A Two-Stage Triplet Network Training Framework for Image Retrieval

Weiqing Min, *Member, IEEE*, Shuhuan Mei, Zhuo Li and Shuqiang Jiang, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a novel framework for instance-level image retrieval. Recent methods focus on fine-tuning the Convolutional Neural Network (CNN) via a Siamese architecture to improve off-the-shelf CNN features. They generally use the ranking loss to train such networks, and do not take full use of supervised information for better network training, especially with more complex neural architectures. To solve this, we propose a two-stage triplet network training framework, which mainly consists of two stages. First, we propose a Double-Loss Regularized Triplet Network (DLRTN), which extends basic triplet network by attaching the classification sub-network, and is trained via simultaneously optimizing two different types of loss functions. Double-loss functions of DLRTN aim at specific retrieval task and can jointly boost the discriminative capability of DLRTN from different aspects via supervised learning. Second, considering feature maps of the last convolution layer extracted from DLRTN and regions detected from the region proposal network as the input, we then introduce the Regional Generalized-Mean Pooling (RGMP) layer for the triplet network, and re-train this network to learn pooling parameters. Through RGMP, we pool feature maps for each region and aggregate features of different regions from each image to Regional Generalized Activations of Convolutions (R-GAC) as final image representation. R-GAC is capable of generalizing existing Regional Maximum Activations of Convolutions (R-MAC) and is thus more robust to scale and translation. We conduct the experiment on six image retrieval datasets including standard benchmarks and recently introduced INSTRE dataset. Extensive experimental results demonstrate the effectiveness of the proposed framework.**

I. Introduction

Instance-level image retrieval aims at retrieving all images that contain the same object instance as the query image from a large unordered collection of images. It has received much attention in the multimedia community [1], [2], [3], [4] for its wide range of applications, such as visual geo-localization [5], [6], organization of personal photo collections and 3D reconstruction [7]. Earlier works heavily rely on hand-crafted local descriptors, e.g., SIFT [8], [9], [1] and their variants [10], [11], [2]. Recent advances [12], [13], [3], [4] show that CNNs offer an attractive alternative for image search with small memory cost.

Most of deep retrieval methods use CNN as a generic feature extractor from pre-trained models on the ImageNet [14]. The extracted features are usually activations of high layers (e.g., fully connected layers and convolution layers), taken as holistic feature representations for image retrieval. For example, Tolias *et al.* [15] proposed a Regional Maximum Activations of Convolutions (R-MAC) obtained via aggregating activation features of convolutions in a fixed layout of spatial regions. Although R-MAC improves existing image retrieval methods, the use of off-the-shelf CNN features may not be the optimal choice since their features are not necessarily suited for the special retrieval task. Therefore, fine-tuning CNN is an alternative to improve the adaptation ability. Recently, Gordo *et al.* [16] proposed a triplet network that combined three streams with a ranking loss to fine-tune CNN to produce better feature representation than [15]. However, they do not take full use of supervised information, and thus do not effectively utilize different types of loss functions for better network training, especially in more complex triplet networks.

Multiple types of loss functions can constrain parameters of a neural network for the specific retrieval task from different aspects, and help regularly learn a CNN to boost its discriminative capability [17]. For example, the softmax loss function is used for minimizing the cross-entropy loss over all training samples. The pairwise ranking loss accounts for the ordinal ranking over all the training images to understand fine-grained differences between image pairs. To the best of our knowledge, none of previous works have trained the network, especially the more complex triplet network with different types of losses for image retrieval. In this paper, focusing on CNN based image retrieval, we extend the network architecture [16] by introducing the classification sub-network and propose a double-loss regularized triplet network, which jointly uses both triplet ranking loss and classification loss for fine-tuning the CNN with both ranking sub-network and classification sub-network. The activations of convolution layers from such fine-tuned network can be used to produce more discriminative global features via pooling and aggregation.

Previously, a number of pooling strategies have been used. These range from max pooling [18], average pooling [19] to regional pooling [15]. Recently, generalized-mean pooling [20] is proposed to generalize max and average pooling to boost the retrieval performance. We also introduce the generalized-mean pooling layer to the triplet network. Different from the work [20] using a spatial generalized-mean pooling over activations of the convolution layer on the whole image, we utilize the

This work was supported in part by the National Natural Science Foundation of China under Grant 61532018 and 61602437, in part by Beijing Natural Science Foundation under Grant L182054, in part by National Program for Special Support of Eminent Professionals and National Program for Support of Top-notch Young Professionals. W. Min, Z. Li and S. Jiang is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China, and also with University of Chinese Academy of Sciences, Beijing, 100049, China email: minweiqing@ict.ac.cn, sqjiang@ict.ac.cn, zhuo.li@vipl.ict.ac.cn. S. Mei is with Shandong University of Science and Technology, Shandong, 266590, China, and also an intern with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China shuhuan.mei@vipl.ict.ac.cn.

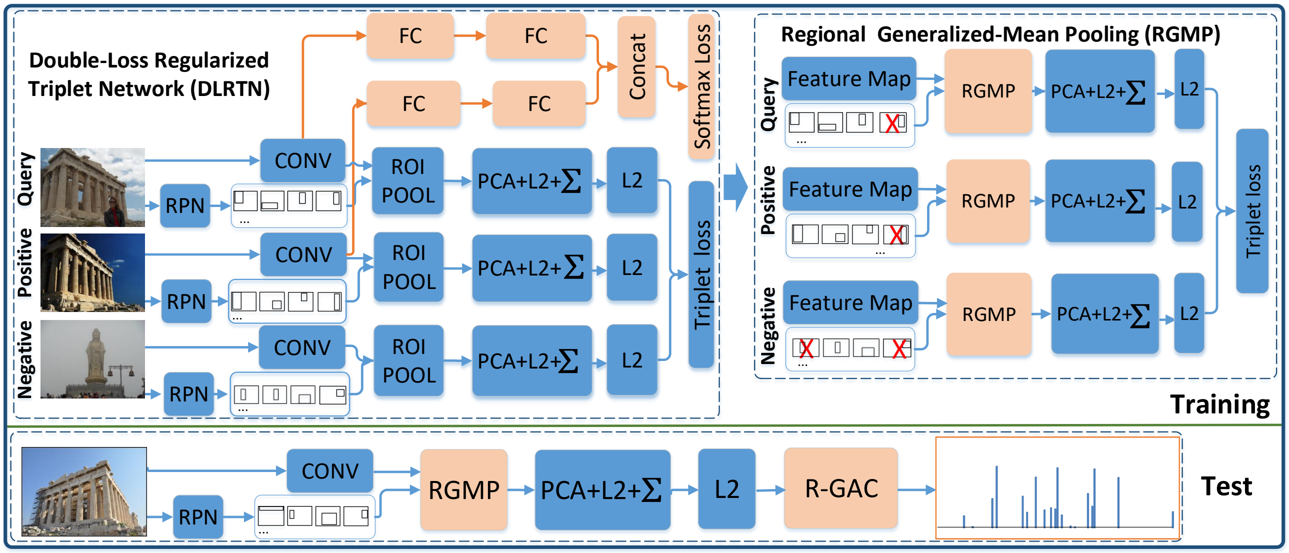


Fig. 1: The overview of our proposed framework. In the training phrase, a Double-Loss Regularized Triplet Network (DLRTN) is first trained via simultaneously optimizing both triplet loss and softmax loss. The triplet network with introduced Regional Generalized-Mean Pooling (RGMP) layer is then re-trained to learn pooling parameters. In the test phrase, one image is fed into the trained network to produce a global image representation R-GAC that can be compared with images from the dataset to produce ranked retrieval results. Note that feature maps from the last convolutional layer are more competitive than the full-connected layer [15], and thus we adopt feature maps from the last convolutional layer of DLRTN.

generalized-mean pooling layer to pool feature maps for each detected image region, and then aggregate pooled features from different regions to the Regional Generalized Activations of Convolutions (R-GAC) as the final image representation. R-GAC can generalize existing R-MAC and is more robust to scale and translation.

The combination of the above-mentioned two factors produces a novel framework (Fig. 1), which mainly consists of two stages (top of Fig. 1). Given the candidate regions detected from the Region Proposal Network (RPN), we first propose a Double-Loss Regularized Triplet Network (DLRTN), which extends basic triplet network by attaching the classification sub-network and is trained via simultaneously optimizing both loss functions with back-propagation. Meanwhile, irrelevant regions are removed in the training process. Considering the extracted feature maps of the last convolutional layer and remaining regions from DLRTN, we then introduce the Regional Generalized-Mean Pooling (RGMP) layer for the triplet network, and re-train the network to learn pooling parameters. Through RGMP, we take feature maps from each region as the input to produce a pooled feature vector for each region, and then aggregate feature vectors from different regions to Regional Generalized Activations of Convolutions (R-GAC) as final image representation. In the test stage (bottom of Fig. 1), one image is fed into the trained network to produce a global image representation R-GAC that can be compared with images from the dataset with a dot-product to produce ranked retrieval results. An extensive experimental study on six image retrieval datasets has demonstrated the effectiveness of our framework.

The major contributions of this paper are summarized as follows.

- We propose a novel two-stage triplet network training framework, which can jointly utilize different types of loss functions from both ranking sub-network and classification sub-network, and regional generalized-mean pooling method to produce a global and compact fixed-length representation.
- We propose a Double-Loss Regularized Triplet Network (DLRTN), which consists of both ranking sub-network and classification sub-network to fine-tune the network. Double-loss functions of DLRTN aim at the specific retrieval task and jointly boost the discriminative capability of DLRTN from two different aspects.
- We conduct an extensive experimental study on six image retrieval datasets and experimental results demonstrate the effectiveness of our proposed framework.

II. Related Work

The proposed framework is mainly related with CNN-based image retrieval and CNN based metric learning, which will be briefly reviewed in this section.

CNN-based Image retrieval A more comprehensive survey of instance retrieval is provided in [21]. Recently, CNN-based visual features have been widely used for image retrieval because of their effective feature representation [22], [23], [3], [4]. It mainly consists of two types: off-the-shelf CNN and fine-tuning CNN for image retrieval. Razavian *et al.* [24] earlier applied off-the-shelf CNN features to evaluate the retrieval performance quantitatively. Several methods are then proposed to overcome their limits, such as the scaling, cropping and clutter. For example, Babenko *et al.* [19] adopted the sum-pooling mechanism to aggregate local deep features to obtain compact global descriptors, namely SPoC for image retrieval. Kalantidis *et al.* [25] extended SPoC by allowing cross-dimensional weighting and aggregation of neural codes.

Other approaches proposed hybrid models with certain encoding e.g., Fisher Vector [26] and VLAD [27]. Tolias *et al.* [15] further aggregated activation features of a pre-trained CNN in a fixed layout of spatial regions to produce the R-MAC descriptor, which significantly improved existing CNN-based recognition methods. In addition, some works [28] integrated fully-connected activations of CNN and SIFT features at different levels to improve the retrieval performance. Recently, Shuhuan *et al.* [29] first detected the image region using Faster R-CNN and then extracted off-the-shelf CNN features from detected regions as image features for retrieval.

Instead of using off-the-shelf CNN features on the ImageNet, some methods have explicitly fine-tuned the network more suited for the retrieval task. The first fine-tuning approach for image retrieval is proposed by Babenko *et al.* [12], where extensive evaluation showed that pre-trained models for object classification could be improved by fine-tuning them on an external data set. Some works [16], [30], [13], [5] also confirmed that fine-tuning pre-trained models for retrieval can bring a significant improvement under the combination of a good image representation and a ranking loss. For example, Arandjelovic *et al.* [5] proposed a NetVLAD network to learn a ranking network using external dataset for retrieval. Gordo *et al.* [30] trained a triplet network with a triplet loss to produce a global image representation suited for image retrieval. In our work, we also fine-tune the CNN for image retrieval. Different from these methods, we introduce additional fusion sub-network, which consists of a set of convolutional layers culminating in a fully-connected layer with softmax loss and fine-tune the network by simultaneously optimizing the ranking loss and classification loss.

In addition, different pooling methods for CNNs also influence the performance of image retrieval. Because lower layers are more general and preserve certain spatial information [31], many works [18], [32], [15] used different pooling methods to fuse activations of convolutional layers as image features, such as max pooling [18], average pooling [19], sum pooling [32] and weighted sum pooling [25]. Tolias *et al.* [15] further proposed a hybrid scheme, which aggregated the maximum activation over multiple spatial regions sampled on the convolution layer using a fixed layout to produce a global representation. Recently, Radenovic *et al.* [20] proposed a trainable generalized-mean pooling method that generalized both max and average pooling with learned parameters. However, they should first collect large-scale instance-relevant samples, and then use them to 3D reconstruction. Different from the work [20], which used a spatial generalized-mean pooling over activations of convolution layers on the whole image to produce Generalized Activations of Convolutions (GAC), we utilize the generalized-mean pooling layer for each region from the image, and then aggregate pooled features from different regions to obtain Regional Generalized Activations of Convolutions (R-GAC). All these methods can be combined with other post-processing techniques such as query expansion [33] and spatial verification [34]. [13], [20] for further improving the retrieval performance.

CNN based Metric Learning CNN based metric learning methods first learn a non-linear transformation of an input image via CNN and is then fine-tuned on task-specific datasets. To jointly learn all parameters of the CNN and embedding, ranking loss functions operating on image pairs or triplets are used. One of the most widely used pairwise loss functions for metric learning is the contrastive loss function [35], [36]. This loss function minimizes the squared Euclidean distance between positive feature vectors while encouraging a margin between positive and negative pairs. To train networks with this loss function, Siamese networks have widely been used for metric learning [35], [37]. Triplet networks have further been used for metric learning [38], [16], [39]. For example, Gordo *et al.* [16] proposed to train a triplet deep network with a triplet ranking loss for image retrieval. Different from their proposed triplet networks [16], which are trained based on only the ranking loss, we propose a double-loss regularized triplet network with joint classification loss and ranking loss. In addition, our proposed framework introduces the regional generalized-mean pooling layer to improve global feature representation.

III. Proposed Framework

Our framework mainly consists of two components: Double-Loss Regularized Triplet Network (DLRTN) and Regional Generalized-Mean Pooling (RGMP), which will be introduced in more details in the following. In addition, we will briefly introduce the global image feature generation in the test stage from our framework and triplets selection.

A. Double-Loss Regularized Triplet Network (DLRTN)

1) Learned R-MAC: Our goal is to aggregate features from different regions to produce global image representation, like learned R-MAC [16], which has demonstrated the effectiveness in image retrieval tasks. Therefore, as shown in the top left part of Fig.1, we utilize one basic triplet network architecture. Each stream of the triplet network contains convolutional layers of a pre-trained network (e.g., VGG16 [40])(CONV) and Region of Interest pooling (ROI POOL). The former is used to extract activation features from images while the latter is used for the spatial pooling in different regions. These pooled region features are independently l_2-normalized, whitened with PCA and l_2-normalized again. They are finally sum-aggregated ($\sum$) and l_2-normalized to produce global and compact feature representation.

We employ the Region Proposal Network (RPN) to localize regions of interest in images to produce candidates of regions. The main idea of RPN [41] is to predict a score describing how likely each box contains an object of interest for a set of candidate boxes at all possible locations. Similar to [16], we model this process with a fully-convolutional network built on top of the convolutional layers. This allows one to get region proposals at almost zero cost. We obtain candidate regions of interest from each image via RPN.

Existing methods generally utilize the ranking loss L_r based on image triplets for the triplet network. It explicitly enforces that, given a query, a positive image and a negative one, the positive one is closer to the query than the negative one. Let I_q be one query image with visual representation q, I^+ be one relevant image with descriptor d^+, and I^- be one non-relevant image with descriptor d^-. The triplet ranking loss is defined as

$$L_r(I_q, I^+, I^-) = \frac{1}{2}\max(0, m + \|q - d^+\|^2 - \|q - d^-\|^2) \quad (1)$$

where m is a scalar that controls the margin. Given a triplet with non-zero loss, the sub-gradients are given by

$$\frac{\partial L_r}{\partial q} = d^- - d^+, \frac{\partial L_r}{\partial d^+} = q - d^+, \frac{\partial L_r}{\partial d^-} = q - d^- \quad (2)$$

The sub-gradients are back-propagated through three streams of this network, and convolutional layers together with PCA layers get updated. This approach directly optimizes a ranking objective.

2) DLRTN: Training only based on the ranking loss may not be sufficient to train such complex triplet network. Therefore, we modify this basic triplet neural architecture by attaching a classification sub-network and further introduce the classification loss L_c. Particularly, the feature extractor layers (e.g., the last conv layer conv5_3 in VGG16) for query and positive samples are followed by a fusion sub-network, which consists of a set of Fully-Connected layers (FC) and one fusion layer (Concat) with the softmax loss L_c (Fig. 1). For the softmax loss, the normalized probability of the i-th image pair $[q, d^+]_i$ in the j-th class can be computed by

$$p_{i,j} = \frac{\exp(Q_j([q, d^+]_i))}{\sum_{j=1}^{C}\exp(Q_j([q, d^+]_i))} \quad (3)$$

where $[q, d^+]$ denotes joint representation between q and d^+ from the classification sub-network. $Q_j([q, d^+]_i)$ is to specify discrete probability distribution of the i^{th} image pair $[q, d^+]$ in the j-th class, $j = 1, ..., C$, where C is the number of possible classes. Therefore, it is a multi-class classification over all the classes C.

The softmax loss is specified as

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\bar{p}_{i,j}\log(p_{i,j}) \quad (4)$$

where N is the number of image pairs for query samples and their positive samples. $\bar{p}_{i,j}$ denotes the ground-truth probability of $[q, d^+]_i$ in class j.

To train the triplet network, we should minimize the total loss function L, which is a weighted combination of the ranking loss L_r and the classification loss L_c as follows:

$$L = L_r(I_q, I^+, I^-) + \lambda L_c(I_q, I^+) \quad (5)$$

where λ is the hyper-parameter.

In this case, given a triplet with non-zero loss, the sub-gradients are revised by

$$\begin{aligned}\frac{\partial L}{\partial q} &= d^- - d^+ + \lambda\frac{\partial L_c}{\partial q}\\ \frac{\partial L}{\partial d^+} &= q - d^+ + \lambda\frac{\partial L_c}{\partial d^+}\\ \frac{\partial L_r}{\partial d^-} &= q - d^- + \lambda\frac{\partial L_c}{\partial d^-}\end{aligned} \quad (6)$$

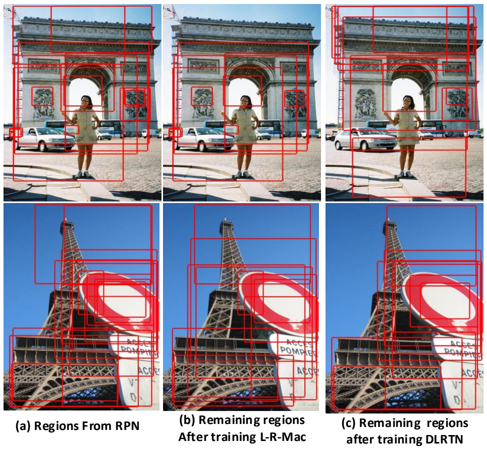

(a) Regions From RPN
(b) Remaining regions After training L-R-Mac
(c) Remaining regions after training DLRTN

Fig. 2: The selected regions from DLRTN and L-R-MAC [16] after the training.

Although DLRTN adopts some basic network layers presented in [16], but with two important differences: First, different from previous three-stream Siamese network only using the ranking loss, we further add a classification sub-network, leading to the joint optimization on both the ranking and classification loss. Different loss functions from our triplet network can constrain parameters from different aspects and help boost its discriminative capability. Second, the goal of [16] is directly to learn visual feature representation while our proposed network mainly focuses on feature map learning for convolution layers. Meanwhile, we harvest discriminative image regions in the training process based on two-losses optimizations. Fig. 2 shows some examples. We can see that the introduced classification sub-network from DLRTN can reduce noisy and fine-grained regions compared with [16].

B. Regional Generalized-Mean Pooling (RGMP)

The selected image regions and extracted feature maps after DLRTN training are further used to effectively learn the triplet network with introduced Regional Generalized-Mean Pooling (RGMP).

For the feature map $\mathcal{X}$ with $W \times H \times K$ dimensions of the convolution layer for each image region, where K is the number of feature channels. We denote the feature map as a set of 2D feature responses, $\mathcal{X} = \{\mathcal{X}_i\}, i = 1...K$, where $\mathcal{X}_i$ is one 2D tensor representing responses of the i^{th} feature channel over the set of valid spatial locations Ω, and $\mathcal{X}_i(p)$ is the response at position p. Note that feature maps from the last convolutional layer are more competitive than the full-connected layer [15], and thus we adopt the last convolutional layer from trained DLRTN.

We next take feature maps $\mathcal{X}$ from each region as the input, and use a pooling layer to produce final feature representation $\mathbf{f}$ as an output of the pooling process. Generally, these feature vectors are produced via max pooling or average pooling as follows:

$$\mathbf{f}_{\Omega}^{m} = [\mathbf{f}_{\Omega,1}^{m}...\mathbf{f}_{\Omega,i}^{m}...\mathbf{f}_{\Omega,K}^{m}], \mathbf{f}_{\Omega,i}^{m} = \max_{p\in\Omega}\mathcal{X}_i(p) \quad (7)$$

$$\mathbf{f}^a_\Omega = [\mathbf{f}^a_{\Omega,1}...\mathbf{f}^a_{\Omega,i}...\mathbf{f}^a_{\Omega,K}], \mathbf{f}^a_{\Omega,i} = \frac{1}{|\Omega|}\sum_{p\in\Omega}\mathcal{X}_i(p) \tag{8}$$

Recently, the work [20] introduced the generalized-mean pooling, which is given by

$$\mathbf{f}^g_\Omega = [\mathbf{f}^g_{\Omega,1}...\mathbf{f}^g_{\Omega,i}...\mathbf{f}^g_{\Omega,K}], \mathbf{f}^g_{\Omega,i} = (\frac{1}{|\Omega|}\sum_{p\in\Omega}\mathcal{X}^\xi_i(p))^{\frac{1}{\xi}} \tag{9}$$

where ξ is the parameter and can be learned as a part of back-propagation.

The output of the pooling is Generalized Activations of Convolutions (GAC), which is the analogy with Maximum Activations of Convolutions (MAC) [15] from max-pooling via Eqn. 7. The generalized-mean pooling generalizes the max-pooling and average-pooling and thus GAC is more effective for specific retrieval task. This is because the parameter ξ is learned based on the dataset from the special task. In our work, we also adopt this pooling method. However, different from the work [20], which used a spatial generalized-mean pooling over activations of the convolution layer on the whole image to produce GAC, we utilize the generalized-mean pooling layer to pool feature maps for each image region detected from RPN, and then aggregate pooled features from different regions to Regional Generalized Activations of Convolutions (R-GAC). Compared with GAC, R-GAC is more robust to scale and translation. As shown in the top right part of Fig. 1, the input is the feature map and selected regions from DLRTN and we re-train the triplet network to learn ξ.

C. R-GAC Generation for Image Retrieval

In the test stage, we feed each image to the trained DLRTN and extract feature maps of convolution layers from this image. For each detected region from RPN, we pool feature maps from DLRTN via Eqn. 9. We then aggregate pooling features from different regions via PCA, l_2 normalization and sum-pooling to obtain R-GAC. A dot-product is used to calculate the similarity between the query and images from the dataset.

D. Triplets Selection

For the triplet network, it is crucial to select image triplets to ensure fast convergence. First, we create a dataset of tuples $(q, m(q), N(q))$, where q represents a query image, $m(q)$ is a positive image that matches the query, and $N(q)$ is a set of negative images that do not match the query. These tuples are used to form training image pairs. Similar to [35], we should select hard positive images and hard negative images. For hard positive images, we select the image which has the highest descriptor distance to the query from the same class as positive image according to

$$m(q) = \max_{i\in M(q)} \|\bar{f}(q) - \bar{i}(q)\| \tag{10}$$

where $M(q)$ is the dataset, which has the same class label with q. $\bar{f}(q)$ and $\bar{i}(q)$ denote feature descriptors.

Similarly, we select hard negative images from different categories which are different from the query image, according to

$$m_1(q) = \min_{j\in N(q)} \|\bar{f}(q) - \bar{j}(q)\| \tag{11}$$

where $N(q)$ is the dataset and the class label from $N(q)$ is different from the query.

The process of generating triples is as follows: for each class, we randomly select one image as the query, and then find one positive image according to Eqn. 10. We select each negative image for each of remaining classes according to Eqn. 11.

Through the above-mentioned method, we obtain initial training triplet set. However, when learning our triplet network, similar to [16], we generate triplets offline every n epoches using the most recent network checkpoint ($n = 5$) for practical considerations, and then compute the losses of all the triplets involving those features and select triplets with a non-zero loss as hard negatives.

IV. Experiment

In this section, we start by describing the dataset and experimental setup. We then evaluate different components of our proposed framework, and finally compare our results with the state of the art.

A. Datasets

We validate the effectiveness of our method based on the following six image retrieval datasets, including Oxford5k [9], Paris6k [42], Oxford105k [9], Paris106k [42], INSTRE-S [43] and INSTRE-S+1M [43]. The first four datasets are standard image retrieval benchmarks while the last two are recently introduced instance search datasets, which contains various objects from buildings to logos with many variations such as different scales, rotations, and occlusions, making them more challenging.

- **Oxford Buildings dataset [9]**. This dataset consists of 5,062 Oxford landmark images (namely ***Oxford 5k***). These landmark images have been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. This gives a set of 55 queries. These 55 query images are also used as the training set for the network training. The remaining is used as the image dataset for retrieval. To test larger-scale instance-level retrieval, we consider the ***Oxford 105k***, where the additional 100,000 images from Flickr are introduced as the distractors.
- **Paris dataset [42]**. This Dataset consists of 6,412 images collected from Flickr by searching for particular Paris landmarks with 12 different landmarks, namely ***Paris 6k***. For each landmark, there are 5 query images. This gives a set of 60 queries, which are also used as the training set for the network training. The remaining is used as the dataset for retrieval. Same as the Oxford dataset, we also use the additional 100,000 images for a larger-scale scenario, namely ***Paris 106k***.
- **INSTRE [43]**. INSTRE consists of two kinds of dataset: the INSTRE-S and INSTRE-M. In this work, we select ***INSTRE-S*** in our experiment, which contains 23,070 images with 200 classes in total and each image is provided with object

TABLE I: The performance(mAP) comparison for different components of our method (%).

Method	Dim.	Oxford5k	Oxford105k	Paris6k	Paris106k	INSTRE-S	INSTRE-S+1M
L-R-MAC[16]	512	83.1	78.6	89.1	79.7	75.6	32.8
DLRTN	512	84.2	79.8	90.8	80.7	77.2	33.6
DLRTN+RGMP	512	**85.3**	**81.4**	**91.6**	**82.4**	**79.6**	**34.2**

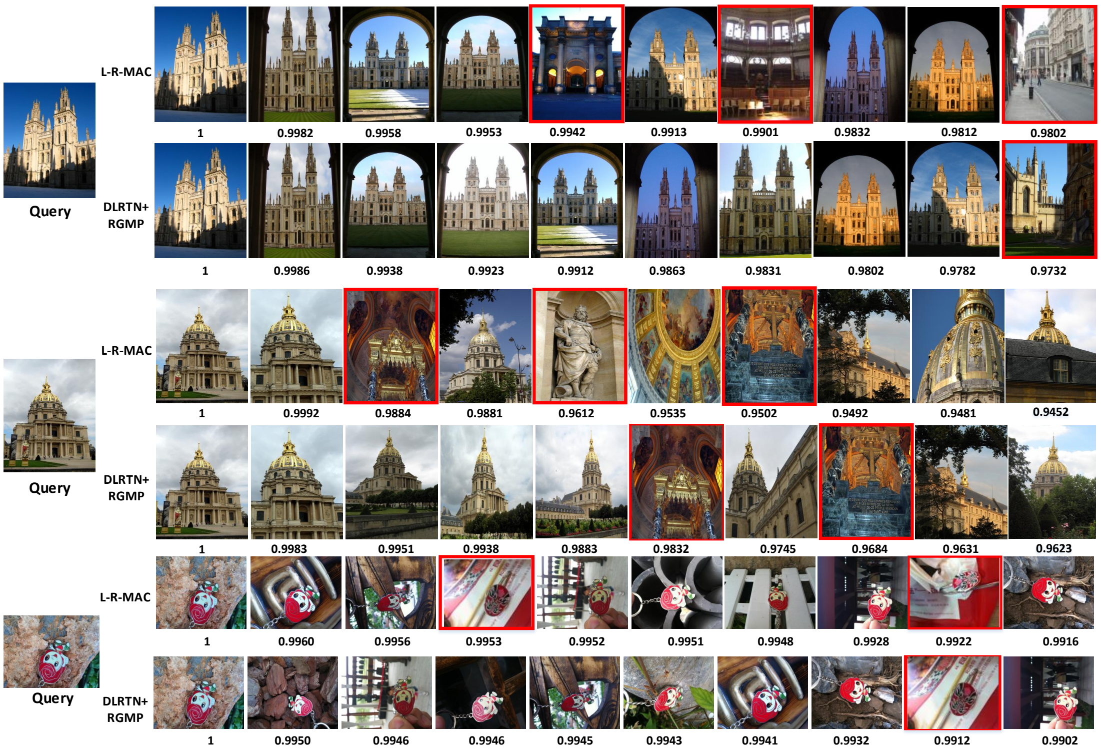


Fig. 3: Examples of top retrieved images between our method (DLRTN+RGMP) and L-R-MAC, where the images with the red box are wrong retrieved ones and the ranked score is shown below each image.

location annotations. For each class, we randomly select 75 images as the query, resulting in $200 \times 75 = 15,000$ queries. These query images are also used as the training set for the network training. The remaining is used as the image dataset for retrieval. In addition, there are one million distractor images crawled from Flickr, namely ***INSTRE-S+1M***.

B. Experimental Setup

Our framework is implemented on the Caffe [44] platform. For the convolutional part of our network, we select the popular architecture VGG16 [40], which is pre-trained on the Imagenet dataset. We apply the Adam [45] algorithm to train our network. The initial learning rate is $l_0 = 1 \times 10^{-6}$ and changes using the Adam rules with exponential decay $l_0 \exp^{-0.1i}$ over epoch i. The momentum is 0.9 and weight decay is 5×10^{-4}. All the training images are resized to a size of 256×256. The batch size of triplets is 8. We empirically set the margin $m = 0.75$. The initial parameter $\xi = 3$.

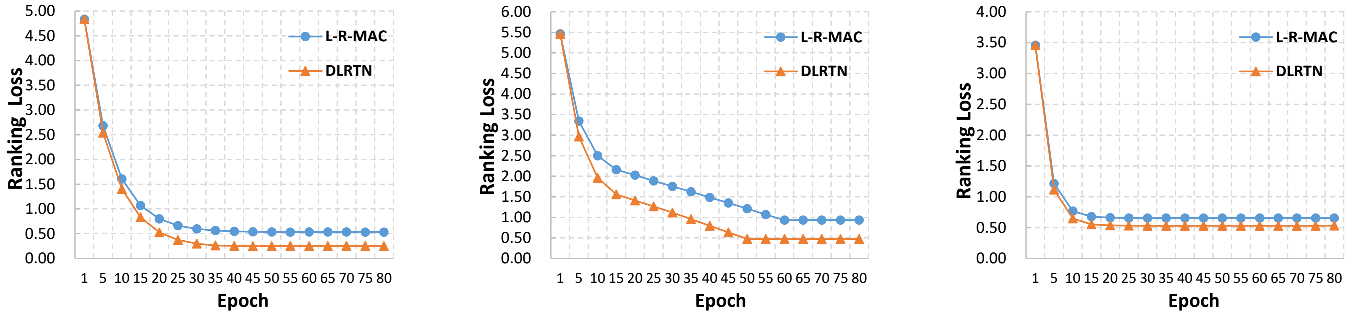
For our training strategy, we only fine-tuned certain layers in the network training. Particularly, in the first training stage, similar to [16], we first fine-tuned RPN network using the training set. Then each region extracted from RPN is fed into the backbone network VGG-16. The last convolutional layer conv5_3 is followed by PCA layers, and FC layers in DLRTN. We only fine-tuned the conv5_3 layer from VGG-16 and its following layers, such as PCA and FC layers. In the second training stage, similarly, we mainly fine-tuned the RGMP layer and its following layers.

In order to construct triplets, in the Oxford and Paris dataset, we use the official provided query images and select a positive example for each query whose label corresponds to ok or junk, and select one negative example from each of the remaining categories. For fair comparison with previous methods, we use all images as query images in the INSTRE-S dataset. The images in each category are very similar, so we select one image in each category as a query for building a tuple. For each query, we select a positive example for each query, and select one of the other 199 categories as a negative example.

We use the standard evaluation metric **m**ean **A**verage **P**recision (**mAP**) for all datasets. As is the standard practice, in Oxford and Paris, we use only the annotated region of interest of the query, while for INSTRE, we use the whole query image for fair comparison with other baselines.

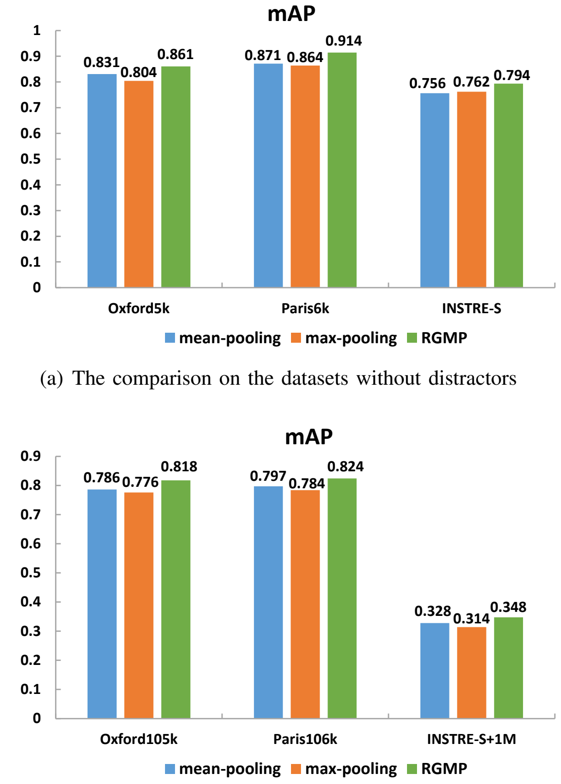
C. Evaluation of the proposal framework

1) Quantitative and Qualitative Evaluation: In this section we evaluate the effectiveness of each component in our frame-



(a) Oxford5k (b) Paris6k (c) INSTRE-S

Fig. 4: Ranking losses with different epochs on three datasets between L-R-MAC and DLRTN.



(a) The comparison on the datasets without distractors

(b) The comparison on the datasets with distractors

Fig. 5: The comparison between RGMP and other pooling methods

work. Since we extended the neural architecture from Learned R-MAC [16], we compare each improvements in our framework with [16]. Table I shows the experimental results,where L-R-MAC denotes the Learned R-MAC. We can see that (1)The performance of DLRTN is better than L-R-MAC. This is because compared with L-R-MAC, DLRTN further introduced the classification sub-network. Two different types of loss functions of DLRTN can help regularly learn a CNN for boosting its discriminative capability. (2) After introducing the RGMP, our method further improves the performance of DLRTN. This verifies the effectiveness of our method in utilizing the generalized mean pooling on different regions from the image. Through RGMP based network training, we can learn the best fusion parameter for the training dataset. Better fusion on the activations of convolutions from different regions leads to more discriminative feature representation. There is about 1.5 percent improvement over DLRTN on average over six datasets. (3) The double-loss regularization and RGMP are both complementary, and jointly improved the performance of image retrieval. We observe that our method on all six different datasets consistently improves the performance of L-R-MAC. There is about 2.5 percent improvement on average over six datasets. In addition, some examples of retrieval results are presented in Fig. 3.

2) The effect of double-losses in DLRTN: Additional classification loss can not only help regularly learn the deep network to boost its discriminative capability, but also lead to lower ranking loss. Fig. 4 showed that ranking losses of the DLRTN model and L-R-MAC trained in the first 80 epochs on all the three datasets Oxford5k, Paris6k and INSTRE-S, respectively. We can see that DLRTN consistently has lower ranking loss than L-R-MAC. In the same epochs, DLRTN has a smaller ranking loss than L-R-MAC. The possible reason is that introducing the classification sub-network can make the network training more adequate.

3) Effect of $\boldsymbol{\xi}$ from RGMP: Similar to [20], we compared the pooling layer with learned parameters ξ with mean-pooling and max-pooling during the CNN fine-tuning with RGMP. We present the results on in Fig. 5. As shown in Fig. 5, the RGMP layer consistently outperforms the conventional average and max pooling on three datasets. Particularly, after the fine-tuning of the triplet network with RGMP, The learned parameter $\boldsymbol{\xi} = 2.72$ for Oxford, $\boldsymbol{\xi} = 2.65$ for Paris and $\xi = 2.85$ for INSTRE-S.

4) Effect of λ: The performance for varying λ is plotted in Fig. 6, where we set $\lambda = \{0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7\}$. The plot suggests that our network works best on all the datasets when $\lambda = 1.5$. Therefore, we select $\lambda = 1.5$ in our experiment.

D. Comparison with the state of the art

In this section, we thoroughly compare our proposed framework with the state-of-the-art methods in the image retrieval task.

We first compare our experimental results with the current state of the art in the Table II and Table IV, with other

TABLE II: The performance (mAP) comparison with the state-of-the-art in Oxford and Paris dataset (%).

Method	Dim.	Oxford5k	Oxford105k	Paris6k	Paris106k	
MAC [18]	512	56.4	47.8	72.3	58.0	
Patch-CKN [46]	256K	56.5	-	-	-	
FM-VLAD[47]	128	59.3	59.0	-	-	-
SPOC [32]	256	68.1	61.1	78.2	68.4	
Crow [25]	512	70.8	65.3	79.7	72.2	
R-MAC [15]	512	66.9	61.6	83.0	75.7	
NetVLAD-Off-Shelf [5]	4,096	66.6	-	77.4	-	
NetVLAD-Finetune [5]	4,096	71.6	-	79.7	-	
Bow-CNN [48]	n/a	73.9	59.3	82.0	64.8	
SIAM-FV [39]	512	81.5	76.6	82.4	-	
UFT-MAC [13]	512	79.7	73.9	83.8	76.4	
L-R-MAC [16]	512	83.1	78.6	87.1	79.7	
SBA [49]	4,096	79.1	73.6	86.1	80.4	
Mask-MAC [50]	**4,096**	83.8	80.6	88.3	**83.1**	
Ours	**512**	**85.3**	**81.4**	**91.6**	82.4	

TABLE III: The performance (mAP) comparison with the state-of-the-art in Oxford and Paris dataset with QE (%).

Method	Dim.	Oxford5k	Oxford105k	Paris6k	Paris106k
BoW(1M)+QE [51]	n/a	82.7	76.7	80.5	71.0
CroW+QE [25]	512	72.2	67.8	85.5	79.7
R-MAC+AML+QE [15]	512	77.3	73.2	86.5	79.8
UFT-MAC [13]	512	85.0	81.8	86.5	78.8
L-R-MAC+QE [16]	512	89.1	87.3	91.2	86.8
SBA+QE [49]	4,096	81.7	80.6	89.2	84.7
R-MAC+SP [4]	512	79.0	70.7	91.2	84.7
R-MAC+FU [4]	512	79.4	72.9	92.1	85.7
Global Diff. [52]	512	85.7	82.7	94.1	92.5
Region Diff. [52]	**5×512**	91.5	84.7	**95.6**	**93.0**
Ours+QE	**512**	**91.8**	**85.7**	95.5	89.8

TABLE IV: The performance (mAP) comparison with the state-of-the-art in INSTRE-S (%).

Method	Dim.	INSTRE-S	INSTRE-S+1M
Bow [53]	n/a	48.1	9.1
RANSAC [54]	n/a	46.5	7.7
SC [55]	n/a	53.4	12.5
GC [56]	n/a	58.6	19.7
COP [57]	2048	63.8	23.5
HE+WGC [58]	2048	67.2	26.8
L-R-MAC [16]	512	75.6	32.8
Ours	**512**	**79.6**	**34.2**

TABLE V: The performance (mAP) comparison with the state-of-the-art in INSTRE-S with QE (%).

Method	Dim.	INSTRE-S	INSTRE-S+1M
L-R-MAC+QE[16]	512	79.8	35.8
Global Diff.[52]	512	70.3	-
Region Diff.[52]	5×512	77.5	-
Ours+QE	**512**	**82.4**	**38.2**

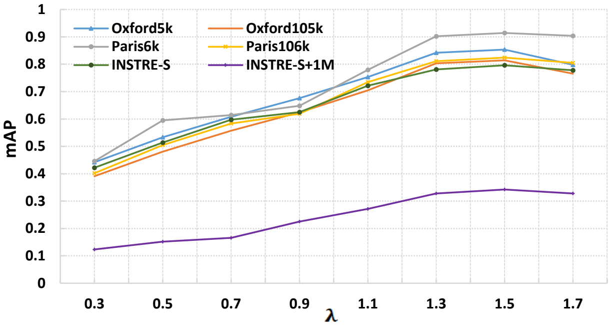


Fig. 6: Performance evaluation for different λ values on different datasets.

methods that also compute the image representations without performing Query Expansion (QE) at test time. Table II shows experimental results on four standard image retrieval benchmarks. Some methods listed in Table II are proposed recently such as Mask-MAC [50] and SBA [49]. We can see that our method achieves the best performance among most of methods on all the standard datasets except the Mask-MAC method on the Pairs106K dataset. However, our method has the lower feature dimensions than this method. Table IV shows the experimental results on the INSTRE-S, which contains various 3D or planar objects from buildings to logos with many variations. We can see that our method consistently outperforms all of them.

We then compare our approach with other methods that rely on both image representation and QE, where QE is a comparatively cheap strategy that significantly increases the final accuracy. Similar to [16], we conduct the experiment with average QE [33], which has a negligible cost (the top 10 returned results are used). Table III and Table V show the experimental results. From Table III, we can see that our method is comparative or even improves the state of the art on most datasets. Note that the performance of our method is higher than recently proposed regional fusion method [52] on two datasets, Oxford5k and Oxford105k. Their higher performance on other two datasets comes from their high-dimensional features. Although they reported higher performance when increasing the feature dimensions, such methods are hardly scalable as they require a lot of storage memory. Table V also shows experimental results on the INSTRE dataset. Note that our experimental setting is different from [52] for the INSTRE-S dataset: we use the INSTRE-S dataset and the experimental setting is the same as [43] while they re-split the whole INSTRE dataset in a different setting.

E. Discussion of latest work

More recent released works, such as [59], [60], [61], [62] tackled the image retrieval task from different angles and achieved better retrieval performance. For example, Radenovic *et al.* [60] solved the image retrieval problem via unsupervised fine-tuning of CNNs on a large image collection in a fully automated manner, where reconstructed 3D models and structure-from-motion methods guide the selection of the training data. Considering local hard negative samples can provide tight constraints to fine tune the metric locally, Zhou *et al.* [59] achieved the optimal local metric adaptation for image retrieval by exploiting easily-available negative samples. Chen *et al.* [62] proposed an end-to-end local similarity learning framework to handle small objects for local feature learning. Gao *et al.* [61] proposed one probabilistic method for cross-region matching based image retrieval. There are also some works, which focus on solving image retrieval tasks under varying illumination conditions [63] or introducing novel losses [64]. In contrast, our method adopts the triplet network with the classification sub-network to learn local image feature representation via metric learning for image retrieval. As the future work, it would be interesting to see what performance would our method reach with ideas from recent works, such as [64], [61], [62].

V. Conclusions

In this paper, we propose a two-stage triplet network training framework for image retrieval. To boost the feature learning capability of CNN, we train the triplet network by simultaneously optimizing both ranking-loss and classification-loss. The added classification sub-network can also lead to faster convergence of the network. With the introduced Regional Generalized Mean Pooling (RGMP) technique, we can effectively pool feature maps of convolution layers from DLRTN, and then aggregate pooled activations of convolutions from different regions of each image to produce global image representation, namely Regional Generalized Activations of Convolutions (R-GAC). We conduct extensive experiment on six datasets, including four standard image retrieval benchmarks and two recently introduced more challenging INSTRE datasets, and have quantitatively evaluated the impact of each component in the framework. The experimental results have demonstrated the effectiveness of the proposed framework.

This work can be extended in the following three directions: 1) We plan to combine RGMP and double-loss in a triplet network and design an end-to-end learning method for this network. We can also introduce another classification layer on negative images to evaluate the performance of our proposed method. 2) We will introduce a state-of-the-art hashing method, e.g., relaxed binary autoencoder [65] into the framework to support large-scale instance-level image retrieval. 3) Recently proposed methods such as [30] and [20] adopt more complex neural architecture (e.g., ResNet [66]), novel multi-scale image representations and query expansion methods [20] to achieve better retrieval performance. Therefore, we also plan to incorporate these modules to our framework to further boost the performance.

References

[1] Z. Zhong, J. Zhu, and S. C. H. Hoi, “Fast object retrieval using direct spatial matching,” *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1391–1397, 2015.

[2] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian, “Democratic diffusion aggregation for image retrieval,” *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1661–1674, 2016.

[3] A. Chadha and Y. Andreopoulos, “Voronoi-based compact image descriptors: Efficient region-of-interest retrieval with vlad and deep-learning-based descriptors,” *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1596–1608, 2017.

[4] S. Pang, J. Ma, J. Zhu, J. Xue, and Q. Tian, “Improving object retrieval quality by integration of similarity propagation and query expansion,” *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 760–770, 2019.

[5] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2018.

[6] Y. Song, X. Chen, X. Wang, Y. Zhang, and J. Li, “6-DOF image localization from massive geo-tagged reference images,” *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1542–1554, 2016.

[7] J. L. Schonberger, F. Radenovic, O. Chum, and J. M. Frahm, “From single image query to detailed 3D reconstruction,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5126–5134.

[8] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[10] A. Bergamo, S. N. Sinha, and L. Torresani, “Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 763–770.

[11] Y. Ke and R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 506–513.

[12] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *European Conference on Computer Vision*. Springer, 2014, pp. 584–599.

[13] F. Radenović, G. Tolias, and O. Chum, “CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples,” in *European Conference on Computer Vision*. Springer, 2016, pp. 3–20.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[15] G. Tolias, R. Sicre, and H. Jégou, “Particular object retrieval with integral max-pooling of CNN activations,” in *International Conference on Learning Representations*, 2016, pp. 1–12.

[16] J. R. Albert Gordo, Jon Almaz’an and D. Larlus, “Deep image retrieval: Learning global representations for image search,” in *European Conference on Computer Vision*. Springer International Publishing, 2016, pp. 241–257.

[17] C. Xu, C. Lu, X. Liang, J. Gao, W. Zheng, T. Wang, and S. Yan, “Multi-loss regularized deep neural network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 12, pp. 2273–2283, 2016.

[18] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, “Visual instance retrieval with deep convolutional networks,” *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016.

[19] A. Babenko and V. Lempitsky, “Aggregating local deep features for image retrieval,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1269–1277.

[20] F. Radenović, G. Tolias, and O. Chum, “Fine-tuning CNN image retrieval with no human annotation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[21] L. Zheng, Y. Yang, and Q. Tian, “SIFT Meets CNN: A decade survey of instance retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[22] Y. Li, X. Kong, L. Zheng, and Q. Tian, “Exploiting hierarchical activations of neural network for image retrieval,” in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 132–136.

[23] F. Yang, J. Li, S. Wei, Q. Zheng, T. Liu, and Y. Zhao, “Two-stream attentive CNNs for image retrieval,” in *Proceedings of the ACM Multimedia Conference*, 2017, pp. 1513–1521.

[24] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519.

[25] Y. Kalantidis, C. Mellina, and S. Osindero, “Cross-dimensional weighting for aggregated deep convolutional features,” in *European Conference on Computer Vision*. Springer, 2016, pp. 685–701.

[26] F. Perronnin and D. Larlus, “Fisher vectors meet neural networks: A hybrid classification architecture,” in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 2015, pp. 3743–3752.

[27] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *European Conference on Computer Vision*. Springer, 2014, pp. 392–407.

[28] K. Yan, Y. Wang, D. Liang, T. Huang, and Y. Tian, “CNN vs. SIFT for image retrieval: Alternative or complementary?” in *Proceedings of the ACM Multimedia Conference*, 2016, pp. 407–411.

[29] S. Mei, W. Min, H. Duan, and S. Jiang, “Instance-level object retrieval via deep region CNN,” *Multimedia Tools Appl.*, vol. 78, no. 10, pp. 13 247–13 261, 2019.

[30] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, “End-to-end learning of deep visual representations for image retrieval,” *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[31] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, “From generic to specific deep representations for visual recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–45.

[32] A. Babenko and V. Lempitsky, “Aggregating deep convolutional features for image retrieval,” in *Proceedings of IEEE International Conference on Computer Vision*, 2015, pp. 1269–1277.

[33] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, “Total recall: Automatic query expansion with a generative feature model for object retrieval,” in *Proceedings of IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[34] X. Shen, Z. Lin, J. Brandt, and Y. Wu, “Spatially-constrained similarity measurefor large-scale object retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1229–1241, 2014.

[35] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[36] X. Ji, W. Wang, M. Zhang, and Y. Yang, “Cross-domain image retrieval with attention modeling,” in *Proceedings of the ACM Multimedia Conference*, 2017, pp. 1654–1662.

[37] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4004–4012.

[38] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *International Workshop on Similarity-Based Pattern Recognition*, 2015, pp. 84–92.

[39] E.-J. Ong, S. Husain, and M. Bober, “Siamese network of deep fisher-vector descriptors for image retrieval,” *arXiv preprint arXiv:1702.00338*, 2017.

[40] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[41] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[42] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in quantization: Improving particular object retrieval in large scale image databases,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[43] S. Wang and S. Jiang, “INSTRE: A new benchmark for instance-level object retrieval and recognition,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 11, no. 3, pp. 37:1–37:21, 2015.

[44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” pp. 675–678, 2014.

[45] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[46] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, “Local convolutional features with unsupervised training for image retrieval,” in *Proceedings of IEEE International Conference on Computer Vision*, 2015, pp. 91–99.

[47] J. Y. H. Ng, F. Yang, and L. S. Davis, “Exploiting local features from deep networks for image retrieval,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 53–61.

[48] E. Mohedano, K. McGuinness, N. E. O’Connor, A. Salvador, F. Marques, and X. Giro-i Nieto, “Bags of local convolutional features for scalable instance search,” in *Proceedings of ACM on International Conference on Multimedia Retrieval*, 2016, pp. 327–331.

[49] J. Xu, C. Wang, C. Qi, C. Shi, and B. Xiao, “Unsupervised semantic-based aggregation of deep convolutional features,” *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 601–611, 2019.

[50] T. Hoang, T.-T. Do, D.-K. Le Tan, and N.-M. Cheung, “Selective deep convolutional features for image retrieval,” in *Proceedings of the ACM Multimedia Conference*, 2017, pp. 1600–1608.

[51] O. Chum, A. Mikulík, M. Perdoch, and J. Matas, “Total recall ii: Query expansion revisited,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 889–896.

[52] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, “Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 926–935.

[53] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1470–1477.

[54] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography,” in *Readings in computer vision*. Elsevier, 1987, pp. 726–740.

[55] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, “Spatial coding for large scale partial-duplicate web image search,” in *Proceedings of the ACM Multimedia Conference*, 2010, pp. 511–520.

[56] W. Zhou, H. Li, Y. Lu, and Q. Tian, “SIFT match verification by geometric coding for large-scale partial-duplicate web image search,” *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 9, no. 1, pp. 4:1–4:18, 2013.

[57] L. Chu, S. Jiang, S. Wang, Y. Zhang, and Q. Huang, “Robust spatial consistency graph model for partial duplicate image retrieval,” *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1982–1996, 2013.

[58] H. Jegou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *European Conference on Computer Vision*. Springer, 2008, pp. 304–317.

[59] J. Zhou and Y. Wu, “Learning visual instance retrieval from failure: Efficient online local metric adaptation from negative samples,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.

[60] F. Radenovic, G. Tolias, and O. Chum, “Fine-tuning cnn image retrieval with no human annotation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2019.

[61] Z. Gao, L. Wang, and L. Zhou, “A probabilistic approach to cross-region matching-based image retrieval,” *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1191–1204, 2019.

[62] Z. Chen, Z. Kuang, W. Zhang, and K. K. Wong, “Learning local similarity with spatial relations for object retrieval,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1703–1711.

[63] T. Jenicek and O. Chum, “No fear of the dark: Image retrieval under varying illumination conditions,” in *The IEEE International Conference on Computer Vision*, 2019.

[64] J. Revaud, J. Almazan, R. S. Rezende, and C. R. d. Souza, “Learning with average precision: Training image retrieval with a listwise loss,” in *The IEEE International Conference on Computer Vision*, October 2019.

[65] T. T. Do, D. K. L. Tan, T. T. Pham, and N. M. Cheung, “Simultaneous feature aggregating and hashing for large-scale image search,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4217–4226.

[66] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.