algorithms

MDPI

Article

Deep Feature Learning with Manifold Embedding for Robust Image Retrieval

Xin Chen [1] and Ying Li [2,*]

[1] College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China; 1410452@tongji.edu.cn

[2] School of Computer and Electronic Information, Nanjing Normal University, Nanjing 210023, China

* Correspondence: ying.li@njnu.edu.cn

Received: 16 November 2020; Accepted: 30 November 2020; Published: 2 December 2020

Abstract: Conventionally, the similarity between two images is measured by the easy-calculating Euclidean distance between their corresponding image feature representations for image retrieval. However, this kind of direct similarity measurement ignores the local geometry structure of the intrinsic data manifold, which is not discriminative enough for robust image retrieval. Some works have proposed to tackle this problem by re-ranking with manifold learning. While benefiting better performance, algorithms of this category suffer from non-trivial computational complexity, which is unfavorable for its application to large-scale retrieval tasks. To address the above problems, in this paper, we propose to learn a robust feature embedding with the guidance of manifold relationships. Specifically, the manifold relationship is used to guide the automatic selection of training image pairs. A fine-tuning network with those selected image pairs transfers such manifold relationships into the fine-tuned feature embedding. With the fine-tuned feature embedding, the Euclidean distance can be directly used to measure the pairwise similarity between images, where the manifold structure is implicitly embedded. Thus, we maintain both the efficiency of Euclidean distance-based similarity measurement and the effectiveness of manifold information in the new feature embedding. Extensive experiments on three benchmark datasets demonstrate the robustness of our proposed method, where our approach significantly outperforms the baselines and exceeds or is comparable to the state-of-the-art methods.

Keywords: image retrieval; deep feature learning; similarity measurement; manifold embedding

1. Introduction

Feature representation and similarity measurement are two critical components in content-based image retrieval (CBIR) [1,2]. Conventionally, the similarity between images is measured by the Euclidean distance between their corresponding features. A retrieval system ranks candidate images according to such similarity. However, Zhou et al. have revealed that a traditional pairwise Euclidean distance is not adequate to demonstrate the intrinsic similarity relationship between images [3].

Many algorithms [4–12] have been proposed to model the geometry structure of the intrinsic data manifold. Among these methods, a graph-based affinity learning algorithm called the diffusion process [13] has shown superior ability, which learns the local structure of data manifold for re-ranking in image retrieval. Nevertheless, the diffusion-based re-ranking method usually incurs extra computational overhead and time expenses. The online retrieval stage of a CBIR system usually requires real-time efficiency, and thus such time and computational resource overhead are unfavorable. On the other hand, the offline database image indexing stage has a relatively lower efficiency requirement. This gives us a hint to embed such geometry structure information of the intrinsic data manifold into image feature representations.

Traditional learning methods usually focus on carefully-designed hand-crafted image features. Global features such as HSV [14] are directly used for image representation. Local features like SIFT [15] are firstly aggregated globally and then used for image representation [16]. Although those kinds of feature representations have achieved certain benefits, their rigid processing architectures limit further improvement for the image retrieval task and leave little room to embed the learned manifold information into such feature representations. Leading by the brilliant work of Krizhevsky et al. [17], the potential of deep learning has been widely explored in the computer vision community. Convolutional Neural Networks (CNNs) demonstrate a powerful representative capability for images and have pushed the performance of a considerable amount of vision tasks to a new state-of-the-art [18], including image retrieval [19].

Most existing works directly took off-the-shelf CNN models pre-trained on classification tasks as the feature extractor for image retrieval [20,21]. However, there is a natural gap between the tasks of image classification and image retrieval, since image classification focuses on class-level discrimination while image retrieval emphasizes instance-level similarity more. Directly using the pre-trained CNN models on image retrieval will result in limited performance. Moreover, it discards the trainable nature of CNN, which is another essential character that leads CNN to success. As a consequence, it is a natural choice to fine-tune the pre-trained CNN model to learn more powerful representations to fit the requirements of image retrieval tasks [22,23].

Previous works [24–29] have demonstrated that Siamese or triplet networks are more proper to learn for ranking-oriented retrieval task with pairwise or triplet similarity supervision. For network training, the quality of supervision is one of the critical factors that affect the quality of feature learning and information embedding. Unfortunately, dataset collecting and label annotating are labor-consuming work, and thus large-scale hand-collected and labeled training data are not a feasible option. Although some researchers have proposed to automatically generate training image pairs or triplets [22,23], a big pre-collected image pool with a strict constraint on image types is still required.

To address the above problems, in this paper, we propose to automatically select training image pairs with the help of geometry structure information of the intrinsic data manifold and embed such information into the feature learning process with a specially designed Siamese network. On the one hand, similarity based on the Euclidean distance is efficient to calculate but usually lacks robustness. On the other hand, local geometry information of the intrinsic data manifold is very effective in improving the reliability of similarity measurement but is not easy to acquire. My approach combines those merits. With the supervision of manifold learning, the automatic image pairs selection process equips selected pairs with such manifold information. Fine-tuning the pre-trained model with those image pairs transfers local geometry information of the intrinsic data manifold into the newly learned feature embedding. Under the newly learned feature embedding, Euclidean distance-based similarity measurement is not only efficient to calculate but is also robust.

An overview of the proposed method is illustrated in Figure 1. The pipeline starts at an old feature embedding and ends up with a new feature embedding. Taking the previous endpoint as a new start point, we can restart the automatic image pairs selection and network fine-tuning process, and thus we can iteratively improve the feature embedding to get better feature representations.

Since our goal is to learn a robust feature embedding with pairwise supervision, with the learned embedding, the output feature representations of the same image pair should be close in the Euclidean space. A similarity embedding loss is adopted to pull those image pairs together. For those not similar in both the original Euclidean space and the learned manifold, we should keep their relationship. We use a feature consistency preserving loss to prevent dramatic change to the new feature embedding. Experimental results are included to demonstrate the effectiveness of our proposed method. Specifically, the proposed method significantly outperforms the baseline and surpasses or is on par with the state-of-the-art methods on three benchmark datasets.

The rest of this paper is organized as follows: Section 2 reviews some closely related work of this paper. In Sections 3–5, the details of the proposed method are presented. Experiments are discussed in Section 6, followed by conclusions in Section 7.

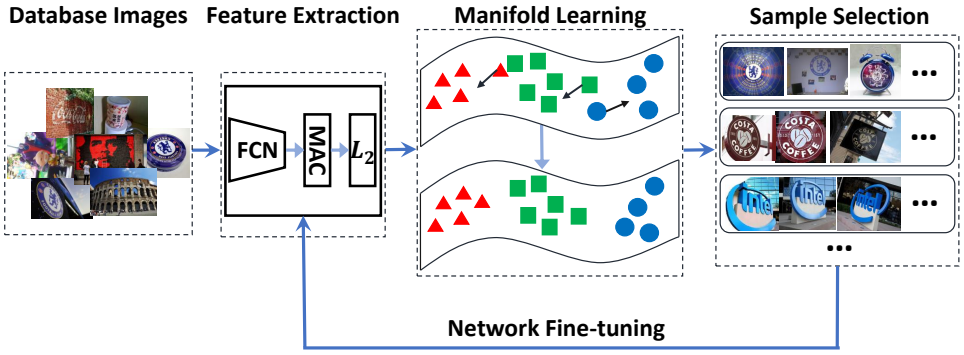

Figure 1. The pipeline of the proposed method. The pipeline consists of two stages: the self-supervised training example selection stage and the network fine-tuning stage. In the training example selection part, database images are firstly fed into the CNN model to extract MAC feature representations. The similarity relationship between images is generated from MAC features and refined by manifold learning. Finally, top-ranked image pairs are selected to supervise network fine-tuning. The above two stages can be conducted iteratively until convergence.

2. Related Work

My work is related to deep learning-based image retrieval and manifold learning for visual re-ranking. In the following, we briefly discuss those works and point out the differences between our work and theirs.

Deep learning-based image retrieval. Witnessing the great success of deep learning on a variety of vision tasks, some pioneering works started leveraging CNN on image retrieval tasks [19,21,30]. Most of these works were based on off-the-shelf neuron activations of pre-trained CNNs. Some of the early works directly utilized the activation of fully-connected (FC) layers of the network as image representation [31]. Razavian et al. proposed to leverage the activations of augmented FC6 layer of AlexNet [17] as image representation. The reported performance outperformed the state-of-the-art SIFT-based methods even with such a rough setting, which demonstrated the powerful representative capability of CNNs. Successive researchers realized that image representations generated from convolutional layers are more suitable for image retrieval. Ng et al. leveraged VLAD [32] to encode column features on each spatial location of feature maps [21]. Gong et al. also used VLAD to aggregate feature maps extracted from local image patches across multiple scales [30]. Babenko et al. proposed to aggregate convolutional feature maps into a compact image representation by sum-pooling [20]. Tolias et al. applied max-pooling over multiple carefully-designed regions and integrated those generated vectors into a single feature vector [33]. In addition, some works tried to combine these two types of features. Li et al. fused convolutional features and FC features to compose a compact image representation [34].

While off-the-shelf models have achieved impressive results on retrieval performance, several works have proved that fine-tuning the pre-trained CNNs is a promising branch for image retrieval. Babenko et al. re-trained existing pre-trained models with a dataset related to buildings and demonstrated the feasibility and effectiveness of fine-tuning [19]. However, this method kept classification-based network architecture, which limited the performance from further promotion.

Fine-tuning pre-trained models with a retrieval oriented objective and dataset drives the learned feature more suitable for pairwise similarity measurement. Arandjelovic et al. inserted a back-propagatable VLAD-like layer into a pre-trained model and fine-tuned the model with triplet loss [35]. The notable point is that this work used only weak supervision. Radenović et al. [23] went one step further to train a retrieval oriented model with a Siamese network in an unsupervised manner. They employed the Structure-from-Motion method and the SIFT-based Bag-of-Word (BoW) method to group images of the same architecture together. The training positive and negative image samples

were automatically selected based on those image groups. Gordo et al. applied a similar pipeline but with triplet loss and a SIFT matching based label generation method instead [22]. Although these works achieved considerable improvement in image retrieval, they all needed a large stand-alone image dataset related to the target dataset, which is challenging to collect.

Different from them, our approach automatically selects relevant training image pairs with learned local geometry information of the intrinsic data manifold, and our goal is to embed such learned manifold information into a new feature embedding. In addition, we have a specially designed loss function.

Manifold learning for visual re-ranking. In image retrieval tasks, manifold learning-based methods are usually applied to refine pairwise image similarity since the original Euclidean distance-based similarity measurement is not reliable enough. Such similarity refinement is especially useful for re-ranking. Bai et al. proposed to do visual re-ranking by collecting the intrinsic manifold information of data with an algorithm called Sparse Contextual Activation (SCA) [36]. Several works demonstrated that the diffusion process was a promising way to do re-ranking in retrieval tasks. The diffusion process uses a graph to represent the similarity relationship of images, where vertices represent images and edges between vertices denote their corresponding similarity. The manifold structure is learned and applied by iteratively diffusing similarity into a part of or the whole graph [13]. Many works showed excellent performance on image retrieval [37–41]. In this work, we leverage manifold learning methods to refine the initial Euclidean-based similarity measurement. While one of the critical points of the proposed method lies in manifold learning, we do not pay extra attention to studying how to improve it. We adopt the state-of-the-art method and embed it into the pipeline to select training image pairs as supervision.

There are also a few works learning to embed from manifold information for a variety of tasks. Xu et al. iteratively learned the manifold embedding via the iterative manifold embedding (IME) layer [42]. However, this work only focuses on the projection from initial feature representation to the iterative manifold embedding representation, which limits the capability to handle unseen images. Iscen et al. [43] proposed to train models with image pairs mined with the help of manifold learning. Compared to this work, we have a different objective and do not need a large stand-alone dataset related to the target dataset.

3. Formulation and Pipeline

3.1. Formulation

Given a dataset with n images: $\mathbf{I} = \{\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \cdots, \mathbf{I}^{(n)}\}$, we define a mapping function $f(\cdot, \theta) : \mathbb{R}^{H_I \times W_I \times c} \rightarrow \mathbb{R}^K$ to embed an image of size $H_I \times W_I$ with c channels into a K-dimensional feature vector, where θ is a set of trainable parameters of the mapping function. In this work, the mapping function is a fully-convolutional CNN concatenated with a max-pooling layer and an l_2-normalization layer. The output of this mapping function is the MAC (Maximum Activations of Convolutions) image representation [33]. With the mapping function, the dataset can be represented as $\mathbf{F} = \{\mathbf{f}^{(1)}, \mathbf{f}^{(2)}, \ldots, \mathbf{f}^{(n)}\}$, where $\mathbf{f}^{(i)}$ is the feature representation of image $\mathbf{I}^{(i)}$. Traditionally, the similarity between images is measured by feature distances in Euclidean space. However, as stated in Section 1, this kind of similarity relationship is not reliable enough. An alternative is to improve it by re-ranking the similarity relationship between images with manifold learning methods. Since online re-ranking is time-consuming, our goal is to improve the feature embedding with manifold information learned from offline "re-ranking". This is achieved by fine-tuning a new feature embedding or mapping function with training examples selected automatically with learned manifold information in an end-to-end manner.

3.2. Pipeline

The pipeline of our work consists of two stages, i.e., training sample selection and network fine-tuning. Each image in the database is firstly passed through the original network to extract MAC

features. The original similarity relationship is generated by computing pairwise feature distance in Euclidean space. Then, it is refined with a kind of re-ranking method based on manifold learning, resulting in a new similarity or affinity matrix. Finally, we can choose top-ranked similar image pairs according to the new similarity or affinity matrix. In the network fine-tuning stage, we feed those pre-mined similar image pairs into a Siamese network and fine-tune the CNN parameters with a carefully designed loss. With the fine-tuned CNN model, we can restart the training sample selection again, which means that the above two stages can work iteratively until convergence. An overview of the pipeline is demonstrated in Figure 1.

4. Self-Supervised Training Example Selection

4.1. MAC Image Representation

Previous work [23] has proved that MAC image representation is effective and efficient for image retrieval. The MAC image representation is based on fully-convolutional CNNs. In our implementation, we leverage convolutional layers of general image recognition networks, such as VGGNet [44] and ResNet [45].

Specifically, given an input image $\mathbf{I}^{(i)}$, the output of the fully-convolutional CNN is a tensor consisting of K feature maps of shape $W \times H$. For the j-th feature map $\mathbf{M}_j^{(i)}$, we extract the maximum value among it:

$$f_j^{(i)} = \max_{x \subset \mathbf{M}_j^{(i)}} x. \tag{1}$$

Applying Equation (1) to every feature map of the output tensor, we have a K-dimensional vector. After necessary l_2-normalization, we obtain a feature vector $\mathbf{f}^{(i)} = [f_1^{(i)}, f_2^{(i)}, \cdots, f_K^{(i)}]^{\mathrm{T}}$ to represent image $\mathbf{I}^{(i)}$, i.e., the MAC.

4.2. Refining Distance Metric with Manifold Learning

By calculating the pairwise Euclidean distance between those MACs, we get a distance matrix $\mathbf{D}$, where D_{ij} denotes the Euclidean distance between MACs of the image $\mathbf{I}^{(i)}$ and $\mathbf{I}^{(j)}$. If we regard image $\mathbf{I}^{(i)}$ as the query and the rest images in the dataset as database images, then the i-th row of $\mathbf{D}$ shows the dissimilarity between query-database image pairs. Sorted in ascending order, it can be taken as the retrieval result of query $\mathbf{I}^{(i)}$.

Manifold learning-based re-ranking methods can refine the original similarity relationship. In this work, we mainly choose the Regularized Diffusion Process (RDP) [40], a state-of-the-art scheme, as the re-ranking algorithm. RDP takes a distance matrix with the query as input and learns the local geometry structure of the intrinsic manifold of the input feature space by a choreographed graph-based diffusion process. The output of RDP is an affinity matrix $\mathbf{A}$, where A_{ij} denotes refined similarity between image $\mathbf{I}^{(i)}$ and $\mathbf{I}^{(j)}$. The re-ranking module in our approach is not limited to the adopted RDP scheme and can be replaced with any re-ranking method with high efficiency and effectiveness.

4.3. Training Pairs Selection

The previously calculated affinity matrix provides refined pairwise similarity between images. Selecting image $\mathbf{I}^{(i)}$ as an anchor image, we can put its k-nearest neighbor set $N(\mathbf{I}^{(i)}, k)$ into a similar image pool according to the i-th row of $\mathbf{A}$, where k is the number of nearest neighbors. However, Ref. [46] demonstrated that it is not sufficiently reliable to determine actual neighbors due to the asymmetry of k-nearest neighbor. The k-reciprocal nearest neighbor is more reliable to uncover the potential genuine neighbors for the retrieval task [46,47]. As a consequence, we choose the k-reciprocal nearest neighbor set $R(\mathbf{I}^{(i)}, k)$ of $\mathbf{I}^{(i)}$ to be the similar image pool. $R(\mathbf{I}^{(i)}, k)$ is defined as follows:

$$R(\mathbf{I}^{(i)}, k) = \{\mathbf{I}^{(j)} \mid \mathbf{I}^{(j)} \in N(\mathbf{I}^{(i)}, k), \mathbf{I}^{(i)} \in N(\mathbf{I}^{(j)}, k)\}, \tag{2}$$

which means that image $\mathbf{I}^{(j)}$ is a k-reciprocal nearest neighbor of $\mathbf{I}^{(i)}$ only when they are in the k-nearest neighbor set of each other. With the new similarity and reciprocal relationship, image pairs can be mined for the fine-tuning process. By taking every image in the dataset as the anchor image and applying the training pairs selection process, we can generate our training examples $T = \{(\mathbf{I}^{(i)}, \mathbf{I}^{(j)}) \mid \mathbf{I}^{(j)} \in R(\mathbf{I}^{(i)}, k)\}$.

5. Siamese Feature Learning

5.1. Network Structure

We adopt a Siamese network to fine-tune network parameters. Our goal is to embed the learned similarity into the new feature embedding, and we apply a variant of the Siamese network with four inputs. The first two channels are the image pairs, and the other two channels are their corresponding MAC features extracted by the pre-trained model. Image pairs are firstly fed into two fully-convolutional networks (FCNs) separately, which have identical network architecture and shared parameters. In this work, the FCNs are popular CNN architectures trained for image classification on ImageNet [48] (e.g., VGGNet and ResNet), with fully-connected layer and global average pooling layer discarded. The pre-trained parameters are adopted to initialize the networks. The outputs of the FCNs are non-negative because of the ReLU layer before it. Each FCN is followed by a MAC layer, which generates compact MAC representations for input images. After an l_2-normalization layer, the MAC vector is normalized into a unit vector. For the other two input channels, two normalized MACs are extracted with the pre-trained model.

A sketchy network structure is illustrated in Figure 2. Suppose that image $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$ compose a similar image pair. They are fed into the identical FCNs to get MACs $\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$. Meanwhile, the corresponding pre-extracted initial MACs $\widetilde{\mathbf{f}}^{(1)}$ and $\widetilde{\mathbf{f}}^{(2)}$ are directly sent to the loss part. All MACs are l_2 normalized before calculating the loss.

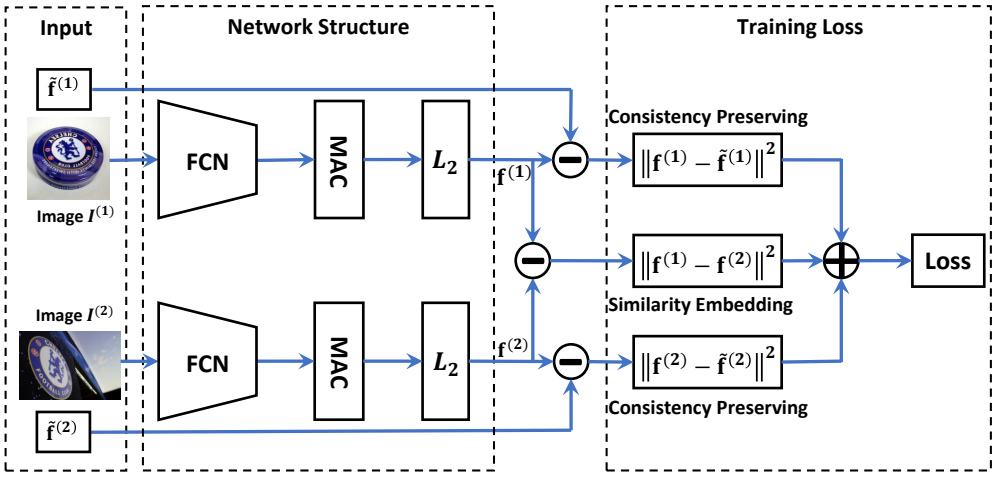

Figure 2. The network architecture of the proposed network. A pair of images are fed into two fully-convolutional networks (FCNs) separately to generate MAC representations. These two FCNs are identical in architecture and share parameters. The generated MACs and their corresponding pre-extracted MACs (with original network parameters) are then sent to calculate the loss. The loss consists of a similarity embedding part to embed manifold information into network parameters and a consistency preserving part to prevent dramatic modification of model parameters.

5.2. Objective and Optimization

The objective function consists of two parts, i.e., the similarity embedding part and the feature consistency preserving part. The similarity embedding part constrains the network from needing

to embed similarity information carried by input image pairs so that the learned MAC feature distance of the same pair should be small in Euclidean space. With this part, manifold information is transferred into fine-tuned feature embedding. The feature consistency preserving part constrains the learned feature embedding from changing the original feature embedding dramatically, preventing the similarity context of unpaired images from corruption. Without this part, the fine-tuning process will be devoted to fitting those similar image pairs. In addition, this is one of the reasons that we do not strictly need negative pairs in this work, which will be explained in detail later.

The loss function for input image pair $(\mathbf{I}^{(1)}, \mathbf{I}^{(2)})$ is defined as follows:

$$\mathcal{L}(\mathbf{I}^{(1)}, \mathbf{I}^{(2)}) = \| \mathbf{f}^{(1)} - \mathbf{f}^{(2)} \|^2 + \beta \sum_{i=1}^{2} \| \mathbf{f}^{(i)} - \widetilde{\mathbf{f}}^{(i)} \|^2, \tag{3}$$

where β is a balance factor, $\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$ are MACs generated by those two FCNs and $\widetilde{\mathbf{f}}^{(1)}$ and $\widetilde{\mathbf{f}}^{(2)}$ are pre-extracted MACs for image $\mathbf{I}^{(1)}$ and $\mathbf{I}^{(2)}$, respectively. Minimizing the first term means smaller Euclidean distance in the newly learned embedding space for selected image pairs. However, this will make the second term, i.e., the Euclidean distance of features between the newly learned embedding and the original embedding space enlarged. A proper balance factor can ensure efficient similarity embedding, while the correct similarity relationship of the original feature embedding is maximally kept.

Note that, in training, we do not involve negative image pairs. In this work, our goal is to embed similarity information, especially nearest neighbor information learned by manifold learning algorithms, into a new feature embedding. Since such information is carried by similar pairs or positive pairs, there is no need to train our model with additional negative pairs. In addition, the feature consistency preserving part of the learning objective ensures that the fine-tuning process does not modify the original model dramatically. As a consequence, the similarity relationships of negative pairs are not corrupted after fine-tuning because they are embedded in the model as parameters. For the above two reasons, negative pairs are not strictly needed in our proposed method.

5.3. Iteratively Mining and Learning

After running the process of distance matrix calculation and refining, training image pairs selection, and Siamese feature learning step by step, we harvest an improved feature embedding. The improved feature embedding is the output of the above process, and can also be the input of another such process, in other words, a new initial point. Based on this, we can extract new image features for database images, calculate a new distance matrix and a refined affinity matrix, mine new manifold information and training pairs, and learn a new feature embedding. These facts indicate that the proposed method allows us to iteratively improve the feature embedding with the CNN-based feature embedding module until all of the manifold context is inherited. A flowchart of this process is illustrated in Figure 1, where a loop containing such a process is composed.

6. Experiments

6.1. Experimental Setup

6.1.1. Datasets

We evaluate our proposed method on three public benchmark datasets, i.e., the UKBench dataset [49], the INRIA Holidays dataset [50], and the INSTRE dataset [51]. The UKBench dataset consists of 10,200 images, divided into 2550 groups. Each group contains four images, which represent one unique scene or object under various viewpoints, illumination conditions, scale change, etc. Every image is taken as the query in turn, and the corresponding four images belonging to the same category constitute the ground truth. The retrieval accuracy is measured by the N-S score, the average number of relevant images in the top 4 returned images. The Holidays dataset contains 1491 images collected from personal albums. These images belong to 500 groups, and one image is taken as the query in each group. Mean Average Precision (mAP) is adopted to measure the retrieval accuracy.

The INSTRE dataset consists of 28,543 images from 250 different object classes, which vary from everyday objects to famous logos under different conditions like scale change, rotation, occlusion, etc. In particular, 100 classes are retrieved from online sources; 150 classes are taken by the dataset creators. In the 150 classes, 100 are single-object classes, and the remaining 50 are two-object classes with a pairwise combination of the first 100 classes. Following the protocol from the official website, 1250 images are chosen to be queries, with five per class, and the remaining 27,293 are database images. We only use bounding boxes of queries in this work. We also use mAP to measure the retrieval accuracy.

6.1.2. Implementation Details

During the selection of training image pairs, we adopt RDP to refine the similarity for its tremendous effectiveness. It should be mentioned that other similarity learning algorithms can also be used here. Similarity refinement is performed with publicly released code by the author of [52]. All parameters about RDP are set following the original paper [52].

Network fine-tuning is conducted with the open-source deep learning platform TensorFlow [53]. Convolutional layers of VGGNet-16 (VGGNet-16 denotes 16-layer VGGNet [44] model) and ResNet-50 (ResNet-50 denotes 50-layer ResNet [45] model) and their corresponding ReLU, pooling, and/or batch normalization layers are selected to be the backbone of the Siamese network. Considering limited computational resources, we only fine-tune three conv5 layers of VGGNet-16, while all convolutional layers of ResNet-50 are fine-tuned. In the training stage, images are resized to 352×352 and then fed into the network. We adopt a Stochastic Gradient Descent (SGD) optimizer with a momentum of $\mu = 0.9$ and a weight decay of $\lambda = 5 \times 10^{-4}$ to tune network parameters. The learning rate is kept to be 0.001 and 0.01 for VGGNet-16 and ResNet-50, respectively, which are one-tenth of the initial learning rate of the pre-trained models. Learning rate decay is not adopted in this work because no evident performance gain is observed. The batch size is set to 8 for both VGGNet-16 and ResNet-50 due to the limitation of GPU memory. Previous works have demonstrated that properly using large images and maintaining the aspect ratio yield better performance in image retrieval tasks [54]. For feature extraction, we directly feed images into the network without resizing (For the Holidays dataset, all images are resized in advance so that the long side is 1024 and the aspect ratio maintained). All of our models are trained on a single Nvidia K40 GPU with 12 GB memory.

With different dataset sizes and backbones, the convergence speed varies. For VGGNet-16, the training is done for at most 20 epochs, which takes approximately 10, 2, and 20 h for UKBench, Holidays, and INSTRE, respectively. For ResNet-50, it has a faster convergence speed with at most 15 epochs of training, and the training time is 4, 1, and, 8 h, respectively. It is notable that, due to the variant capability of manifold learning algorithm on different datasets, we perform two iterations of training pairs mining and network fine-tuning for both UKBench and INSTRE and only one iteration on Holidays. Since output features are l_2 normalized, we use the cosine distance to measure image similarity for online retrieval.

6.2. Impact of Parameters

6.2.1. Balance Factor β

β is a key parameter to balance the similarity embedding part and the feature consistency preserving part of the objective function. We study the impact of β on retrieval accuracy and convergence speed, which is represented by the number of epochs. The UKBench is used, and the result is shown in Table 1. The retrieval accuracy reaches the best when $\beta = 0.5$, while the fastest convergence is achieved at $\beta = 0.125$. A smaller β means a lower weight of feature consistency preserving part in the objective function, which emphasizes the similarity embedding part. As a result, faster convergence is achieved at a smaller β. Considering a balance of retrieval accuracy and convergence speed, we fix $\beta = 0.5$ in the rest of the experiments.

Table 1. Impact of balance factor β on retrieval accuracy and convergence speed. The experiment is conducted on UKBench with VGGNet-16, and retrieval accuracy is measured by N-S Score. Convergence speed is measured by the number of epochs to reach the best retrieval accuracy.

β	N-S Score	Epochs
0.125	3.86	8
0.25	3.87	10
0.5	3.88	12
1	3.84	16
2	3.82	20

6.2.2. Impact of Nearest Neighbor Number

Nearest neighbor number k is an essential parameter in training image pairs selection since k-reciprocal nearest neighbor is leveraged. Different values of k will affect the quality and the quantity of mined training examples. To study the impact of k, we conduct experiments to evaluate the number of mined training image pairs and the precision of mined pairs. The precision is measured with the ratio of true similar pairs over all mined training pairs.

As illustrated in Figure 3, the number of mined image pairs rises with the increase of k, while the precision decreases for both UKBench and INSTRE. However, while the precision changes more smoothly on INSTRE, it drops dramatically on UKBench after $k = 4$, which is arguably because there are three other relevant images for each image in the database. Considering the balance of the quality and the number of training examples, we choose $k = 4$ for both UKBench and INSTRE. We empirically set $k = 2$ for the Holidays dataset since there is only one relevant image for many anchor images on Holidays.

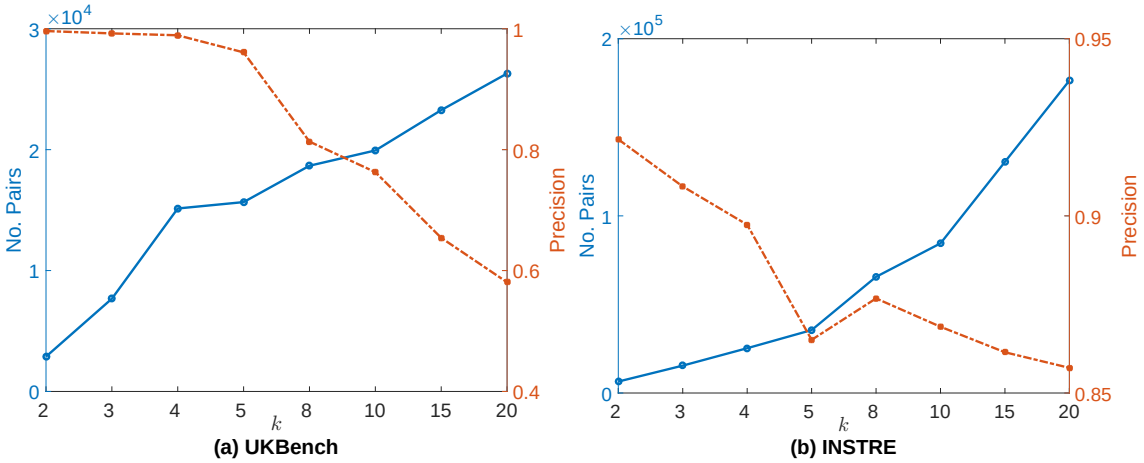

Figure 3. Impact of the number of nearest neighbor k on the number of mined image pairs (no. of pairs) and its corresponding precision. The experiment is conducted on UKBench (**a**) and INSTRE (**b**) with ResNet-50. Only the first mining iteration is recorded. Precision denotes the ratio of true positive pairs over all mined image pairs. Note that anchor image is counted, so k starts from 2.

6.3. Evaluation and Comparison

In the following part, we evaluate our proposed approach in terms of retrieval accuracy on three benchmark datasets. Firstly, we conduct experiments to prove the effectiveness of our proposed approach. Then, comparison with baselines and the state-of-the-art algorithms is carried out.

6.3.1. Effectiveness of Manifold Learning-Based Training Example Selection

To validate the effectiveness of our proposed method, a set of comparison experiments is conducted. In the comparison group, training image pairs are directly generated by selecting k-nearest

neighbors of the original similarity relationship instead of refined similarity. Retrieval performance is compared on the UKBench under the same experimental setting except for the above difference.

As demonstrated in Table 2, the manifold learning-based training pairs selection method outperforms the original one. Specifically, with the mining method based on the manifold-refined similarity, it reaches an N-S Score of 3.88, while, with the mining method based on the original similarity, the N-S Score is 3.84 on VGGNet-16. A similar situation happens on ResNet-50 with a performance boost from 3.93 to 3.96.

This improvement comes from two aspects. On the one hand, refined similarity contains extra local geometry information of the intrinsic data manifold compared to original similarity. The extra information helps to mine out some hard pairs, while it can not be done with the original one. On the other hand, the refined similarity is more accurate than the original one. Thus, some false pairs are filtered out. Fine-tuning with a cleaner training set naturally brings us better retrieval performance.

Table 2. Performance comparison of different training pairs selection methods on UKBench. Baseline refers to performance without fine-tuning. "Original" denotes the training pairs selection method based on the original similarity relationship. "Manifold" represents the method based on refined similarity with the manifold relationship. Retrieval accuracy is measured by N-S Score.

Mining Method	Network	N-S Score
Baseline	VGGNet-16	3.77
Original	VGGNet-16	3.84
Manifold	VGGNet-16	3.88
Baseline	ResNet-50	3.90
Original	ResNet-50	3.93
Manifold	ResNet-50	3.96

6.3.2. Effectiveness of Iteratively Mining and Training

In our proposed approach, the iterative mining and training process can also bring some performance improvement. In Table 3, we demonstrate the effectiveness of the iteratively mining and training process on UKBench. From Table 3, it can be observed that the first iteration boosts the N-S The score from a baseline of 3.77 to 3.86 with VGGNet-16 and 3.90 to 3.95 with ResNet-50, respectively. With one more iteration, the N-S Score ulteriorly reaches 3.88 with VGGNet-16 and 3.96 with ResNet-50, respectively. Thus, a further performance improvement of 0.02 with VGGNet-16 and 0.01 with ResNet-50 is achieved.

The improvement comes from gradually collecting and embedding hidden manifold information from the intrinsic data manifold. The first mining and training iteration may collect and embed much of the manifold information from the intrinsic data manifold but not all. With the newly learned feature embedding, feature representations of similar images move closer to each other in Euclidean space. Under this new similarity, some additional manifold information can be dug out by manifold learning methods in the form of training pairs, and then embedded into image features by fine-tuning. By iteratively conducting the mining and training process, additional manifold information is dug out and embedded until there is little left. In practice, there is little performance improvement after two or three iterations. Thus, we set the number of iterations of mining and training processes to be two for both UKBench and INSTRE. Note that the manifold learning-based re-ranking method on Holidays does not work as well as the other two datasets, and we set the number of iterations to be 1 for Holidays. In Figure 4, we show an extreme case benefiting from the iteratively mining and training process to demonstrate its effectiveness.

Additionally, some retrieval results are shown in Figure 4. For every query, the first row contains the top-3 returned images when retrieving with image features extracted with the original pre-trained VGGNet-16 model and the second row with fine-tuned VGGNet-16 model. In Figure 4a, three good cases illustrate that the fine-tuned network improves retrieval accuracy. For the first one located on the

top, one irrelevant image is in the first row. After fine-tuning, this irrelevant image is eliminated from the top-3 returned list. For the second query located in the middle, all three returned images in the first row are in the same coarse category, but only the first one is a true relevant image of this query. In the second row, two other true relevant images of the same object take the place of these two false relevant images. This is because our approach makes the fine-tuned model emphasize instance-level similarity while preserving original class-level discrimination. For the third query, there is no relevant image in the first row, while all three images are true relevant images in the second row. It is incredible since there seems to be no manifold information to learn. This owes to the iteratively mining and training process. After early mining and training iteration, these true relevant images move closer to the query in terms of feature distance, which brings some information that can be learned by the manifold learning method. With later mining and training iterations, relevant images move close enough to be top-ranked.

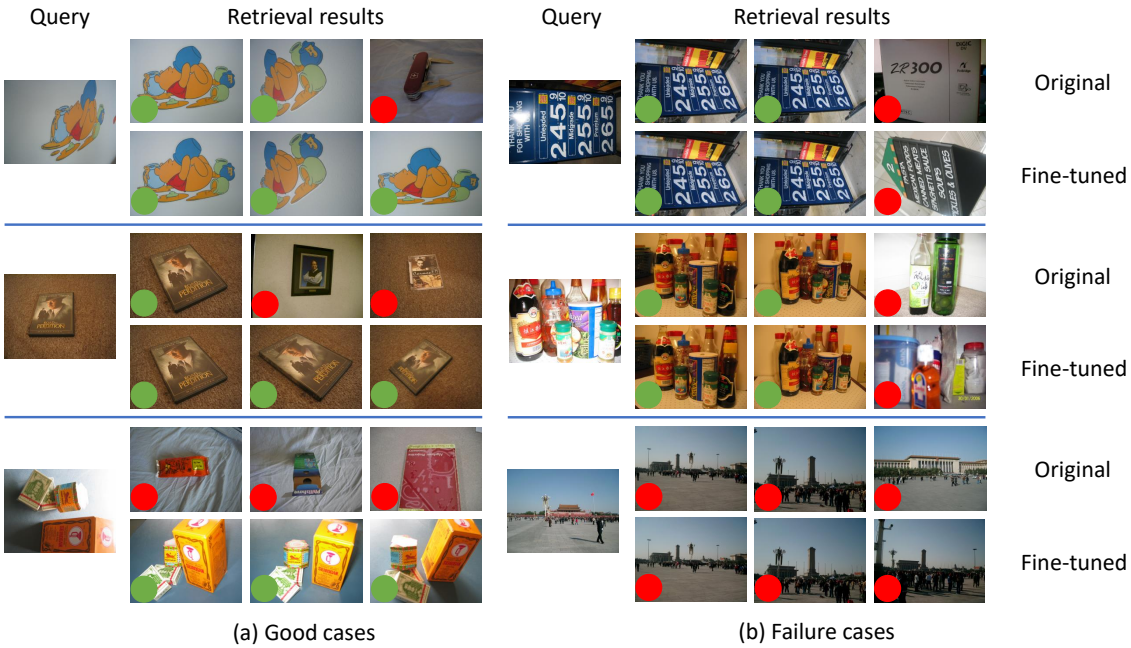

Figure 4. Selected samples, including both good cases (**a**) and failure cases (**b**). For each case, the query is on the left, and top-3 returned results are on the right. The images in the first row of each case are the results of the original pre-trained model, and in the second row are the results of the fine-tuned model. True positives and false positives are marked with green and red circles on the lower-left corner of each result image, respectively.

Table 3. Demonstration of the effectiveness of the iteratively mining and training process on UKBench. Networks are fine-tuned for three iterations, and the performance is evaluated with a N-S Score.

Network	Baseline	1st Iter.	2nd Iter.	3rd Iter.
VGGNet-16	3.77	3.86	3.88	3.88
ResNet-50	3.90	3.95	3.96	3.96

Figure 4b depicts three cases where retrieval accuracy keeps unchanged after fine-tuning. For the first and second queries, fine-tuning does not improve retrieval accuracy. However, the returned images are visually more similar to the corresponding queries. The last is a particular failure case. There is no relevant image in both the first and the second row, thus all top-3 returned images are false according to the ground truth. However, if we visually check the returned images and pay attention to the street lamp, we can find that the first two returned images contain the same street lamp in the first row, and all top three returned images contain the same street lamp as the query in the second

row. A retrieval system does not know which object users prefer to retrieve from the database when given a query of multiple objects. It is a common problem in content-based image retrieval, called the intention gap. If the target object was the street lamp instead of the building nearby, our approach should improve the retrieval accuracy.

6.3.3. Performance Improvement upon Baseline

We evaluate our proposed approach to show the performance improvement upon baseline. A comparison is performed on retrieval accuracy between the original pre-trained model and the fine-tuned model. Retrieval results are shown in Table 4, where the fine-tuned model significantly boosts retrieval accuracy on UKBench and INSTRE. Specifically, the retrieval accuracy increases from 3.77 to 3.88 on UKBench and 0.316 to 0.529 on INSTRE after fine-tuning on the VGGNet-16 model. In other words, an accuracy promotion of 0.11 and 0.213 on UKBench and INSTRE is achieved, respectively. A similar situation is observed on the ResNet-50 model, where a performance improvement from 3.90 to 3.96 on UKBench and 0.472 to 0.564 for INSTRE is achieved.

Table 4. Performance comparison between baseline and fine-tuned model with different backbones. Retrieval accuracy is measured by the N-S Score for the UKBench dataset and mAP for the Holidays dataset and the INSTRE dataset.

Dataset	Network	Baseline	Fine-Tuned
UKBench	VGGNet-16	3.77	3.88
	ResNet-50	3.90	3.96
Holidays	VGGNet-16	0.802	0.821
	ResNet-50	0.855	0.869
INSTRE	VGGNet-16	0.316	0.529
	ResNet-50	0.472	0.564

However, only limited improvement is obtained on Holidays comparing to the other two datasets. The fine-tuned model only improves mAP from 0.802 to 0.821 on the VGGNet-16 model and 0.855 to 0.869 on the ResNet-50 model, respectively. The reason comes from the dataset itself. As we know, for many images in the Holidays dataset, there is only one relevant image. Unfortunately, it is tough to get information from the intrinsic data manifold under such a situation. Most manifold learning-based methods failed to achieve performance improvement on Holidays [52]. As the state-of-the-art method, RDP only achieved a marginal performance gain on Holidays. On the other hand, this result demonstrates the effectiveness of our proposed method. Once there is extra learned information, it can be embedded into the fine-tuned model by our method.

One notable thing is that all images in UKBench are taken as queries, which may violate the common sense that no query should be seen during the fine-tuning period. We pick one image out for each category (2550 images in total) and make them unavailable during training pairs selection and network fine-tuning. In the test phase, only the pre-selected images of each category are taken as queries. Under this scenario, the N-S Score slightly dropped to 3.87 and 3.94 with VGG-16 and ResNet-50, respectively, still significantly outperforming the baseline. However, this is not possible for Holidays because there is only one relevant image for many categories. Luckily, the query set is already separated apart from the database images for INSTRE, and we achieve a remarkable performance gain of 0.213 and 0.092 with VGG-16 and ResNet-50, respectively.

6.3.4. Comparison with the State-of-the-Art

We compare our experimental results with recent state-of-the-art image retrieval methods based on compact CNN representations. As shown in Table 5, our approach achieves superior retrieval accuracy on UKBench and comparable results on Holidays.

Table 5. Performance comparison with state-of-the-art CNN-based image retrieval methods. The involved network models are marked with "A", "V", and "R" for AlexNet, VGGNet-16, and ResNet-50, respectively. Methods adopting a fine-tuned model are marked with "F"; otherwise, off-the-shelf models are implied. Retrieval accuracy is measured by the N-S Score for the UKBench dataset and mAP for the Holidays dataset. "Dims" denotes the dimensionality of image representation.

Method	Network	Dims	UKBench	Holidays
SPoC [20]	V	512	3.65	0.802
Neural Codes [19]	FA	4096	3.55	0.789
NetVLAD [35]	FV	256	-	0.821
Radenović et al. [23]	FV	512	-	0.825
Gordo et al. [22]	FV	512	3.78	0.864
Ours	FV	512	3.88	0.821
Ours	FR	2048	3.96	0.869

My approach achieves a remarkable retrieval accuracy of 3.88 with VGGNet-16 and 3.96 with ResNet-50. We also outperforms some state-of-the-art feature fusion schemes, which fuse multiple image features, including hand-crafted SIFT features and CNN features. For example, our approach exceeds Collaborative Index Embedding (CIE) [55] of N-S Score 3.86 and Query-Adaptive Late Fusion (QaLF) [56] of N-S Score 3.84. It is worth noticing that image features extracted by our fine-tuned model can be used as one of the features participating in feature fusion by CIE or QaLF, which can further boost retrieval accuracy. We also outperform RDP by 0.03 N-S Score on UKBench (3.96 vs. 3.93).

For Holidays, our approach outperforms SPoC [20] based on an off-the-shelf pre-trained model while it is comparable to other methods using a fine-tuned model. The reason lies in the intrinsic dataset property, which has been depicted in Section 6.3.3. However, for the other methods adopting fine-tuned models, they either need another labeled training dataset or need to mine training examples from an additional dataset with plenty of images related to the target dataset. To be specific, in [35], the CNN model was fine-tuned with the Pittsburgh Dataset [57] of 250K images on street view. In [19], a labeled dataset with 213,678 images of 672-category landmark buildings was adopted as the training set. In [22,23], a 713-category landmark dataset with 163,671 images and cleaned 586-category landmark dataset with 49,000 images were adopted to help the training process. As discussed previously in Section 1, these methods suffer from a time-consuming dataset collecting process and labor-expensive image labeling. It is notable that the best-performed method, Gordo et al. [22] in Table 5, adopted a different baseline of mAP 0.858 and achieved an mAP of 0.864, where the performance gain of 0.6% is marginal. My approach achieves a performance gain of 1.4% from 0.855 to 0.869 with a similar baseline based on ResNet-50.

7. Conclusions

In this paper, we propose to learn a robust feature embedding with the help of manifold relationships for image retrieval. The proposed approach applies affinity learning techniques to learn a more robust similarity, which can reflect the local geometry structure of the intrinsic data manifold. The learned similarity, in turn, is used to guide the selection process of training image pairs. As a result, fine-tuning the Siamese network with those selected image pairs transfers this learned manifold information into the newly learned feature embedding. With the newly learned feature embedding, the similarity between images is efficiently measured with easy-calculating Euclidean distance between the corresponding image features, while keeping the robust manifold relationships. Extensive experiments on several benchmark datasets demonstrate the effectiveness of our proposed approach.

Author Contributions: Conceptualization, X.C.; Methodology, X.C.; Validation, Y.L.; Writing Original Draft Preparation, X. C.; Writing Review & Editing, X.C. and Y.L.; Funding Acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Jiangsu Province under Grant No. BK20200725.

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

References

1. Zheng, L.; Yang, Y.; Tian, Q. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1224–1244. [CrossRef]
2. Al-Jubouri, H.A. Content-based image retrieval: Survey. *J. Eng. Sustain. Dev.* **2019**, *23*, 42–63. [CrossRef]
3. Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; Schölkopf, B. Ranking on data manifolds. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 13–18 December 2004; pp. 169–176.
4. Kontschieder, P.; Donoser, M.; Bischof, H. Beyond pairwise shape similarity analysis. In Proceedings of the Asian Conference on Computer Vision (ACCV), Xi'an, China, 23–27 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 655–666.
5. Luo, L.; Shen, C.; Zhang, C.; van den Hengel, A. Shape similarity analysis by self-tuning locally constrained mixed-diffusion. *IEEE Trans. Multimed.* **2013**, *15*, 1174–1183. [CrossRef]
6. Yang, X.; Koknar-Tezel, S.; Latecki, L.J. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 22–24 June 2009; pp. 357–364.
7. Pedronette, D.C.G.; Almeida, J.; Torres, R.D.S. A scalable re-ranking method for content-based image retrieval. *Inf. Sci.* **2014**, *265*, 91–104. [CrossRef]
8. Pedronette, D.C.G.; Torres, R.D.S. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognit.* **2013**, *46*, 2350–2360. [CrossRef]
9. Yang, F.; Hinami, R.; Matsui, Y.; Ly, S.; Satoh, S. Efficient image retrieval via decoupling diffusion into online and offline processing. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 9087–9094.
10. Zhou, J.; Liu, X.; Liu, W.; Gan, J. Image retrieval based on effective feature extraction and diffusion process. *Multimed. Tools Appl.* **2019**, *78*, 6163–6190. [CrossRef]
11. Rodrigues, J.; Cristo, M.; Colonna, J.G. Deep hashing for multi-label image retrieval: A survey. *Artif. Intell. Rev.* **2020**, *53*, 5261–5307. [CrossRef]
12. Bai, S.; Zhang, F.; Torr, P.H. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [CrossRef]
13. Donoser, M.; Bischof, H. Diffusion processes for retrieval revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 1320–1327.
14. Bosch, A.; Zisserman, A.; Muñoz, X. Scene classification using a hybrid generative/discriminative approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 712–727. [CrossRef]
15. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
16. Sivic, J.; Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2003; p. 1470.
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
18. 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, 23–28 June 2014; pp. 580–587.
19. 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.
20. Babenko, A.; Lempitsky, V. Aggregating local deep features for image retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1269–1277.
21. Ng, J.Y.H.; 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, 7–12 June 2015; pp. 53–61.

22. Gordo, A.; Almazán, J.; Revaud, J.; Larlus, D. Deep image retrieval: Learning global representations for image search. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 241–257.
23. Radenović, F.; Tolias, G.; Chum, O. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 3–20.
24. Wan, J.; Wang, D.; Hoi, S.C.H.; Wu, P.; Zhu, J.; Zhang, Y.; Li, J. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the ACM International Conference on Multimedia (MM), Orlando, FL, USA, 3–7 November 2014; pp. 157–166.
25. Han, X.; Leung, T.; Jia, Y.; Sukthankar, R.; Berg, A.C. Matchnet: Unifying feature and metric learning for patch-based matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3279–3286.
26. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
27. Cao, R.; Zhang, Q.; Zhu, J.; Li, Q.; Li, Q.; Liu, B.; Qiu, G. Enhancing remote sensing image retrieval using a triplet deep metric learning network. *Int. J. Remote Sens.* **2020**, *41*, 740–751. [CrossRef]
28. Min, W.; Mei, S.; Li, Z.; Jiang, S. A Two-Stage Triplet Network Training Framework for Image Retrieval. *IEEE Trans. Multimed.* **2020**, *22*, 3128–3138. [CrossRef]
29. Wiggers, K.L.; Britto, A.S.; Heutte, L.; Koerich, A.L.; Oliveira, L.S. Image retrieval and pattern spotting using siamese neural network. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
30. Gong, Y.; Wang, L.; Guo, R.; Lazebnik, S. Multi-scale orderless pooling of deep convolutional activation features. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 392–407.
31. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 23–28 June 2014; pp. 512–519.
32. Jegou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Perez, P.; Schmid, C. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1704–1716. [CrossRef]
33. Tolias, G.; Sicre, R.; Jégou, H. Particular object retrieval with integral max-pooling of CNN activations. *arXiv* **2015**, arXiv:1511.05879.
34. Li, Y.; Kong, X.; Zheng, L.; Tian, Q. Exploiting Hierarchical Activations of Neural Network for Image Retrieval. In Proceedings of the ACM International Conference on Multimedia (MM), Amsterdam, The Netherlands, 15 October 2016; pp. 132–136.
35. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
36. Bai, S.; Bai, X. Sparse contextual activation for efficient visual re-ranking. *IEEE Trans. Image Process.* **2016**, *25*, 1056–1069. [CrossRef]
37. Bai, S.; Zhou, Z.; Wang, J.; Bai, X.; Latecki, L.J.; Tian, Q. Ensemble Diffusion for Retrieval. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 774–783.
38. Bai, S.; Bai, X.; Tian, Q. Scalable person re-identification on supervised smoothed manifold. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; Volume 6, p. 7.
39. Iscen, A.; Tolias, G.; Avrithis, Y.; Furon, T.; Chum, O. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 926–935.
40. Bai, S.; Bai, X.; Tian, Q.; Latecki, L.J. Regularized Diffusion Process for Visual Retrieval. In Proceedings of the AAAI, San Francisco, CA, USA, 4–9 February 2017; pp. 3967–3973.
41. Li, Y.; Kong, X.; Fu, H.; Tian, Q. Node-sensitive Graph Fusion via Topo-correlation for Image Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 3777–3787. [CrossRef]

42. Xu, J.; Wang, C.; Qi, C.; Shi, C.; Xiao, B. Iterative Manifold Embedding Layer Learned by Incomplete Data for Large-scale Image Retrieval. *arXiv* **2017**, arXiv:1707.09862.
43. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Mining on Manifolds: Metric Learning without Labels. *arXiv* **2018**, arXiv:1803.11095.
44. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
45. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Jegou, H.; Schmid, C.; Harzallah, H.; Verbeek, J. Accurate image search using the contextual dissimilarity measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 2–11. [CrossRef] [PubMed]
47. Qin, D.; Gammeter, S.; Bossard, L.; Quack, T.; Van Gool, L. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 777–784.
48. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
49. Nister, D.; Stewenius, H. Scalable recognition with a vocabulary tree. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2161–2168.
50. Jegou, H.; Douze, M.; Schmid, C. Hamming embedding and weak geometric consistency for large scale image search. In Proceedings of the European Conference on Computer Vision (ECCV), Marseille, France, 12–18 October 2008; pp. 304–317.
51. Wang, S.; Jiang, S. INSTRE: A new benchmark for instance-level object retrieval and recognition. *ACM Trans. Multimed. Comput. Commun. Appl.* **2015**, *11*, 37. [CrossRef]
52. Bai, S.; Bai, X.; Tian, Q.; Latecki, L.J. Regularized Diffusion Process on Bidirectional Context for Object Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1213–1226. [CrossRef]
53. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. *OSDI* **2016**, *16*, 265–283.
54. Zheng, L.; Zhao, Y.; Wang, S.; Wang, J.; Tian, Q. Good practice in CNN feature transfer. *arXiv* **2016**, arXiv:1604.00133.
55. Zhou, W.; Li, H.; Sun, J.; Tian, Q. Collaborative index embedding for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1154–1166. [CrossRef]
56. Zheng, L.; Wang, S.; Tian, L.; He, F.; Liu, Z.; Tian, Q. Query-adaptive late fusion for image search and person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1741–1750.
57. Torii, A.; Sivic, J.; Pajdla, T.; Okutomi, M. Visual place recognition with repetitive structures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 883–890.