	91.9	89.6	91.9	87.6	-	-
	*ours (VGG16)	yes	512	90.0	86.9	94.2	89.9	91.2	82.8
	*ours (VGG19)	yes	512	90.8	87.4	94.6	90.5	91.6	83.3
Res	R-MAC+QE [21] ‡	no	2048	78.9	75.5	89.7	85.3	-	-
	R-MAC+QE [43]	yes	2048	90.6	89.4	96.0	93.2	-	-
	GeM+αQE [27]	yes	2048	91.0	89.5	95.5	91.9	-	-
	*ours	yes	2048	92.0	90.5	96.7	93.8	93.7	89.5

†: The results of our evaluation of SPoC and mac using PCAw and off-the-shelf networks. ‡: Results of evaluating R-MAC using [39] and off-the-shelf networks

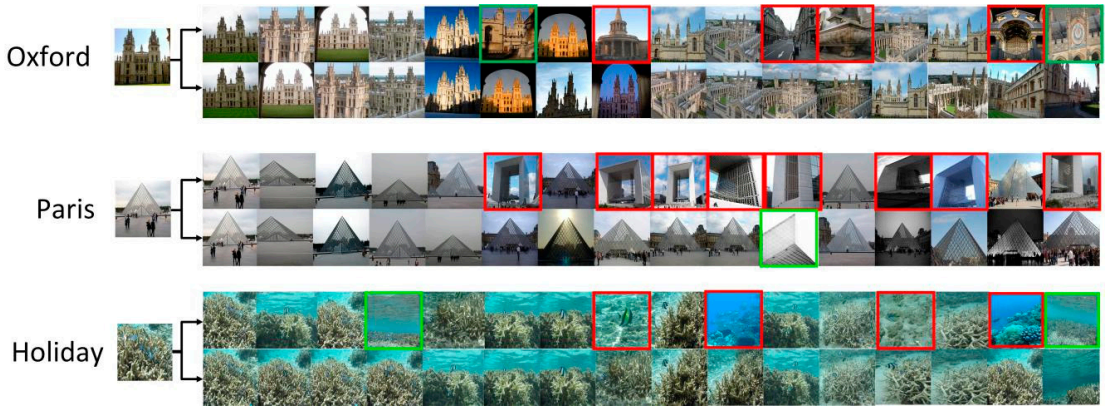

Figure 11. Several query examples with the best retrieval results. Top (off-the-shelf) rows are examples of the bad ones while bottom (ours) rows are the good results the finetuning of Resnet101. Note, a red bounding box marks non-relevant images and a green bounding box marks not-perfect image. See text for more detail.

5. Conclusions

In this paper, we proposed a new metric learning loss called multiple loss based on L2R. For contrastive loss, triple loss, and quadruple loss, we can use multiple examples simultaneously to improve the robustness of the model. In our method, we calculate a distance sequence, choose the maximum positive distance and the different negative pairwise distances after adjusting the threshold to calculate the final loss. In this way, the ranking-based multiple loss trains the network using a dissimilar positive image and multiple negative images, selected from different clusters, thus sharing different similarities with the query image. We used VGG16 and ResNet101 as the base model to perform some comparative experiments with different metric losses.

The comparison shows that our multiple loss network based on L2R has achieved the top performance. We have then compared our approach to some of the most advanced methods available today. On several benchmark datasets, including Oxford, Paris, and Holiday, our approach shows better performance than those methods.

In the future, we will investigate other ways to improve the performance of images retrieved, such as increasing the number of negative samples for better feature extraction. Furthermore, we will expand our work by building up a more effective network architecture where a positive sample sorting sequence is introduced to improve multiple loss. This makes the proposed architecture obtain robustness in image processing, and also extracts some high-level cues, so as to accurately retrieve images.

Author Contributions: All the authors contributed to this study. Conceptualization, Lili Fan and Pingping Liu; Methodology, Lili Fan and Pingping Liu; Software, Haoyu Zhao and Huangshui Hu; Writing, Lili Fan; Writing—review, Pingping Liu and Hongwei Zhao.

Funding: This research was funded by 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, and the State Key Laboratory of Applied Optics Open Fund Project, grant number 20173660.

Acknowledgments: We would like to thank Wei Wang for his suggestions to the language editing.

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

References

1. Gong, Y.C.; Wang, L.W.; Guo, R.Q.; Lazebnik, S. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision*; Part VII; Springer: Cham, Switzerland, Volume 8695; pp. 392–407.
2. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1717–1724.
3. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1437–1451. [CrossRef] [PubMed]
4. Hershey, J.R.; Chen, Z.; Le Roux, J.; Watanabe, S. Deep clustering: Discriminative embeddings for segmentation and separation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 20–25 March 2016; pp. 31–35.
5. 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.
6. Cui, Y.; Zhou, F.; Lin, Y.; Belongie, S. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1153–1162.
7. Chen, W.H.; Chen, X.T.; Zhang, J.G.; Huang, K.Q. Beyond triplet loss: A deep quadruplet network for person re-identification. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (Cvpr 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 1320–1329.

8. Hermans, A.; Beyer, L.; Leibe, B. In Defense of the Triplet Loss for Person Re-Identification. *arXiv* **2017**, arXiv:1703.07737.
9. Xiao, Q.; Luo, H.; Zhang, C. Margin Sample Mining Loss: A Deep Learning Based Method for Person Re-identification. *arXiv* **2017**, arXiv:1710.00478.
10. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In Proceedings of the Advances in Neural Information Processing Systems 30 (Nips 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
11. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Object retrieval with large vocabularies and fast spatial matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; Volume 1–8, pp. 1545–1588.
12. Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; Zisserman, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AL, USA, 24–26 June 2008; Volume 1–12, pp. 2285–2298.
13. Jegou, H.; Douze, M.; Schmid, C. Hamming embedding and weak geometric consistency for large scale image search. I. Proceedings of 10th European Conference on Computer Vision, ECCV 2008, Marseille, France, 12–18 October 2008; Volume 5302, pp. 304–317.
14. Babenko, A.; Slesarev, A.; Chigorin, A.; Lempitsky, V. Neural codes for image retrieval. In Proceedings of the Computer Vision—Eccv 2014, Zurich, Switzerland, 6–12 September 2014; Volume 8689, pp. 584–599.
15. Husain, S.S.; Bober, M. REMAP: Multi-layer entropy-guided pooling of dense CNN features for image retrieval. *IEEE Trans. Image Proc.* **2019**. [CrossRef] [PubMed]
16. Perronnin, F.; Liu, Y.; Sanchez, J.; Poirier, H. Large-scale image retrieval with compressed fisher vectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), San Francisco, CA, USA, 13–18 June 2010; pp. 3384–3391.
17. Jegou, H.; Perronnin, F.; Douze, M.; Sanchez, J.; Perez, P.; Schmid, C. Aggregating Local Image Descriptors into Compact Codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1704–1716. [CrossRef] [PubMed]
18. Radenovic, F.; Jegou, H.; Chum, O. Multiple measurements and joint dimensionality reduction for large scale image search with short vectors. In Proceedings of the ICMR'15: Proceedings of the 2015 Acm International Conference on Multimedia Retrieval, Shanghai, China, 23–26 June 2015; pp. 587–590.
19. Arandjelovic, R.; Zisserman, A. All about VLAD. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Portland, OR, USA, 23–28 June 2013; pp. 1578–1585.
20. Tolias, G.; Furon, T.; Jegou, H. Orientation Covariant Aggregation of Local Descriptors with Embeddings. VI. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Volume 8694, pp. 382–397.
21. Tolias, G.; Sicre, R. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
22. Kalantidis, Y.; Mellina, C.; Osindero, S. Cross-Dimensional Weighting for Aggregated Deep Convolutional Features. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 685–701.
23. Mohedano, E.; McGuinness, K.; O'Connor, N.E.; Salvador, A.; Marques, F.; Giro-I-Nieto, X. Bags of Local Convolutional Features for Scalable Instance Search. In Proceedings of the ICMR'16: Proceedings of the 2016 Acm International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; pp. 327–331.
24. Ong, E.J.; Husain, S.; Bober, M. Siamese Network of Deep Fisher-Vector Descriptors for Image Retrieval. *arXiv* **2017**, arXiv:1702.00338.
25. Chum, O.; Mikulik, A.; Perdoch, M.; Matas, J. Total recall II: Query expansion revisited. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr), Colorado Springs, CO, USA, 20–25 June 2011; pp. 889–896.
26. Tolias, G.; Jegou, H. Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recognit.* **2014**, *47*, 3466–3476. [CrossRef]
27. 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]
28. Shen, X.; Lin, Z.; Brandt, J.; Wu, Y. Spatially-constrained similarity measurefor large-scale object retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1229–1241. [CrossRef] [PubMed]

29. Avrithis, Y.; Kalantidis, Y. Approximate gaussian mixtures for large scale vocabularies.computer vision. III. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Volume 7574, pp. 15–28.
30. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.H.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
31. Gordo, A.; Almazan, J.; Revaud, J.; Larlus, D. Deep image retrieval: learning global representations for image search. VI. In Proceedings of the Computer Vision—Eccv 2016, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9910, pp. 241–257.
32. Chopra, S.; Hadsell, R.; Lecun, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition (CVPR), Toronto, ON, Canada, 20 June 2005; pp. 539–546.
33. Hadsell, R.; Chopra, S.; Lecun, Y. Dimensionality Reduction by Learning an Invariant Mapping. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition (CVPR), New York, NY, USA, 17 June 2006; Volume 2, pp. 1735–1742.
34. Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wu, Y. Learning Fine-Grained Image Similarity with Deep Ranking. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 1386–1393.
35. Schönberger, J.L.; Radenović, F.; Chum, O.; Frahm, J.M. From single image query to detailed 3d reconstruction. In Proceedings of the Computer Vision & Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
36. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN Features off-the-shelf: An astounding baseline for recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 23–28 June 2014; pp. 512–519.
37. Babenko, A.; Lempitsky, V. Aggregating Deep Convolutional Features for Image Retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1269–1277.
38. Bell, S.; Bala, K. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph.* **2015**, *34*, 98. [CrossRef]
39. Song, H.O.; Xiang, Y.; Jegelka, S.; Savarese, S. Deep metric learning via lifted structured feature embedding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4004–4012.
40. Jegou, H.; Chum, O. Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening. Pt II. In Proceedings of the Computer Vision—ECCV 2012, Florence, Italy, 7–13 October 2012; Volume 7573, pp. 774–787.
41. Imbriaco, R.; Sebastian, C.; Bondarev, E. Aggregated Deep Local Features for Remote Sensing Image Retrieval. *Remote Sens.* **2019**, *11*, 493. [CrossRef]
42. Mikulik, A.; Perdoch, M.; Chum, O.; Matas, J. Learning Vocabularies over a Fine Quantization. *Int. J. Comput. Vis.* **2013**, *103*, 163–175. [CrossRef]
43. Gordo, A.; Almazán, J.; Revaud, J.; Larlus, D. End-to-End Learning of Deep Visual Representations for Image Retrieval. *Int. J. Comput. Vis.* **2017**, *124*, 237–254. [CrossRef]