This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning Deep Local Features with Multiple Dynamic Attentions for Large-Scale Image Retrieval

Hui Wu[1] Min Wang[2*] Wengang Zhou[1,2*] Houqiang Li[1,2]
[1]CAS Key Laboratory of Technology in GIPAS, EEIS Department, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
wh241300@mail.ustc.edu.cn, wangmin@iai.ustc.edu.cn, {zhwg,lihq}@ustc.edu.cn

Abstract

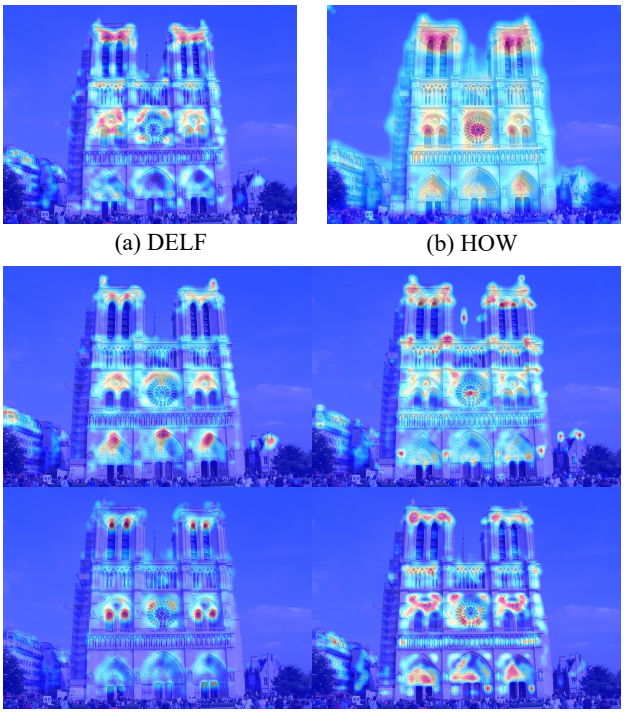
In image retrieval, learning local features with deep convolutional networks has been demonstrated effective to improve the performance. To discriminate deep local features, some research efforts turn to attention learning. However, existing attention-based methods only generate a single attention map for each image, which limits the exploration of diverse visual patterns. To this end, we propose a novel deep local feature learning architecture to simultaneously focus on multiple discriminative local patterns in an image. In our framework, we first adaptively reorganize the channels of activation maps for multiple heads. For each head, a new dynamic attention module is designed to learn the potential attentions. The whole architecture is trained as metric learning of weighted-sum-pooled global image features, with only image-level relevance label. After the architecture training, for each database image, we select local features based on their multi-head dynamic attentions, which are further indexed for efficient retrieval. Extensive experiments show the proposed method outperforms the state-of-the-art methods on the Revisited Oxford and Paris datasets. Besides, it typically achieves competitive results even using local features with lower dimensions. Code will be released at https://github.com/CHANWH/MDA.

1. Introduction

Given a large image corpus, instance image retrieval [32, 27, 17, 38, 18, 9, 10, 19] aims to effectively identify images containing the same object or describing the same scene as the query image. This task is challenging due to the various conditions observed in large-scale datasets, such as lighting variation, occlusion, viewpoint changes, *etc*. To this end, many research efforts are devoted to image representation with descriptive and discriminative local features.

Given an image, the extraction of local feature usually involves local region detection and image patch description, where the former identifies salient regions of interest in the image which is further described into a vector of predefined dimension by the latter. Before the advent of deep learning, local visual features are designed in a hand-crafted way [20, 2], which is also regarded as a kind of shallow feature. With the introduction of deep learning to computer vision, tremendous progress [35, 21, 7, 6, 37, 23] has been made on local feature learning in a data-driven paradigm.

Early works on deep local feature ignores the local region detection. Some works [35, 21, 7] assume the im-



(a) DELF (b) HOW (c) Ours

Figure 1. Attention maps of three methods. (a) and (b): attention maps generated by DELF [23] and HOW [37]. (c): attention maps generated by our method. HOW and DELF only generate a single attention map focusing on limited patterns, while our method focuses on diverse patterns by generating multiple attention maps.

*Corresponding Author: Min Wang and Wengang Zhou.

age patch is ready from hand-crafted detectors and focus on learning a vector from an image patch with CNNs in a supervised learning way. Some other works [32, 38, 1, 19, 9, 27] directly take the activation map from a convolution layer and regard the channel features in each spatial position of the map as a local feature. Such local feature essentially corresponds to a relatively fixed receptive field in the input image. Recently, more and more research efforts [37, 23, 4, 34] resort to attention learning to discriminate deep local features with only image-level annotation. In those methods, a single attention map is typically extracted to measure the significance of each deep local feature, as illustrated in Fig. 1. Generally, an attention map corresponds to some semantic-aware visual pattern. Considering the diverse content in an image, a single attention map is unlikely to comprehensively capture all potential semantic patterns in an image.

To address the above issue, we propose a new framework with multiple dynamic attentions to detect diverse local features corresponding to different semantic patterns. In our method, we make use of intermediate feature maps from CNNs to generate attention maps. To decouple different semantic patterns, we introduce a channel mapping layer to adaptively reorganize the channels of input feature maps into multiple groups, each of which is fed to an attention head. In the attention head, we design a new dynamic attention module. Specially, we introduce a diversity regularization to ensure different heads focus on different patterns within the image. In the training stage, we perform attention pooling with the feature map and the multiple attention maps to generate a set of global representations and the whole network is optimized with image-level label. In the testing stage, we make use of those dynamic attention maps to select deep local features to represent each image. To achieve efficient retrieval on large image database, we quantize those deep local features with a codebook and match images with binarized aggregated match kernel [37].

Compared with previous attention-based local feature leaning methods, our approach is able to discover more diverse and distinctive local patterns, which favorably benefit the semantic content matching between images. We evaluate our approach on the Revisited Oxford and Paris datasets, which are further mixed with a million distractors. Ablation studies justify the effectiveness of the channel mapping layer, dynamic attention module, and diversity regularization loss. Our approach achieves superior performance over the existing state-of-the-art methods under similar setting.

2. Related Work

2.1. Local feature learning

Traditional local features [20, 2] are extracted with hand-crafted detectors and descriptors based on low-level visual information, which limits the discriminative power of local features. More recently, many methods have been proposed to learn the descriptors or detectors with deep neural networks. Savinov *et al*. [30] train a Quad-network for keypoint detection. Tian *et al*. [35] introduce second-order similarity to improve patch descriptors for image matching. Mishchuk *et al*. [21] propose a loss which maximizes the distance between the closest positive and closest negative example in the batch to learn powerful descriptors. Recently, D2-Net [8] and SuperPoint [6] propose an end-to-end framework to detect keypoints and compute descriptors. Although convincing results have been reported, these methods are not designed for the image retrieval task. They focus on image matching or image registration while our work focuses on large-scale image retrieval where memory requirements matter.

The most relevant works to ours come from [23, 34, 37]. Noh *et al*. [23] first explore learning local features for image retrieval with visual attention and image-level annotation. The locations with the strongest values on the attention map are selected, while the descriptors are slices of the feature map at selected locations. Based on DELF, Teichmann *et al*. [34] introduce region proposal network [28] (RPN) to detect objects presented in the image. They only consider local features in the region of interest and filter out irrelevant features. Tolias *et al*. [37] propose a non-parametric attention module and integrate dimensionality reduction into the network, which is trained by contrastive loss. Compared with the above methods, our work generates multiple disentangled attention maps independently for a single image. This prevents the model from focusing only on limited patterns of the image, thereby ignoring some patterns that may be helpful for image retrieval.

2.2. Local Feature Aggregation

Due to a large amount of memory required to store local features, many methods have been proposed to aggregate local features into a compact descriptor, such as BOW [33], Hamming Embedding [13], VLAD [14], FV [15] Triangulation [16] and ASMK [36]. Recently, some novel deep local feature aggregation methods are proposed to reorganize deep local features into a compact descriptor. Tolias *et al*. [38] use sliding windows to extract regional information, and then max-pooling is applied to the activation of each region. Babenko and Lempitsky [1] first propose to use sum-pooling, which performs well due to the subsequent descriptor whitening. Kalantidis *et al*. [19] apply channel and spatial attention on the activation before sum pooling. After that, GeM-pooling has been shown to give excellent results when trained with ArcFace loss [5]. It allows more than one position in the activation contributing to the aggregated representation, while still being more selective than the sum-pooling. The main advantage of these

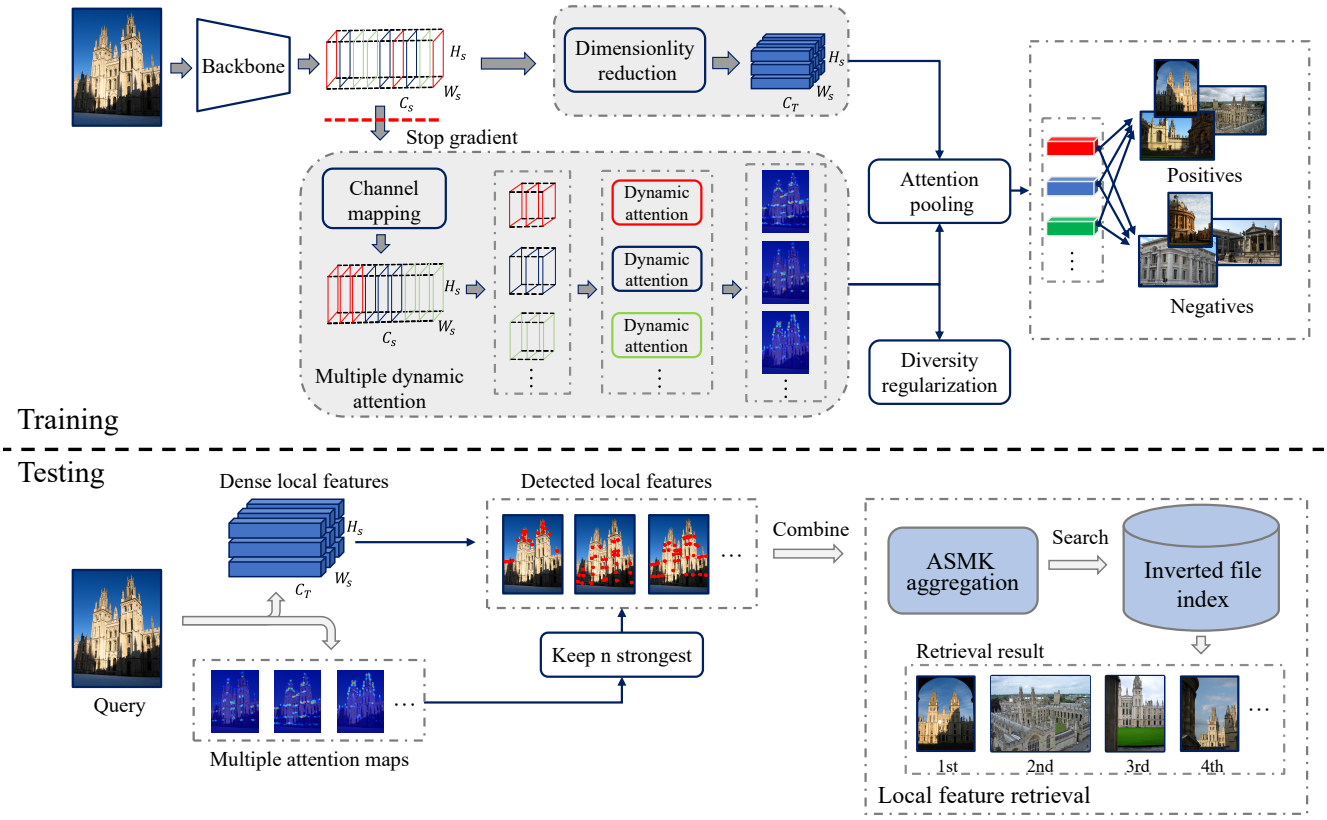


Figure 2. Framework of our method. The components highlighted in gray are used during testing and training. The whole framework is trained with contrastive loss and diversity regularization to generate multiple discriminative deep local features. Gradient back-propagation is stopped from the multiple dynamic attention into the CNN backbone. During testing, the top n strongest local features, according to the multiple attention maps, are kept to represent the image. Local features are aggregated with binarized ASMK [36] for efficient retrieval.

aggregation methods is that they provide high-performance image retrieval with a small memory footprint. In this work, we adopt the binarized ASMK as the aggregation method for local features.

2.3. Dimensionality Reduction and Whitening

Dimensionality reduction is widely used on image features in image retrieval since the memory footprint is critical for retrieval systems. Features with high dimensionality cost a large amount of memory, further slowing down online retrieval. On the other hand, as shown in [12], whitening reduces the co-occurrence of local features, which is beneficial for retrieval. Recently, Gordo *et al*. [9] propose to replace PCA/whitening with a fully-connected layer, which is learned during training. Tolias *et al*. [37] follow [9] in using a fully-connected layer to perform dimensionality reduction and whitening, but the parameters of their fully connected layer are learned using PCA and remain constant during training. Cao *et al*. [4] use an autoencoder to reduce the dimensionality of the extracted local features, and their autoencoder can also be trained end-to-end. In this work, we use the 1×1 convolutional layer to realize dimensionality reduction on the local features, with parameters initialized by the results of PCA and further updated during training.

3. Method

3.1. Framework

The training and testing pipelines of our approach are illustrated in Fig. 2. Given an image, we apply a CNN backbone to obtain a feature map $\mathcal{S} \in \mathcal{R}^{H_S \times W_S \times C_S}$, where H_S, W_S, C_S correspond to the height, width and number of channels. This feature map can be seen as a set of local descriptors. To select relevant patterns for image retrieval, we develop a new multiple dynamic attention (MDA) module M with N heads to predict which local features are discriminative for the objects of interest. Each head in MDA detects a certain pattern, while different heads focus on different patterns. M first adaptively reorganizes $\mathcal{S}$ with a channel mapping layer and then divides it into different groups uniformly. Each group is fed into the attention generation module to learn the attention maps independently. Since thousands of local features will be detected from an image, they must be compact to save memory cost. To this

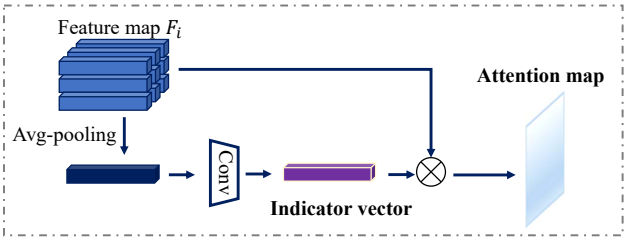


Figure 3. An overview of our dynamic attention module. The indicator vector varies with different instances.

end, we use a convolutional layer for dimension reduction. The local descriptors are obtained as $\mathcal{L} = T(\mathcal{S})$, where $\mathcal{L} \in \mathcal{R}^{H_S \times W_S \times C_T}$, and T is a 1×1 convolutional layer with C_T filters.

During training, we obtain a set of global descriptors by pooling the local descriptors with each attention map. These global descriptors will be used to train the whole network only with image-level annotations. Specifically, we use a diversity regularization term to force different heads to focus on different patterns within the image. During testing, the top n strongest descriptors l from $\mathcal{L}$ are kept according to the importance in multiple attention maps. These local descriptors are used for large-scale image retrieval combined with binarized ASMK [36].

3.2. Multiple Dynamic Attention Module

To obtain attention maps that capture different patterns, we devise a multiple dynamic attention module. The feature maps of a convolutional neural network can be interpreted as a collection of 2D response maps of pattern detectors. However, feature channels corresponding to the same visual pattern are often not arranged in order. Given a feature map $\mathcal{S}$ generated by backbone, we first use a 1×1 convolutional channel mapping layer to reorganize the channels:

$$\hat{\mathcal{S}} = Conv_{1\times1}(\mathcal{S}), \tag{1}$$

where $\hat{\mathcal{S}} \in \mathcal{R}^{H_S \times W_S \times C_S}$. Then, we divide the feature map $\hat{\mathcal{S}}$ into N different groups to represent multiple independent patterns. The collection of the groups is represented by $\mathcal{F} = \{F^1, F^2, \cdots, F^N\}$, where F^i corresponds to channel $\left[(i-1) \times \lfloor \frac{C_S}{N} \rfloor + 1, i \times \lfloor \frac{C_S}{N} \rfloor\right]$ of $\hat{\mathcal{S}}$. Finally, by feeding each $F^i \in \mathcal{F}$ into the attention generation module, N pieces of attention maps are generated. We denote the set $\mathcal{A} = \{A^{(1)}, A^{(2)}, \cdots, A^{(N)}\}$ as the generated collection of attention maps, where $A^{(i)} \in \mathcal{R}^{H_S \times W_S}$.

Fig. 3 illustrates the attention generation module, termed as **Dynamic Attention Head (DA)**. Given an intermediate feature map $F^i \in \mathcal{R}^{H_S \times W_S \times \lfloor \frac{C_S}{N} \rfloor}$ as input, DA first aggregates spatial information of the feature map using average pooling , generating a spatial context descriptors F^i_{avg}. This descriptor is then fed to a 1×1 convolutional layer to produce our indicator vector $F^i_g \in \mathcal{R}^{1\times1\times\lfloor \frac{C_S}{N} \rfloor}$. In short, the indicator vector is computed as:

$$\begin{aligned} F^i_g &= \sigma(Conv_{1\times1}(AvgPool(F^i))) \\ &= \sigma(\mathbf{W_g} * F^i_{avg}), \end{aligned} \tag{2}$$

where σ denotes ReLU function, and $\mathbf{W}_g \in \mathcal{R}^{\lfloor \frac{C_S}{N} \rfloor \times \lfloor \frac{C_S}{N} \rfloor}$ is learned during training. Then, the attention map can be generated as:

$$A^{(i)} = \text{Softplus}(F^i_g \otimes F^i), \tag{3}$$

where $A^{(i)} \in \mathcal{R}^{H_S \times W_S}$ and $\otimes$ denotes element-wise dot product. During multiplication, the indicator vector is broadcasted along the spatial dimension. Softplus is a smooth approximation to the ReLU function and can be used to constrain the output to always be positive.

In all, our MDA module produces multiple attention maps, each of them focuses on one specific pattern. Multiple attention maps capture different patterns. As shown in Fig. 1, our model focuses on many important and distinguishable patterns within the image, some of which are ignored by other methods.

3.3. Training Stage

To extract multiple discriminative and repeatable deep local feature regions for each image, we formulate two optimization objectives, *i.e.*, contrastive loss and diversity regularization, to optimize the whole network.

Given image pairs (i, j) and labels $Y(i, j) \in \{0, 1\}$ in which label 1 denotes the image pair is matched, 0 otherwise. We obtain a set of global descriptors $\mathcal{G} = \{G^{(1)}, G^{(2)}, \cdots, G^{(N)}\}$ for each image by pooling the local descriptors $\mathcal{L}$ with each attention map in $\mathcal{A}$:

$$G^{(i)} = \sum_{h \in H_S, w \in W_S} A^{(i)}_{h,w} \mathcal{L}_{h,w}, \text{ for } i = 1, 2, \cdots, N, \tag{4}$$

each corresponding to a certain head. Contrastive loss is applied to the global descriptor of each head separately. The loss summation for all heads can be defined as follows:

$$\begin{aligned} L_C = &\ (1 - Y(i,j)) \sum_{n=1}^{N} \max\left(m - \left\|\hat{G}_i^{(n)} - \hat{G}_j^{(n)}\right\|_2\right)^2 \\ &+ Y(i,j) \sum_{n=1}^{N} \left\|\hat{G}_i^{(n)} - \hat{G}_j^{(n)}\right\|_2^2, \end{aligned} \tag{5}$$

where m is the margin, N is the head number and $\hat{G}_i^{(n)}$ is the l_2-normalized global descriptor $G_i^{(n)}$. It enforces the corresponding global descriptor of each head to be highly distinguishable to discriminate whether the given image pair is matched or not.

For a given image, there is no guarantee that the attention map generated by one attention head is different from that of another head. Therefore, multiple attention heads may focus on the same pattern within image. We need to ensure

that different heads focus on different patterns of the given image to maximize the pattern information contained in the final descriptors. In other words, we should maximize the distance between any two attention maps.

For each attention map $A_i \in \mathcal{A}$, we first flatten it into a vector $a_i \in \mathcal{R}^{H_S W_S}$ and apply the softmax function on it to make sure that the attention scores at all positions are positive and add up to one, as follows:

$$\hat{a}_i = \text{softmax}(a_i), \text{ for } i = 1, 2, \cdots, N. \tag{6}$$

Then, the Hellinger distance [3] between $\hat{a}_i \in \mathcal{R}^{H_S W_S}$ and $\hat{a}_j \in \mathcal{R}^{H_S W_S}$ is defined as:

$$H(\hat{a}_i, \hat{a}_j) = \frac{1}{\sqrt{2}} \left\| \sqrt{\hat{a}_i} - \sqrt{\hat{a}_j} \right\|_2. \tag{7}$$

Since $\sum_{l=1}^{H_S W_S} \hat{a}_{i,l} = 1$:

$$H(\hat{a}_i, \hat{a}_j)^2 = 1 - \left\langle \sqrt{\hat{a}_i}, \sqrt{\hat{a}_j} \right\rangle, \tag{8}$$

where $\left\langle \sqrt{\hat{a}_i}, \sqrt{\hat{a}_j} \right\rangle$ represents the dot product between vector $\sqrt{\hat{a}_i}$ and $\sqrt{\hat{a}_j}$.

To increase the diversity of the attention maps, we should maximize the distance between each $\hat{a}_i$ and $\hat{a}_j$. Thus, the regularization term to measure the similarities between attention maps is defined as:

$$L_{reg} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j \neq i}^{N} \left(\left\langle \sqrt{\hat{a}_i}, \sqrt{\hat{a}_j} \right\rangle - 1 \right). \tag{9}$$

The final objective function is a combination of contrastive loss and diversity regularization balanced by λ, as follows:

$$L = L_C + \lambda L_{reg}. \tag{10}$$

3.4. Testing Stage

During testing, local features are selected according to the importance given by $\mathcal{A}$. Attention values from all heads are ranked jointly, and the local features corresponding to the top n largest attention values are returned. Fig. 4 shows some examples of local features detected by different attention heads, which are well aligned in two views of a sample landmark. Furthermore, multi-scale extraction is performed during testing. We combine local features from all scales as the local descriptor set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$. Binarized aggregated match kernel framework [36] (ASMK*) is adopted to aggregate local features and compute similarities between images. The aggregated vectors are further indexed for efficient retrieval.

In ASMK* framework, an image X is described by a set of n C_T-dimensional local descriptors $\mathcal{X}$. These descriptors are quantized by a codebook $\mathcal{C}$ with C visual words, learned using k-means. Denote $\mathcal{X}_c = \{x \in \mathcal{X} : q(x) = c\}$ as the subset of descriptors in X which are assigned to visual word c. The similarity between two images X and Y can be computed as:

$$K(X, Y) = \gamma(\mathcal{X})\gamma(\mathcal{Y}) \sum_{c \in \mathcal{C}} \sigma_\alpha(\Phi(\mathcal{X}_c)^T \Phi(\mathcal{Y}_c)), \tag{11}$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote local features sets of X and Y respectively. $\Phi(\mathcal{X})$ is an aggregated vector representation, $\sigma_\alpha(\cdot)$ denotes a scalar selectivity function and $\gamma(\mathcal{X})$ is the normalization factor. See [36] for the detailed definitions of above formulations.

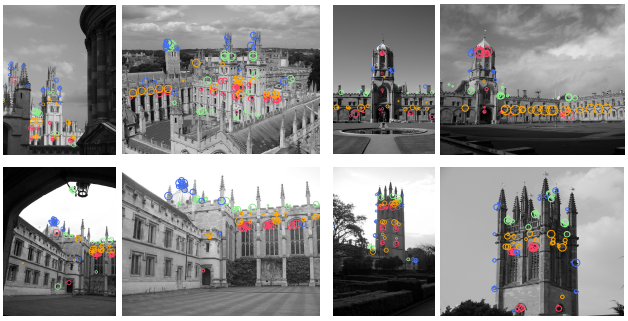


Figure 4. Qualitative examples of local features detected by four different attention heads on four pairs of matched images in $\mathcal{R}$Oxf dataset [26]. Features detected by same head are shown in the same color; only top 25 local features of each head are plotted.

4. Experiment

4.1. Experimental Setup

Training dataset. The training dataset *SfM120k* [27] is used. It is obtained by 3D reconstruction of large collections of unlabelled images [31]. Given a large unannotated image collection, images are clustered and a 3D model is constructed per cluster. Matching pairs (anchor-positive) are formed by images with a sufficient number of co-observed 3D points (same 3D model). Non-matching pairs (anchor-negative) come from different 3D models. We use 551 3D models for training and 162 for validation.

Evaluation datasets and metrics. $\mathcal{R}$Oxf, $\mathcal{R}$Par, $\mathcal{R}$Oxf + $\mathcal{R}$1M, and $\mathcal{R}$Par + $\mathcal{R}$1M are used to evaluate our method. The $\mathcal{R}$Oxf and $\mathcal{R}$Par [26] datasets are the revisited version of the original Oxford5k [24] and Paris6k datasets [25]. Both datasets contain 70 query images, and additionally include $4,993$ and $6,322$ database images respectively. Mean Average Precision (mAP) is used to evaluate the performance. There are three evaluation setups for these two datasets: Easy, Medium, and Hard. We report the Medium and Hard performance of $\mathcal{R}$Oxf and $\mathcal{R}$Par. Large-scale results are further reported with the $\mathcal{R}$1M distractors [26], which contains 1M images.

Implementation details. We use ResNet50 as backbone network, initialized by pre-training on ImageNet [11]. We obtain the feature map $\mathcal{S}$ from the *Conv4* output. Following

Tolias *et al*. [37], we also perform 3×3 average pooling before dimensionality reduction.

For training, each mini-batch contains 5 tuples, where a tuple consists of 7 images (1 query, 1 positive, and 5 negative images). Before each epoch, we randomly choose 2000 anchor-positive pairs and $20,000$ candidate negative images. Hard-negative mining is performed to select 5 most difficult negative samples for each anchor-positive pair. We resize training images with the larger dimension equal to 921 pixels, preserving the aspect ratio. The model is optimized using Adam with weight decay equal to 10^{-6}, and an exponentially decaying learning rate with a decay rate of 0.99. The initial learning rate is set as 10^{-5} and $5 * 10^{-5}$ for the backbone and attention module, respectively. We set the margin $m = 0.9$, the weight for L_{reg} to $\lambda = 0.3$ and local feature dimension $C_T = 128$. We conduct experiments with head numbers within $\{1, 2, 4, 6, 8, 10, 12\}$ and report results for the best performing one.

For testing, the default ASMK* configuration from Tolias *et al*. [36] is adopted. We use a codebook of size $C = 65,536$. The codebook is learned on the local descriptors of the 5000 training image extracted at the original scale. Images are resized to the same size as training. We use 7 different scales, ranging from 0.25 to 2.0 and increasing by a factor of $\sqrt{2}$, to construct the image pyramids. These pyramids are fed to the network, and the resulting local features from all the scales are combined. The top 2000 strongest local descriptors are selected for each image. When searching online, each local feature of the query image is assigned to 5 nearest visual words. We use the inverted index to speed up the online search. Specifically, using k-means for codebook creation may lead to the randomness of results. We run each experiment 5 times and report the mean and standard deviation. In large-scale experiments, multiple visual word assignments are not performed to reduce the computational cost of online retrieval. Unless otherwise stated, the default configuration is used.

4.2. Ablation Experiments

Head numbers. Fig. 5 shows the mAP of our method on the Hard and Medium protocols of $\mathcal{R}$Oxf-$\mathcal{R}$Par for N within $\{1, 2, 4, 6, 8, 10, 12\}$. The mAP increases as N is raised from 1 to 8, then drops when $N \geq 10$. When $N = 1$, our method achieves average performance equal to 57.8 and 64.2 in terms of mAP on Hard protocol for $\mathcal{R}$Oxf-$\mathcal{R}$Par, which outperforms HOW by 0.9 and 1.4 respectively. It proves that our DA can learn better local features.

Attention generation module. As shown in Fig. 6, the attention module of DELF [23] and HOW [37] can be regarded as the dot product of the indicator vector and the current feature map. In DELF, the attention map is simply generated by two 1×1 convolutional layers. The first convolutional layer is used for dimensionality reduction, and the

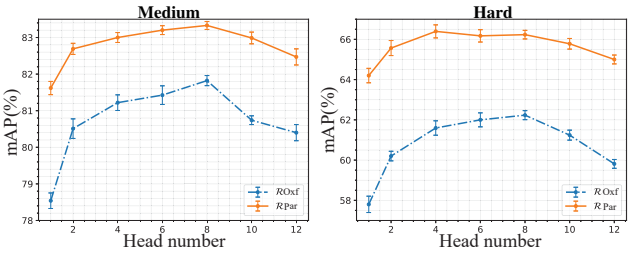


Figure 5. Comparison of mAP against head num on $\mathcal{R}$Oxf-$\mathcal{R}$Par.

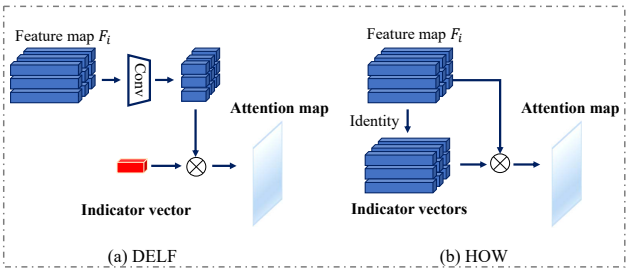


Figure 6. Comparisons of different attention modules. DELF [23] uses a 2-layer CNN as the attention module, and the indicator vector is the weight of the second convolutional layer, which is learned during training. HOW [37] uses l_2 norm of the feature descriptor as the attention score, thus the indicator vectors are the current feature descriptors.

Attention Module	Stop Gradient	Medium		Hard	
		$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
1×1 conv [23]	✓	80.5 ± 0.2	82.7 ± 0.1	60.6 ± 0.3	65.4 ± 0.3
l_2 norm [37]	✓	80.9 ± 0.2	81.5 ± 0.1	60.5 ± 0.3	64.0 ± 0.2
1×1 conv + l_2 norm	✓	80.4 ± 0.1	82.3 ± 0.2	60.5 ± 0.2	64.8 ± 0.3
DA (ours)	✗	80.5 ± 0.3	82.1 ± 0.2	60.3 ± 0.6	64.0 ± 0.1
DA (ours)	✓	$\mathbf{81.8 \pm 0.3}$	$\mathbf{83.3 \pm 0.2}$	$\mathbf{62.2 \pm 0.5}$	$\mathbf{66.2 \pm 0.2}$

Table 1. Impact of the attention generation module on mAP on $\mathcal{R}$Oxf and $\mathcal{R}$Par [26]. Results with head num $N = 8$, feature num $n = 2000$ and local dimension $C_T = 128$.

indicator vector is the weight of the second convolutional layer, which can be interpreted as a generalized feature of the foreground. It outputs similarly for various foreground features. However, foreground objects are diverse in different images. Only using a fixed indicator vector limits the discriminative power of local features. For HOW, the indicator vectors are equal to the current feature map. The l_2 norm of the corresponding local feature is directly used as attention value, which is also sub-optimal for extracting discriminative local features.

In Tab. 1, we compare the attention generation module of different methods. 1×1 conv and l_2 norm denote the attention generation methods of DELF and HOW, as shown in Fig. 6. Besides, we simply combine the attention maps generated by these two approaches (1×1 conv + l_2 norm). DA is the proposed attention generation module that dynamically generates different indicator vectors for each image, as discussed in Section 3.2. It can be seen that our DA head outperforms 1×1 conv and l_2 norm.

Stop gradient. The last two rows of Tab. 1 shows that

naively optimizing the whole network leads to suboptimal results because the attention module is trained from scratch, which may produce wrong gradients in the early stages of training. We address this issue by stopping gradient back-propagation from the attention branch to the network backbone. This means that the network backbone is optimized solely based on L_C, and produces the desired distinctive feature representation. As shown in Tab. 1, controlling the gradient allows for better training of the network.

Channel Mapping	Diversity Regularization	Medium		Hard	
		$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
	✓	79.7 ± 0.3	80.1 ± 0.2	59.4 ± 0.6	61.2 ± 0.2
✓		80.3 ± 0.1	82.2 ± 0.1	60.2 ± 0.2	64.7 ± 0.2
✓	✓	$\mathbf{81.8 \pm 0.3}$	$\mathbf{83.3 \pm 0.2}$	$\mathbf{62.2 \pm 0.5}$	$\mathbf{66.2 \pm 0.2}$

Table 2. Ablation studies on each component corresponding to the diversity of attention maps. We gradually remove the channel mapping layer and diversity regularization in MDA. All models are trained with head num $N = 8$, local dim $C_T = 128$ and we extract $n = 2000$ local features for each image.

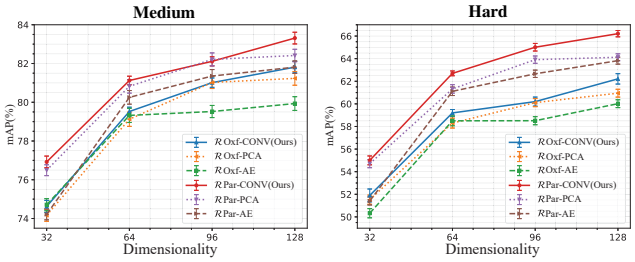


Figure 7. Comparison of local features, trained separately or jointly, with different methods for dimensionality reduction.

Diversity of attention maps. To differentiate the distributions of attention maps, we introduce a channel mapping layer and diversity regularization. As shown in Tab. 2, without the channel mapping layer, diversity regularization alone yields unsatisfactory results. We attribute the loss to feature channels that respond to certain types of visual patterns are usually not arranged in order. If we simply divide the channel into several parts, each head will fail to learn different attention maps. Without diversity regularization, each head may generate similar attention maps and our MDA module may degrade to a single attention module. By jointly employing both of these two components, our method obtains obvious performance improvements.

Some qualitative results of the attention maps are shown in Fig. 8, where each row is the attention maps generated by the same attention head. It can be seen that different heads focus on different patterns in the image and each head focuses on a specific pattern, which implicitly enables pattern-alignment. For example, the first head focuses on the upper edge of the window, while the third head focuses on the sharp bumps and contours.

Dimensionality reduction. We compare the influence of feature dimension when local features are trained with different attention modules. The performance for varying descriptor dimensions is shown in Tab. 3. Our method works better in all dimensions than DELF and HOW.

Method	Local Dim	Medium		Hard	
		$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
DELF [23]†	32	72.2 ± 0.3	75.2 ± 0.1	48.2 ± 0.5	51.3 ± 0.2
	64	76.6 ± 0.3	80.1 ± 0.2	54.8 ± 0.4	60.7 ± 0.3
	96	78.5 ± 0.1	81.2 ± 0.2	56.6 ± 0.3	62.4 ± 0.3
HOW [37]	32	72.5 ± 0.3	72.7 ± 0.4	49.6 ± 0.4	50.1 ± 0.3
	64	76.4 ± 0.3	79.1 ± 0.2	54.1 ± 0.3	59.6 ± 0.3
	96	78.5 ± 0.2	80.2 ± 0.3	56.4 ± 0.2	61.4 ± 0.2
Ours	32	74.6 ± 0.3	76.9 ± 0.3	51.9 ± 0.6	55.0 ± 0.4
	64	79.5 ± 0.2	81.1 ± 0.2	59.2 ± 0.3	62.7 ± 0.2
	96	81.0 ± 0.3	82.1 ± 0.3	60.2 ± 0.4	65.0 ± 0.2

Table 3. The effect of local feature dimension. DELF marked by † is an improvement of the original method: dimensionality reduction is no longer used as post-processing but is integrated into the network for end-to-end training and contrastive loss is used.

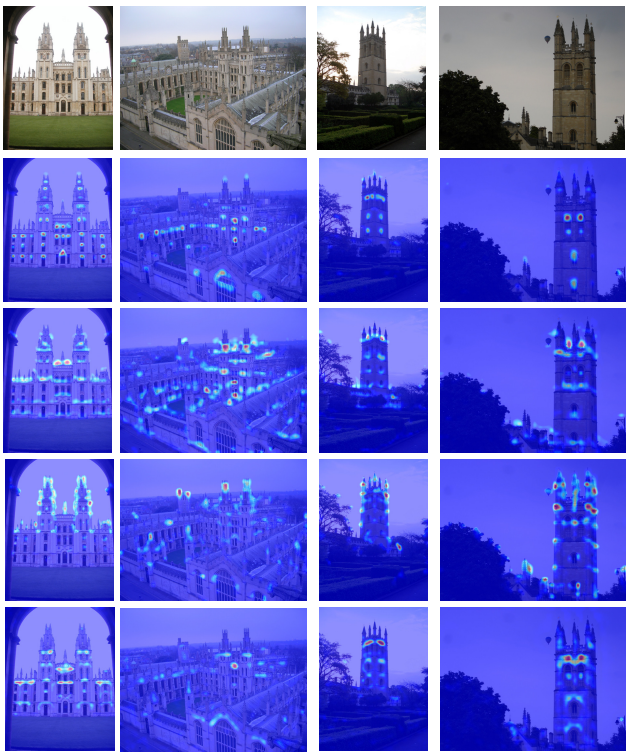


Figure 8. Qualitative examples of multiple dynamic attention maps on the $\mathcal{R}$Oxf datasets [26]. Each column depicts the source image and four corresponding attention maps obtained for four randomly selected heads.

In Fig. 7, we present ablation experiments on different types of dimensionality reduction. First, we use a 1×1 convolutional layer to reduce the dimensionality, with parameters learned during training (CONV). Second, dimensionality reduction is used as a post-processing, which is denoted as PCA. High-dimensional features are used during training and PCA is used to reduce the dimensionality of the features at the end of training. Third, we jointly train an autoencoder (AE) for dimensionality reduction. It consists of two convolutional layers, with the first layer reducing the dimension of

Method	Loss	Train Set	Memory(GB)	Medium				Hard			
				$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+$\mathcal{R}$1M	$\mathcal{R}$Par	$\mathcal{R}$Par+$\mathcal{R}$1M	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+$\mathcal{R}$1M	$\mathcal{R}$Par	$\mathcal{R}$Par+$\mathcal{R}$1M
Compact global descriptors											
R101-R-MAC [9]	Triplet	NC-clean	7.6	60.9	39.3	78.9	54.8	32.4	12.5	59.4	28.0
R101-GeM [27]	Triplet	SfM-120k	7.6	64.7	45.2	77.2	52.3	38.5	19.9	56.3	24.7
R101-GeM↑ [32]	Triplet	SfM-120k	7.6	65.3	46.1	77.3	52.6	39.6	22.2	56.6	24.8
R101-GeM-AP [29]	AP	NC-clean	7.6	67.5	47.5	80.1	52.5	42.8	23.2	60.5	25.1
R101-GeM-AP [29]	AP	GLDv1-noisy	7.6	66.3	-	80.2	-	42.5	-	60.8	-
R101-GeM+SOLAR [22]	Triplet/SOS	GLDv1-noisy	7.6	69.6	53.5	81.6	59.2	47.9	29.9	64.5	33.4
R50-DELG [4]	ArcFace	GLDv1-noisy	7.6	69.7	55.0	81.6	59.7	45.1	27.8	63.4	34.1
R101-DELG [4]	ArcFace	GLDv1-noisy	7.6	73.2	54.8	82.4	61.8	51.2	30.3	64.7	35.5
R50-DELG [4]	ArcFace	GLDv2-clean	7.6	73.6	60.6	85.7	68.6	51.0	32.7	71.5	44.4
R101-DELG [4]	ArcFace	GLDv2-clean	7.6	**76.3**	**63.7**	**86.6**	**70.6**	**55.6**	**37.5**	**72.4**	**46.9**
Local features aggregation with binarized ASMK											
HesAff–rSIFT-ASMK* [26], $C_T = 128$	-	-	62.0	60.4	45.0	61.2	42.0	36.4	25.7	34.5	16.5
¯R50-DELF-ASMK*+SP [26], $n = 1000$, $C_T = 128$	CE	GLDv1-noisy	9.2	67.8	53.8	76.9	57.3	43.1	31.2	55.4	26.4
¯R50-DELF-D2R-R-ASMK* [34], $n = 1000$, $C_T = 128$	CE	GLDv1-noisy	27.6	73.3	61.0	80.7	60.2	47.6	33.6	61.3	29.9
¯R50-DELF-D2R-R-ASMK*+SP [34], $n = 1000$, $C_T = 128$	CE	GLDv1-noisy	27.6	76.0	64.0	80.2	59.7	52.4	38.1	58.6	29.4
¯R50-HOW-ASMK* [37], $n = 1000$, $C_T = 128$	Contrastive	SfM-120k	7.6	78.3	63.6	80.1	58.4	55.8	36.8	60.1	30.7
¯R50-HOW-ASMK* [37], $n = 1200$, $C_T = 128$	Contrastive	SfM-120k	9.2	78.8	64.5	80.6	59.6	56.7	37.7	61.0	31.7
¯R50-HOW-ASMK* [37], $n = 1400$, $C_T = 128$	Contrastive	SfM-120k	10.6	79.1	64.9	81.0	60.4	56.8	38.2	61.5	32.6
¯R50-HOW-ASMK* [37], $n = 2000$, $C_T = 128$	Contrastive	SfM-120k	14.3	79.4	65.8	81.6	61.8	56.9	38.9	62.4	33.7
¯R50-MDA-ASMK* (Ours), $n = 2000$, $C_T = 64$	Contrastive	SfM-120k	7.4	79.5	65.6	81.1	61.3	59.2	42.6	62.7	35.0
¯R50-MDA-ASMK* (Ours), $n = 1000$, $C_T = 128$	Contrastive	SfM-120k	7.6	81.5	66.5	81.8	61.8	61.9	43.1	63.7	36.1
¯R50-MDA-ASMK* (Ours), $n = 1200$, $C_T = 128$	Contrastive	SfM-120k	8.9	81.9	67.5	82.3	62.9	62.3	43.7	64.6	37.1
¯R50-MDA-ASMK* (Ours), $n = 1400$, $C_T = 128$	Contrastive	SfM-120k	10.1	**82.0**	67.9	82.7	63.6	**62.6**	44.6	65.2	37.8
¯R50-MDA-ASMK* (Ours), $n = 2000$, $C_T = 128$	Contrastive	SfM-120k	13.4	81.8	**68.7**	**83.3**	**64.7**	62.2	**45.3**	**66.2**	**38.9**

Table 4. mAP comparison against retrieval state-of-the-art methods on the $\mathcal{R}$Oxford ($\mathcal{R}$Oxf) and $\mathcal{R}$Paris ($\mathcal{R}$Par) datasets (and their large-scale extensions $\mathcal{R}$Oxf+$\mathcal{R}$1M and $\mathcal{R}$Par+$\mathcal{R}$1M), with Medium and Hard evaluation protocols. Memory is reported for $\mathcal{R}$1M distractor set. The datasets used for training are shown in brackets. ↑:upsampling; SP:spatial matching [24]. We denote ResNet101 and ResNet50 by R101 and R50. ¯R50 refers to a version of ResNet50 with the last block skipped.

features and the second layer restoring the original features. Mean-squared error loss that measures how well the autoencoder can reconstruct features is used to guide its learning. From Fig. 7, our method (CONV) achieves the best results compared to other methods for both datasets and different evaluation protocols.

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

mAP comparison. We conduct an extensive comparison of our method with state-of-the-art methods. All methods are tested on $\mathcal{R}$Oxf, $\mathcal{R}$Oxf+$\mathcal{R}$1M, $\mathcal{R}$Par and $\mathcal{R}$Par+$\mathcal{R}$1M. The results are shown in Tab. 4. Our MDA achieves the best mAP performance in most cases. We outperform previous state-of-the-art local aggregation methods in terms of mAP in the most challenging Hard protocol for $\mathcal{R}$Oxf and $\mathcal{R}$Par by significant 5.7% and 3.8% gains respectively. For $\mathcal{R}$1M, MDA also achieves the highest performance across local aggregation methods, outperforming in mAP the SOTA by 2.9% on $\mathcal{R}$Oxf-Medium, 6.4% on $\mathcal{R}$Oxf-Hard; and by 2.9% on $\mathcal{R}$Par-Medium, 5.2% on $\mathcal{R}$Par-Hard. The bottom fifth row of Tab. 4 shows that when we use 64-dimensional local features, we can still achieve comparable performance to the previous state-of-the-art method. The improvements are even higher when compared to global descriptors.

Speed and memory costs. In Tab. 4, we list the memory footprint required by different methods for $\mathcal{R}$1M. It should be noted that the memory requirement for local aggregation methods is higher than for global features *e.g.* 14.3GB as reported in HOW [37] *vs.* 7.6GB for GeM [27] descriptors in the $\mathcal{R}$1M-distractors set. Our method and HOW have almost the same memory footprint. For images with $n = 1000$ and $n = 2000$ local features, our method requires 7.6GB and 13.4GB of memory, respectively. We can further reduce the memory requirement by reducing the dimension of local features *e.g.* 7.4GB when $n = 2000$ and $C_T = 64$, which is less than GeM descriptors. It takes an average of 140ms for our method to extract 2000 local features from an image, which is a bit slower than HOW [37] (*e.g.* 127ms), on an RTX 2080Ti GPU. When 1000 local features are extracted from an image, it takes on average 0.71 seconds to search on $\mathcal{R}$Oxf+$\mathcal{R}$1M with the help of inverted index implemented by Python, which proves the potential of our method on real-time image retrieval.

5. Conclusions

In this paper, we propose a novel local feature learning framework for image retrieval. It contains a multi-dynamic attention module to simultaneously detect multiple discriminative local patterns in images, which allows for a more comprehensive capture of various semantic information of the image. The proposed network only needs image-level annotations and can be trained end-to-end. Extensive experiments demonstrate superior performance on image retrieval benchmark datasets. In the future, we will expand the research in two directions. First, how to integrate the large visual codebook and quantization process into a framework for end-to-end training. Second, how to introduce multi-scale structures into the network, without resizing images to multi-scales during testing.

Acknowledgements. This work was supported in part by the National Key R&D Program of China under contract 2018YFB1402605, in part by the National Natural Science Foundation of China under Contract 61822208 and 62021001, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1269–1277, 2015. 2

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 404–417, 2006. 1, 2

[3] Rudolf Beran et al. Minimum hellinger distance estimates for parametric models. *The annals of Statistics*, pages 445–463, 1977. 5

[4] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 726–743, 2020. 2, 3, 8

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pages 224–236, 2018. 1, 2

[7] Jiri Matas Dmytro Mishkin, Filip Radenovic. Repeatability is not enough: Learning discriminative affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1

[8] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8092–8101, 2019. 2

[9] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision (IJCV)*, pages 237–254, 2017. 1, 2, 3, 8

[10] Zhang Hanwang, Zha Zheng-Jun, Yan Shuicheng, Bian Jingwen, and Chua Tat-Seng. Attribute feedback. *Proceedings of the ACM international conference on Multimedia (MM)*, pages 79–88, 2012. 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[12] Hervé Jégou and Ondrej Chum. Negative evidences and cooccurrences in image retrieval: the benefit of pca and whitening. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 3

[13] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–317, 2008. 2

[14] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010. 2

[15] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, pages 1704–1716, 2011. 2

[16] Hervé Jégou and Andrew Zisserman. Triangulation embedding and democratic aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3310–3317, 2014. 2

[17] Cai Junjie, Zha Zheng-Jun, Wang Meng, Zhang Shiliang, and Tian Qi. An attribute-assisted reranking model for web image search. *IEEE transactions on image processing (TIP)*, pages 261–272, 2014. 1

[18] Cai Junjie, Zha Zheng-Jun, Zhou Wengang, and Tian Qi. Attribute-assisted reranking for web image retrieval. *Proceedings of the ACM international conference on Multimedia (MM)*, pages 873–876, 2012. 1

[19] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 685–701, 2016. 1, 2

[20] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, pages 91–110, 2004. 1, 2

[21] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Conference and Workshop on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2

[22] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. SOLAR: second-order loss and attention for image retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 253–270, 2020. 8

[23] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3456–3465, 2017. 1, 2, 6, 7

[24] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 5, 8

[25] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 5

[26] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5706–5715, 2018. 5, 6, 7, 8

[27] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, pages 1655–1668, 2018. 1, 2, 5, 8

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Conference and Workshop on Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 2

[29] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5107–5116, 2019. 8

[30] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-networks: Unsupervised learning to rank for interest point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[31] Johannes L. Schonberger, Filip Radenovic, Ondrej Chum, and Jan-Michael Frahm. From single image query to detailed 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[32] Oriane Simeoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 8

[33] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1470, 2003. 2

[34] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5109–5118, 2019. 2, 8

[35] Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, and Vassileios Balntas. SOSNet: Second order similarity regularization for local descriptor learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[36] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1401–1408, 2013. 2, 3, 4, 5, 6

[37] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 460–477, 2020. 1, 2, 3, 6, 7, 8

[38] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular Object Retrieval With Integral Max-Pooling of CNN Activations. In *International Conference on Learning Representations (ICLR)*, pages 1–12, 2016. 1, 2