Learning and aggregating deep local descriptors for instance-level recognition

Giorgos Tolias, Tomas Jenicek, and Ondřej Chum

Visual Recognition Group, Faculty of Electrical Engineering
Czech Technical University in Prague

Abstract. We propose an efficient method to learn deep local descriptors for instance-level recognition. The training only requires examples of positive and negative image pairs and is performed as metric learning of sum-pooled global image descriptors. At inference, the local descriptors are provided by the activations of internal components of the network. We demonstrate why such an approach learns local descriptors that work well for image similarity estimation with classical efficient match kernel methods. The experimental validation studies the trade-off between performance and memory requirements of the state-of-the-art image search approach based on match kernels. Compared to existing local descriptors, the proposed ones perform better in two instance-level recognition tasks and keep memory requirements lower. We experimentally show that global descriptors are not effective enough at large scale and that local descriptors are essential. We achieve state-of-the-art performance, in some cases even with a backbone network as small as ResNet18.

Keywords: deep local descriptors, deep local features, efficient match kernel, ASMK, image retrieval, instance-level recognition

1 Introduction

Instance-level recognition tasks are dealing with a very large number of classes and relatively small intra-class variability. Typically, even instance-level classification tasks are cast as instance-level search in combination with nearest neighbor classifiers. The first instance-level search approach to achieve good performance, *i.e.* Video Google [42], is based on local features and the Bag-of-Words (BoW) representation. Representing images as collections of vector-quantized descriptors of local features allows for efficient spatial verification [32], which turns out to be a key ingredient in search for small objects. Follow-up approaches improve the BoW paradigm either with finer quantization and better matching schemes [20,44,2] or with compact global descriptors generated through aggregation [22,1]. Good performance is achieved even without spatial verification.

The advent of deep networks made it easy to generate and train global image descriptors. A variety of approaches exist [14,35,3,28,50,16] that differ in the training data, in the loss function, or in the global pooling operation. However, the performance of global descriptors deteriorates for very large collections of

Fig. 1. Learned local features and descriptors matched with ASMK. Features assigned to the same visual word (65k words codebook) are shown in the same color; only top 20 common visual words (out of 94) are included. Accurate localization is not required since we do not use spatial verification to perform instance-level recognition.

images. Noh *et al.* [29] are the first to exploit the flexibility of global descriptor training in order to obtain local features and descriptors, called DELF, for the task of instance-level recognition. DELF descriptors are later shown [34] to achieve top performance when combined with the state-of-the-art image search approach, *i.e.* the Aggregated Selective Match Kernel (ASMK) [44]. Compared to compact global descriptors, this comes with higher memory requirements and search time cost. In contrast to other learned local feature detectors [13] that use keypoint-level supervision and non-maxima suppression, DELF features are not precisely localized and suffer from redundancy since deep network activations are typically spatially correlated.

In this work[1], we propose a local feature detector and descriptor based on a deep network. It is trained through metric learning of a global descriptor with image-level annotation. We design the architecture and the loss function so that the local features and their descriptors are suitable for matching with ASMK, see Figure 1. ASMK is known to deliver good performance even without spatial verification, *i.e.* precise feature localization is not crucial, and it deals well with repeated or bursty features. Therefore, the common drawbacks of existing deep local features for instance-level recognition are overcome. Unlike classical local features that attempt to offer precise localization to extract reliable descriptors, multiple nearby locations can give rise to a similar descriptor in our training; multiple similar responses are not suppressed, but averaged.

The main contribution of this work is the proposed combination of deep feature detector and descriptor with ASMK matching, which outperforms existing global and local descriptors on two instance-level recognition tasks, *i.e.* classification and search, in the domain of landmarks. Our ablation study shows that the proposed components reduce the memory required by ASMK. The learned local descriptors outperform by far deep global descriptors as well as other deep local descriptors combined with ASMK. Finally, we provide insight into why the image-level optimization is relevant for local-descriptors and ASMK matching.

[1] `https://github.com/gtolias/how`

2 Related work

We review the related work in learning global or local descriptors for instance-level matching task and local descriptors for registration tasks.

Global descriptors. A common approach to obtain global image descriptors with deep fully-convolutional neural networks is to perform global pooling on 3D feature maps. This approach is applied to activations generated by pre-trained networks [4,45,23] or end-to-end learned networks [35,14,15]. One of the first examples is SPoC descriptor by Babenko and Lempitsky [4] that is generated by simple global sum-pooling. Weighted sum-pooling is performed in CroW by Kalantidis *et al.* [23], where the weights are given by the magnitude of the activation vectors at each spatial location of the feature map. Such a 2D map of weights, seen as an attention map, is related to our approach as discussed in Section 4.2. Inspired by classical embeddings, Arandjelović *et al.* [3] extend the VLAD [22] descriptor to NetVLAD. Its contextually re-weighted counterpart, proposed by Kim and Frahm [24], introduces a learned attention map which is generated by a small network.

Local features and descriptors for instance-level recognition. Numerous classical approaches that are based on hand-crafted local features [27,25] and descriptors [26,8] exist in the literature of instance-level search [42,32,44,51,30]. Inspired by classical feature detection, Simeoni *et al.* [41] perform MSER [25] detection on activation maps. The features detected at one feature channel are used as tentative correspondences, hence no descriptors are required; the approach is applicable to any network and does not require learning. Learning of attentive deep local features (DELF) is introduced in the work of Noh *et al.* [29]. A global descriptor is derived from a network that learns to attend on feature map positions. The global descriptor is optimized with category-level labels and classification loss. At test time, locations with the strongest attention scores are selected while the descriptors are the activation vectors at the selected locations. This approach is highly relevant to ours. We therefore provide a number of different ablation experiments to reveal the key differences. A recent variant shows that it is possible to jointly learn DELF-like descriptors and global descriptors with a single model [11]. A similar achievement appears in the work of Yang *et al.* [49] with a scope that goes beyond instance-level recognition and covers image registration too.

Local features and descriptors for registration. A richer line of work exists in learning local feature detection and description for image registration where denser point correspondences are required. As in the previous tasks, some methods do not require any learning and are applicable on any pre-trained network. This is the case of the work Benbihi *et al.* [9] where activation magnitudes are back-propagated to the input image and local-maxima are detected. Learning is performed with or without labeling at the local level in a number of different approaches [38,13,12,7,10,47]. A large number of features is typically required for good performance. This line of research differentiates from our work; our focus is on large-scale instance-level recognition where memory requirements matter.

3 Background

In this section, the binarized versions of Selective Match Kernel (SMK) and its extension, the Aggregated Selective Match Kernel (ASMK) [44][2], are reviewed as the necessary background. This paper exploits the ASMK indexing and retrieval.

In SMK, an image is represented by a set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d\}$ of $n = |\mathcal{X}|$ d-dimensional local descriptors. The descriptors are quantized by k-means quantizer $q : \mathbb{R}^d \to \mathcal{C} \subset \mathbb{R}^d$, where $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$ is a codebook comprising $|\mathcal{C}|$ vectors (visual words). Descriptor $\mathbf{x}$ is assigned to its nearest visual word $q(\mathbf{x})$. We denote by $\mathcal{X}_c = \{x \in \mathcal{X} : q(x) = \mathbf{c}\}$ the subset of descriptors in $\mathcal{X}$ that are assigned to visual word $\mathbf{c}$, and by $\mathcal{C}_\mathcal{X}$ the set of all visual words that appear in $\mathcal{X}$. Descriptor $\mathbf{x}$ is mapped to a binary vector through function $b : \mathbb{R}^d \to \{-1, 1\}^d$ given by $b(\mathbf{x}) = \text{sign}(r(\mathbf{x}))$, where $r(\mathbf{x}) = \mathbf{x} - q(\mathbf{x})$ is the residual vector w.r.t. the nearest visual word and sign is the element-wise sign function.

The SMK similarity of two images, represented by $\mathcal{X}$ and $\mathcal{Y}$ respectively, is estimated by cross-matching all pairs of local descriptors with match kernel

$$S_{\text{SMK}}(\mathcal{X}, \mathcal{Y}) = \gamma(\mathcal{X})\,\gamma(\mathcal{Y}) \sum_{\mathbf{x}\in\mathcal{X}} \sum_{\mathbf{y}\in\mathcal{Y}} [q(\mathbf{x}) = q(\mathbf{y})] k(b(\mathbf{x}), b(\mathbf{y})), \tag{1}$$

where $[\cdot]$ is the Iverson bracket, $\gamma(\mathcal{X})$ is a scalar normalization that ensures unit self-similarity[3], *i.e.* $S_{\text{SMK}}(\mathcal{X}, \mathcal{X}) = 1$. Function $k : \{-1, 1\}^d \times \{-1, 1\}^d \to [0, 1]$ is given by

$$k(b(\mathbf{x}), b(\mathbf{y})) = \begin{cases} \left(\frac{b(\mathbf{x})^\top b(\mathbf{y})}{d}\right)^\alpha, & \frac{b(\mathbf{x})^\top b(\mathbf{y})}{d} \geq \tau \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\tau \in [0, 1]$ is a threshold parameter. Only descriptor pairs that are assigned to the same visual word contribute to the image similarity in (1). In practice, not all pairs need to be enumerated and image similarity is equivalently given by

$$S_{\text{SMK}}(\mathcal{X}, \mathcal{Y}) = \gamma(\mathcal{X})\,\gamma(\mathcal{Y}) \sum_{\mathbf{c}\in\mathcal{C}_\mathcal{X}\cap\mathcal{C}_\mathcal{Y}} \sum_{\mathbf{x}\in\mathcal{X}_\mathbf{c}} \sum_{\mathbf{y}\in\mathcal{Y}_\mathbf{c}} k(b(\mathbf{x}), b(\mathbf{y})), \tag{3}$$

where cross-matching is only performed within common visual words.

ASMK first *aggregates* the local descriptors assigned to the same visual word into a single binary vector. This is performed by $B(\mathcal{X}_c) = \text{sign}\left(\sum_{x\in\mathcal{X}_c} r(\mathbf{x})\right)$, with $B(\mathcal{X}_c) \in \{-1, 1\}^d$. Image similarity in ASMK is given by

$$S_{\text{ASMK}}(\mathcal{X}, \mathcal{Y}) = \gamma(\mathcal{X})\,\gamma(\mathcal{Y}) \sum_{\mathbf{c}\in\mathcal{C}_\mathcal{X}\cap\mathcal{C}_\mathcal{Y}} k(B(\mathcal{X}_c), B(\mathcal{Y}_c)). \tag{4}$$

[2] The binarized versions are originally [44] referred to as SMK* and ASMK*. Only binarized versions are considered in this work and the asterisk is omitted.

[3] To simplify, we use the same notation, *i.e.* $\gamma(\cdot)$, for the normalization of different similarity measures in the rest of the text. In each case, it ensures unit self-similarity of the corresponding similarity measure.

This is computationally and memory-wise more efficient than SMK. In practice, it is known to perform better due to handling the burstiness phenomenon [18]. Efficient search is performed by using an inverted-file indexing structure.

Simplifications. Compared to the original approach [44], we drop IDF weighting, pre-binarization random projections, and median-value thresholding, as these are found unnecessary.

4 Method

Learning local descriptors with ASMK in an end-to-end manner is challenging and impractical due to the use of large visual codebooks and the hard-assignment of descriptors to visual words. In this section, we first describe a simple framework to generate global descriptors that can be optimized with image-level labels and provide an insight into why this is relevant to optimizing the local representation too. Then, we extend the framework by additional components and discuss their relation to prior work.

4.1 Derivation of the architecture

In the following, let us assume a deep Fully Convolutional Network (FCN), denoted by function $f : \mathbb{R}^{w\times h\times 3} \to \mathbb{R}^{W\times H\times D}$, that maps an input image I to a 3D tensor of activations $f(I)$. The FCN is used as an extractor of dense deep local descriptors. The 3D activation tensor can be equivalently seen as a set $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^D\}$ of $W \times H$ D-dimensional local descriptors[4]. Each local descriptor is associated to a keypoint, also called local feature, that is equivalent to the receptive field, or a fraction of it, of the corresponding activations.

Let us consider global image descriptors constructed by global sum-pooling, known as SPoC [4]. Pairwise image similarity is estimated by the inner product of the corresponding ℓ_2-normalized SPoC descriptors. This can be equivalently seen as an efficient match kernel. Let $\mathcal{U} = f(I)$ and $\mathcal{V} = f(J)$ be sets of dense feature descriptors in images I and J, respectively. Image similarity is given by

$$S_{\text{SPoC}}(\mathcal{U}, \mathcal{V}) = \gamma(\mathcal{U})\,\gamma(\mathcal{V}) \sum_{\mathbf{u}\in\mathcal{U}} \sum_{\mathbf{v}\in\mathcal{V}} \mathbf{u}^\top \mathbf{v} \tag{5}$$

$$= \left(\gamma(\mathcal{U}) \sum_{\mathbf{u}\in\mathcal{U}} \mathbf{u}\right)^\top \left(\gamma(\mathcal{V}) \sum_{\mathbf{v}\in\mathcal{V}} \mathbf{v}\right) = Z_{\text{SPoC}}(\mathcal{U})^\top Z_{\text{SPoC}}(\mathcal{V}), \tag{6}$$

where $\gamma(\mathcal{U}) = 1/||\sum_{\mathbf{u}\in\mathcal{U}} \mathbf{u}||$. The optimization of the network parameters is cast as metric learning.

The interpretation of global descriptor matching in (6) as local descriptor cross-matching in (5) provides some useful insight. Local descriptor similarity

[4] Both $f(I)$ and $\mathcal{U}$ correspond to the same representation seen as a 3D tensor and a set of descriptors, respectively. We write $\mathcal{U} = f(I)$ implying the tensor is transformed into a set of vectors. $\mathcal{U}$ is, in fact, a multi-set, but it is referred to as set in the paper.

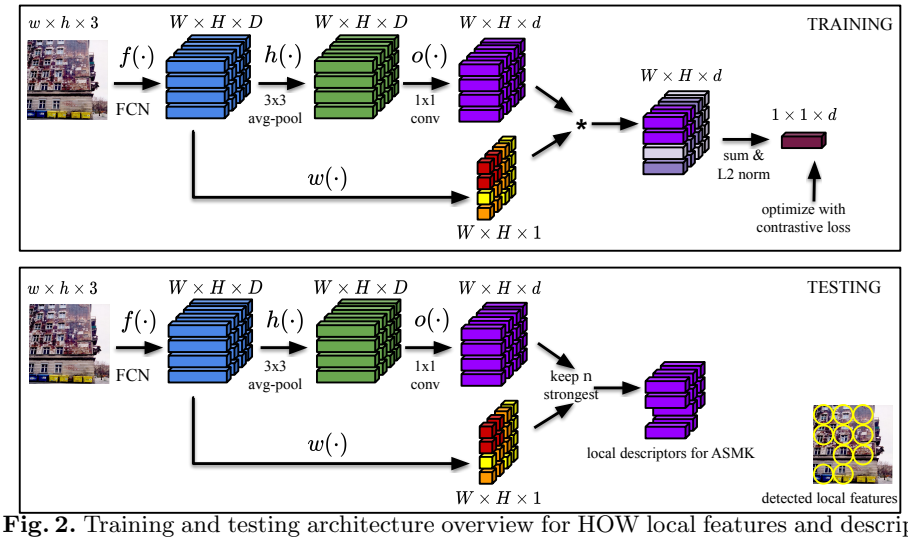

Fig. 2. Training and testing architecture overview for HOW local features and descriptors. A global descriptor is generated for each image during training and optimized with contrastive loss and image-level labels. During testing the strongest local descriptors (features), according to the attention map, are kept to represent the image. Then, these are used with ASMK for image search.

is estimated by $\mathbf{u}^\top \mathbf{v} = ||\mathbf{u}|| \cdot ||\mathbf{v}|| \cdot \cos(\mathbf{u}, \mathbf{v})$. Optimizing SPoC with image-level labels and contrastive loss implicitly optimizes the following four cases. First, local descriptors of background features, or image regions, are pushed to have small magnitude, *i.e.* small $||\mathbf{u}||$, so that their contribution in cross-matching is minimal. Similarly, local descriptors of foreground features are pushed to have large magnitude. Additionally, local descriptors of truly-corresponding features, *i.e.* image locations depicting the same object or object part, are pushed closer in the representation space, so that their inner product is maximized. Finally, local descriptors of non-corresponding features are pushed apart, so that their inner product is minimized. Therefore, optimizing global descriptors is a good surrogate to optimize local descriptors that are used to measure similarity via efficient match kernels, such as in ASMK. Importantly, this is possible with image-level labels and no local correspondences are required for training. In the following, we introduce additional components to the presented model, that are designed to amplify these properties.

Feature strength and attention. The feature strength, or importance, is estimated as the ℓ_2 norm of the feature descriptor $\mathbf{u}$ by the attention function $w(\mathbf{u}) = ||\mathbf{u}||$. During training, the feature strength is used to weigh the contribution of each feature descriptor in cross-matching. This way, the impact of the weak features is limited during the training, which is motivated by only the strongest features being selected in the test time. The attention function is fixed, without any parameters to be learned.

Local smoothing. Large activation values tend to appear in multiple channels of the activation tensor on foreground features [41]. Moreover, these activa-

tions tend not to be spatially well aligned. We propose to spatially smooth the activations by average pooling in an $M \times M$ neighborhood. The result is denoted by $\bar{\mathcal{U}} = h(f(I))$ or $\bar{\mathcal{U}} = h(\mathcal{U})$. Our experiments show that it is beneficial for the aggregation operation performed in ASMK. It is a fixed function, without any further parameters to be learned, and parameter M is a design choice.

Mean subtraction. Commonly used FCNs (all that are used in this work) generate non-negative activation tensors since a Rectified Linear Unit (ReLU) constitutes the last layer. Therefore, the inner product for all local descriptor pairs in cross-matching is non-negative and contributes to the image similarity. Mean descriptor subtraction is known to capture negative evidence [19] and allows to better disambiguate non-matching descriptors.

Descriptor whitening. Local descriptor dimensions are de-correlated by a linear whitening transformation; PCA-whitening improves the discriminability of both local [6] and global descriptors [36]. For efficiency, dimensionality reduction is performed jointly with the whitening. Formally, we group mean subtraction, whitening, and dimensionality reduction into function $o : \mathbb{R}^D \to \mathbb{R}^d$ given by $o(\mathbf{u}) = P(\mathbf{u} - \mathbf{m})$, where $P \in \mathbb{R}^{d \times D}$, $d \leq D$. Function $o(\cdot)$ is implemented by 1×1 convolution with bias. In practice, we initialize P and $\mathbf{m}$ according to the result of PCA whitening on a set of local descriptors from the training set and keep them fixed during the training.

Learning. Let $\bar{\mathcal{V}} = h(\mathcal{V})$ and $\bar{\mathbf{v}} \in \bar{\mathcal{V}}$ denote the activation vector after local average pooling $h(\cdot)$ at the same spatial location as $\mathbf{v} \in \mathcal{V}$. Similarly for $\bar{\mathcal{U}}$ and $\bar{\mathbf{u}}$. The image similarity that is being optimized during learning is expressed as

$$S_{how}(\mathcal{U}, \mathcal{V}) = \gamma(\bar{\mathcal{U}})\,\gamma(\bar{\mathcal{V}}) \sum_{\mathbf{u} \in \mathcal{U}} \sum_{\mathbf{v} \in \mathcal{V}} w(\mathbf{u}) \cdot w(\mathbf{v}) \cdot o(\bar{\mathbf{u}})^\top o(\bar{\mathbf{v}}) \tag{7}$$

$$= \left(\gamma(\bar{\mathcal{U}}) \sum_{\mathbf{u} \in \mathcal{U}} w(\mathbf{u}) o(\bar{\mathbf{u}}) \right)^\top \left(\gamma(\bar{\mathcal{V}}) \sum_{\mathbf{v} \in \mathcal{V}} w(\mathbf{v}) o(\bar{\mathbf{v}}) \right) = Z_{how}(\mathcal{U})^\top Z_{how}(\mathcal{V}). \tag{8}$$

A metric learning approach is used to train the network. In particular, contrastive loss is minimized: $||Z_{how}(\mathcal{U}) - Z_{how}(\mathcal{V}))||^2$ if $\mathcal{U}$ and $\mathcal{V}$ originate from matching (positive) image pairs, or $([\mu - ||Z_{how}(\mathcal{U}) - Z_{how}(\mathcal{V})||]_+)^2$ otherwise, where $[\cdot]_+$ denotes the positive part.

Test-time architecture. The architecture in test time is slightly modified; no global descriptor is generated. The n strongest descriptors $o(\bar{\mathbf{u}})$ for $\mathbf{u} \in \mathcal{U}$ are kept according to the importance given by $w(\mathbf{u})$. These are used as the local descriptor set $\mathcal{X}$ in ASMK. Multi-scale extraction is performed by resizing the input image at multiple resolutions. Local features from all resolutions are merged and ranked jointly according to strength. Selection of the strongest features is performed in the merged set. Multi-scale extraction is performed only during testing, and not during training. The training and testing architectures are summarized in Figure 2, while examples of detected features are shown in Figure 3.

Fig. 3. Example of multi-scale local feature detection. Left: Strongest 1,000 local features with color-coded strength; red is the strongest. Middle: Only the strongest feature per visual word is shown. Right: Attention maps for input images resized by scaling factors 0.25, 0.5, 1, and 2.

4.2 Relation to prior work

The proposed method has connections to different approaches in the literature which are discussed in this section. The work closest to ours, in terms of training of the local detector and descriptor, is DELF [29]. Following our notation, DELF generates local feature descriptors $Z_{\hat{h}\hat{o}\hat{w}}(\mathcal{U})$, where $\hat{h}(.)$ is identity, *i.e.* no local smoothing, $\hat{o}(.)$ is identity, *i.e.* no mean subtraction, whitening, and dimensionality reduction, and $\hat{w}(.)$ is a learned attention function. In particular, $\hat{w} : \mathbb{R}^D \to \mathbb{R}_+$ is a 2 layer convolutional network with 1×1 convolutions whose parameters are learned during the training. Dimensionality reduction of the descriptor space in DELF is performed as post-processing and is not a part of the optimization.[5] In terms of optimization, DELF performs the training in a classification manner with cross entropy loss. We show experimentally, that when combined with ASMK, the proposed descriptors are superior to DELF.

Fixed attention. Function $w(\cdot)$ is previously used to weigh activations and generate global descriptor, in particular by CroW [23]. It is also used in concurrent work for deep local features by Yang *et al.* [49]. The same attention function is used by Iscen *et al.* [17] for feature detection on top of dense SIFT descriptors without any learning. In our case, during learning, background or domain irrelevant descriptors are pushed to have low ℓ_2 norm in order to contribute less. The corresponding features are consequently not detected during test time.

Learned attention. Further example of learned attention, apart from DELF, is the contextual re-weighting that is performed to construct global descriptors in the work of Kim and Frahm [24]. The attention function is similar to DELF but with larger spatial kernel; a contextual neighborhood is used. A comparison between learned and fixed attention is included in the experimental ablations in Section 5.

Whitening. A common approach to whitening is to apply it as the last step in the pipeline, as post-processing. A similar approach to ours – applying the PCA whitening during training and learning it end-to-end – is followed by Gordo *et al.* [14] in the context of processing and aggregating regional descriptors to construct global descriptors. Comparison between "in-network" and post-processing whitening-reduction is included in Section 5.

[5] The main difference is that we do not follow the two stage training performed in the original work [29]; DELF is trained in a single stage for our ablations.

5 Experiments

We first review the datasets used for training, validation, and testing. Then, we discuss the implementation details for training and for testing with ASMK. Finally, we present our results on two instance-level tasks, namely recognition and search in the domain of landmarks and buildings.

5.1 Datasets

Training. The training dataset *SfM120k* [35] is used. It is the outcome of Stucture-from-Motion (SfM) [40] with 551 3D models for training. Matching pairs (anchor-positive) are formed by images with visual overlap (same 3D model). Non-matching pairs (anchor-negative) come from different 3D models.

Validation. We use the remaining 162 3D models of SfM120k to construct a challenging validation set reflecting the target task; this is different than the validation in [35]. We randomly choose 5 images per 3D model as queries. Then, for each query, images of the same 3D model with enough (more than 3), but not too many (at most 10), common 3D points with the query are marked as positive images to be retrieved. This avoids dominating the evaluation measure by a large number of easy examples. The remaining images of the same 3D model are excluded from evaluation for the specific query [32]. Skipping queries with an empty list of positive images results in 719 queries and 12,441 database images in total. Evaluation on the validation set is performed by instance-level search and performance is measured by mean Average Precision (mAP).

Evaluation on instance-level search. We use $\mathcal{R}$Oxford [32] and $\mathcal{R}$Paris [33] to evaluate search performance in the revisited benchmark [34]. They consist of 70 queries each, and 4993 and 6322 database images, respectively, and 1 million distractors called $\mathcal{R}$1M. We measure mAP on the Medium and Hard setups.

Evaluation on instance-level classification. We use instance-level classification as another task on which to evaluate the performance of the learned local descriptors. We perform search with ASMK and use k-nearest-neighbors classifiers for class predictions. The *Google Landmarks Dataset – version 2* (GLD_2) [48] is used. It consists of 4,132,914 train/database images with known class labels, and 117,577 test/query images which either correspond to the database landmarks, to other landmarks, or to non-landmark images. The query images are split into testing (private) and validation (public) sets with 76,627 and 40,950 images, respectively. Performance is measured by micro Average Precision (μAP) [31], also known as Global Average Precision (GAP), on the testing split. Note that we do not perform any learning on this dataset. We additionally create a mini version of GLD_2 to use for ablation experiments. It includes 1,000 query images, that are sampled from the testing split, and 10,0000 database images with labels, where the images come from 50 landmarks in total. We denote it by Tiny-GLD_2.

Classification is performed by accumulating the N top-ranked images per class. Prediction is given by the top-ranked class and the corresponding confidence is equal to the accumulated similarity. We use three variants, *i.e.* $N = 1$

(CLS1), $N = 10$ (CLS2), and $N = 10$ with accumulation of square-rooted similarity multiplied by a class weight (CLS3). The class weight is equal to the logarithm of the number of classes divided by the class frequency to down-weigh frequent classes.

5.2 Implementation details

Network architecture. We perform experiments with a backbone FCN ResNet18 and ResNet50, initialized by pre-training on ImageNet [39]. Descriptor dimensionality D is equal to 512 and 2048, respectively. We additionally experiment by removing the last block, *i.e.* "conv5_x", where D becomes 256 and 1024, respectively. We set $d = 128$ and $M = 3$ to perform 3×3 average pooling for local smoothing.

Training. We use a batch size of 5 tuples, where a tuple consists of 7 images – an anchor, a positive, and 5 negatives. Training images are restricted to a maximum resolution of 1,024 pixels. For each epoch, we randomly choose 2,000 anchor-positive pairs. The pool of candidate negatives contains 20,000 randomly chosen images, and hard-negative mining is performed before every epoch. We adopt the above choices from the work of Radenovic *et al.* [35], whose public implementation[6] we use to implement our method. To initialize P and $\mathbf{m}$, we use local descriptors from 5,000 training images extracted at a single scale. We use Adam optimizer with weight decay equal to 10^{-4}. The learning rate and margin μ are tuned, per variant, according to the performance on the validation set, by trying values $10^{-6}, 5 \cdot 10^{-6}, 10^{-5}, 5 \cdot 10^{-5}$ for learning rate and 0.5 to 0.9 with step 0.05 for margin μ. Margin μ is set equal to 0.8 for the proposed method, and learning rate equal to $5 \cdot 10^{-6}$ and 10^{-5} for ResNet18 and ResNet50 without the last block, respectively, according to the tuning process. Training is performed for 20 epochs and 1 epoch takes about 22 minutes for ResNet50 without the last block on a single NVIDIA Tesla V100 GPU with 32GB of DRAM. Training with cross entropy loss for ablations is performed with a batch size equal to 64 for 10 epochs. We repeat the training of each model and report mean and standard deviation over 5 runs. In large scale experiments, we evaluate a single model, the one with median performance on the validation set.

Validation. Validation performance is measured with ASMK-based search on the validation set. We measure validation performance every 5 epochs during training and the best performing model is kept.

Testing. We use ASMK to perform testing and to evaluate the performance of the learned local descriptors. The default ASMK configuation is as follows. We set threshold $\tau = 0$, $d = 128$, and use a codebook of $\kappa = 65536$ visual words, which is learned on local descriptors from 20,000 training images extracted at a single scale. Images are resized to have maximum resolution of 1024 pixels and multi-scale extraction is performed by re-scaling with factors 0.25, 0.353, 0.5, 0.707, 1.0, 1.414, 2.0. Assignment to multiple, in particular 5, visual words is performed for the query images. The strongest $n = 1000$ local descriptors are

[6] `https://github.com/filipradenovic/cnnimageretrieval-pytorch`

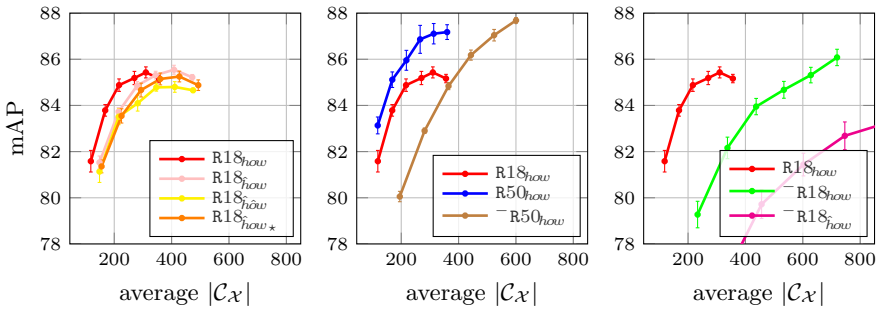

Fig. 4. Ablation study reporting performance versus average number of vectors per image in ASMK on the validation set for varying n (400,600,800,1000,1200,1400). Components used by DELF are denoted with $\hat{}$, while ours without. $\star$: random initialization of $\mathbf{m}$ and P (learned during training). Mean and standard deviation over 5 runs.

kept. The default configuration is used unless otherwise stated. The inverted file is compressed by delta coding.

5.3 Ablation experiments

We denote ResNet18 and ResNet50 by R18 and R50, and their versions with the last block skipped by $^{-}$R18 and $^{-}$R50, respectively. The local smoothing, whitening with reduction, and the fixed attention are denoted by subscripts h, o and w, respectively. The proposed method is denoted by $\mathtt{R18}_{how}$ when the backbone network is ResNet18. Following the same convention, the original DELF architecture is denoted by $^{-}\mathtt{R50}_{\hat{h}\hat{o}\hat{w}}$, where the dimensionality reduction is not part of the network but is performed by PCA whitening as post-processing.

Figure 4 shows the performance on the validation set versus the average number of binary vectors indexed by ASMK for the database images. We perform an ablation by excluding the proposed components and by using different backbone networks. Local smoothing results in larger amount of aggregation in ASMK (red vs pink) and reduces the memory requirements, which are linear in $|\mathcal{C}_\mathcal{X}|$. It additionally improves the performance when the last ResNet block is removed and feature maps have two times larger resolution (green vs magenta). Initializing and fixing $h(\cdot)$ with the result of PCA whitening is a beneficial choice too (orange vs pink). ResNet50 gives a good performance boost compared to ResNet18 (blue vs red), while removing the last block is able to reach higher performance at the cost of larger memory requirements (brown vs blue). More ablation experiments are shown in Table 1. Fixed attention is better than learned attention (3 vs 4). Metric learning with the contrastive loss delivers significantly better performance than cross entropy loss, in a classification manner, as done by Noh *et al.* [29] (4 vs 5). This is a confirmation of results that appear in the literature of instance-level recognition [14,46].

In Figure 5, we present the evolution of the model during training. We evaluate the performance of the optimized global descriptor for nearest neighbor

Method	Loss	Validation mAP	Validation $\lvert\mathcal{C}_\mathcal{X}\rvert$	$\mathcal{R}$Oxford mAP	$\mathcal{R}$Oxford $\lvert\mathcal{C}_\mathcal{X}\rvert$	Tiny-GLD$_2$ μAP	Tiny-GLD$_2$ $\lvert\mathcal{C}_\mathcal{X}\rvert$
1: R18$_{how}$	CO	85.2$\pm$0.3	263.8$\pm$ 0.1	74.8$\pm$0.2	283.6$\pm$ 0.2	81.3$\pm$1.0	252.2$\pm$ 0.5
2: R18$_{\hat{h}ow}$	CO	85.3$\pm$0.2	344.1$\pm$ 0.7	75.4$\pm$0.3	365.9$\pm$ 0.5	80.6$\pm$0.3	332.8$\pm$ 1.0
3: R18$_{\hat{h}\hat{o}w}$	CO	84.8$\pm$0.2	343.5$\pm$ 2.7	73.1$\pm$0.3	365.7$\pm$ 2.8	78.6$\pm$1.0	336.2$\pm$ 3.5
4: R18$_{\hat{h}\hat{o}\hat{w}}$	CO	83.7$\pm$0.9	354.4$\pm$ 2.0	70.0$\pm$1.7	380.7$\pm$ 2.9	74.2$\pm$3.6	358.6$\pm$ 5.5
5: R18$_{\hat{h}\hat{o}\hat{w}}$	CE	75.5$\pm$1.3	391.0$\pm$ 8.2	63.7$\pm$1.6	442.3$\pm$ 9.7	64.0$\pm$1.8	427.5$\pm$15.6
6: R18$_{\hat{h}\hat{o}w}$	CE	77.1$\pm$0.9	375.0$\pm$ 8.5	67.0$\pm$1.0	429.0$\pm$13.8	67.3$\pm$2.1	417.8$\pm$11.7
7: R18$_{\hat{h}o\hat{w}}$	CE	78.4$\pm$0.8	354.6$\pm$10.5	67.8$\pm$1.3	402.8$\pm$12.2	66.7$\pm$1.5	367.2$\pm$14.8
8: R18$_{\hat{h}\hat{o}\hat{w}}$	CE	77.0$\pm$0.9	279.6$\pm$ 5.6	65.4$\pm$0.5	320.6$\pm$ 6.8	68.6$\pm$1.8	300.9$\pm$11.4
9: R18$_{how}$	CE	80.4$\pm$0.5	308.3$\pm$ 4.0	69.9$\pm$1.4	345.4$\pm$ 5.2	71.1$\pm$1.8	308.4$\pm$ 4.1

Table 1. Ablation study for performance and average number of descriptors per image in ASMK (mean and standard deviation over 5 runs). 1: our method, 5: DELF variant. CO: contrastive loss, CE: cross entropy. On Tiny-GLD$_2$, classifier CLS3 is used.

search with multi-scale global descriptors, *i.e.* aggregation of global descriptors extracted at multiple image resolutions (same set of 7 resolutions as for the local descriptors). We additionally evaluate performance of the corresponding local descriptors with ASMK. The descriptor that is directly optimized in the loss performs worse than the internal local descriptors.

In the following we use two backbone networks – R18, which is fast with low memory footprint, and $^{-}$R50, which achieves better results at the cost of slower extraction and more memory.

5.4 Large-scale instance-level search

The performance comparison on $\mathcal{R}$Oxford and $\mathcal{R}$Paris is presented in Table 2. We do our best to evaluate the best available variants or models of the state-of-the-art approaches. The proposed local descriptors and DELF descriptors are evaluated with identical implementation and configuration of ASMK. The

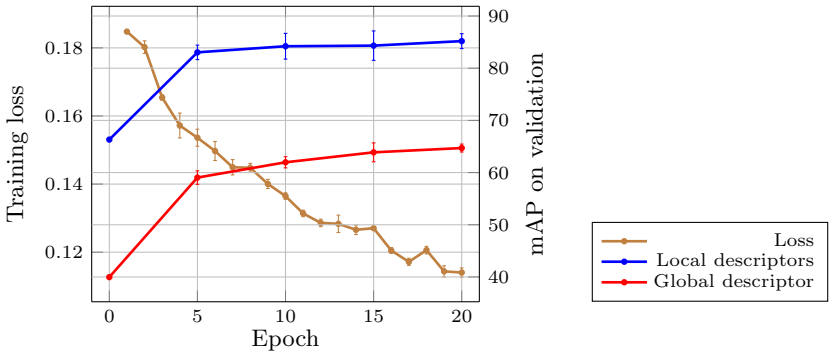

Fig. 5. Training loss and validation performance during the training. Mean and standard deviation ($\times$5 for better visualization) is reported over 5 runs. Both performance curves correspond to the same model, only its inference differs.

Method	FCN	Mem (GB)	Mean		$\mathcal{R}$O		$\mathcal{R}$O+$\mathcal{R}$1M		$\mathcal{R}$Par		$\mathcal{R}$P+$\mathcal{R}$1M	
			all	$\mathcal{R}$1M	med	hard	med	hard	med	hard	med	hard
Global descriptors & Euclidean distance search												
R-MAC [14]	R101	7.6	45.8	33.6	60.9	32.4	39.3	12.5	78.9	59.4	54.8	28.0
GeM [35]	R101	7.6	47.4	35.5	64.7	38.5	45.2	19.9	77.2	56.3	52.3	24.7
GeM [35]†	R101	7.6	47.3	35.2	65.4	40.1	45.1	22.7	76.7	55.2	50.8	22.4
GeM_w †	R101	7.6	49.0	37.2	67.8	41.7	47.7	23.3	77.6	56.3	52.9	25.0
GeM_{AP}[37]	R101	7.6	49.9	37.1	67.5	42.8	47.5	23.2	80.1	60.5	52.5	25.1
GeM_{AP}[37]†	R101	7.6	49.7	36.7	67.1	42.3	47.8	22.5	80.3	60.9	51.9	24.6
GeM_{AP}[37] PQ1	R101	1.9	49.6	36.7	67.1	42.2	47.7	22.5	80.3	60.8	51.9	24.6
GeM_{AP}[37] PQ8	R101	0.2	48.1	35.5	65.1	40.4	46.1	21.6	78.7	58.7	50.7	23.5
GeM_{AP}[37]† ▷1024	R101	3.8	49.0	35.8	66.6	41.6	46.7	21.7	80.0	60.3	51.0	23.8
GeM_{AP}[37]† ▷512	R101	1.9	47.5	33.7	65.9	40.5	43.9	19.7	79.5	59.4	48.9	22.3
GeM_{AP}+ GeM_w†	R101	15.3	53.4	41.4	70.5	45.7	52.6	27.1	**81.9**	**63.4**	57.0	29.1
Local descriptors & ASMK												
DELF [29]†	⁻R50	9.2	53.0	43.1	69.0	44.0	54.1	31.1	79.5	58.9	59.3	28.1
DELF [29] ‡	⁻R50	9.2	52.5	42.7	69.2	44.3	54.3	31.3	78.7	57.4	58.2	26.9
DELF [29][34]	⁻R50	9.2	51.5	42.2	67.8	43.1	53.8	31.2	76.9	55.4	57.3	26.4
R18_{how}, $n = 1000$	R18	4.6	54.8	43.5	75.1	51.7	55.7	32.0	79.4	58.3	57.4	28.9
$^{-}\text{R50}_{how}$, $n = 1000$	⁻R50	7.9	58.0	47.4	78.3	55.8	63.6	36.8	80.1	60.1	58.4	30.7
$^{-}\text{R50}_{how}$, $n = 1200$	⁻R50	9.2	58.8	48.4	78.8	56.7	64.5	37.7	80.6	61.0	59.6	31.7
$^{-}\text{R50}_{how}$, $n = 1400$	⁻R50	10.6	59.3	49.0	79.1	56.8	64.9	38.2	81.0	61.5	60.4	32.6
$^{-}\text{R50}_{how}$, $n = 2000$	⁻R50	14.3	**60.1**	**50.1**	**79.4**	**56.9**	**65.8**	**38.9**	81.6	62.4	**61.8**	**33.7**

Table 2. Performance comparison with global and local descriptors for instance-level search on $\mathcal{R}$Oxford ($\mathcal{R}$O) and $\mathcal{R}$Paris ($\mathcal{R}$P). Memory is reported for $\mathcal{R}$1M. Methods marked by † are evaluated by us using the public models for descriptor extraction. The method marked by ‡ is evaluated by us using the public descriptors [34]. GEM_w is a public model that includes a "whitening" (FC) layer. Dimensionality reduction is denoted by ▷ and descriptor concatenation by +. PQ8 and PQ1 denote PQ quantization using 8D and 1D sub-spaces, respectively.

proposed descriptors outperform all approaches by a large margin at large scale; global descriptors perform well enough at small scale, but at large scale, local representation is essential. Even with a backbone network as small as ResNet18, we achieve the second best performance (ranked after our method with ResNet50) on all cases of $\mathcal{R}$Oxford and at the hard setup of $\mathcal{R}$Paris +$\mathcal{R}$1M. Compared to DELF, our descriptors, named *HOW*, perform better for less memory.

Teichmann *et al.* [43] achieve average performance (mean all in Table 2) equal to 56.0 and 57.3 without and with spatial verification, respectively. They use additional supervision, *i.e.* manually created bounding boxes for 94,000 images, which we do not. The concurrent work of Cao *et al.* [11] achieves average performance equal to 58.3 but requires 485 GB of RAM, and slightly lower performance with 22.6 GB of RAM for binarized descriptors.

In an effort to further compress the memory requirements of competing global descriptors, we evaluate the best variant with dimensionality reduction and with Product Quantization (PQ) [21]. We further improve their performance by concatenating the two best performing ones. Among all these variants, the proposed approach appears to be a good solution in the performance-memory trade-off.

Storing less vectors in ASMK affects memory but also speed. A query of average statistics computes the hamming distance for about $1.2 \cdot 10^6$ (average

Method	FCN	Training set	Memory (GB)	CLS1	CLS2	CLS3
GEM [35]	R101	SfM-120k	31.5	1.9	18.0	24.1
GEM_w	R101	SfM-120k	31.5	3.7	23.4	28.7
GEM-AP [37]	R101	SfM-120k	31.5	2.8	14.8	20.7
DELF [29]	$^-$R50	Landmarks [5]	24.1	2.1	11.9	21.9
R18_{how}	R18	SfM-120k	17.5	8.5	20.0	27.0
$^-\text{R50}_{how}$	$^-$R50	SfM-120k	29.3	18.5	33.1	36.5

Table 3. Performance comparison on instance-level recognition (GLD_2). μAP is reported for classification with 3 different k-nn classifiers. Existing methods are evaluated by us using the public models. Global descriptors are combined with simple nearest neighbor search, and local descriptors are combined with ASMK-based retrieval.

number of vectors stored per inverted list multiplied by average $|\mathcal{C}_\mathcal{X}|$) 128D binary vector pairs in the case of $^-$R18 with our method at large scale. The same number for $^-$R50 is $3.2 \cdot 10^6$. Search on $\mathcal{R}$Oxford +$\mathcal{R}$1M takes on average 0.75 seconds on a single threaded Python non-optimized CPU implementation.

5.5 Large-scale instance-level classification

Performance comparison on GLD_2 is presented in Table 3. We extract DELF keeping at most top 1,000 local descriptors and reduce dimensionality to 128. The proposed local descriptors and DELF descriptors are evaluated with identical configuration of ASMK. Our method outperforms all other methods with memory requirements that are even lower than raw 2048D global descriptors. DELF, R18_{how}, and $^-\text{R50}_{how}$, end up with 347, 252, and 423 vectors to store per image on average, respectively. Due to the strength threshold, DELF extracted 504 local descriptors per image on average which is significantly less than for images of $\mathcal{R}$1M. Multiple visual word assignment is not performed in this experiment, for any of the methods, to reduce the computational cost of search.

6 Conclusions

An architecture for extracting deep local features is designed to be combined with ASMK matching. The proposed method consistently outperforms other methods on a number of standard benchmarks, even if a less powerful backbone network is used. Through an extensive ablation study, we show that the SoA performance is achieved by the synergy of the proposed local feature detector with ASMK. We show that methods based on local features outperform global descriptors in large scale problems, and also that the proposed method outperforms other local feature detectors combined with ASMK. We demonstrate why the proposed architecture, despite being trained with image-level supervision only, is effective in learning image similarity based on local features.

Acknowledgements. The authors would like to thank Yannis Kalantidis for valuable discussions. This work was supported by MSMT LL1901 ERC-CZ grant. Tomas Jenicek was supported by CTU student grant SGS20/171/OHK3/3T/13.

References

1. Arandjelović, R., Zisserman, A.: All about VLAD. In: CVPR (2013)
2. Arandjelović, R., Zisserman, A.: DisLocation: Scalable descriptor distinctiveness for location recognition. In: ACCV (2014)
3. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
4. Babenko, A., Lempitsky, V.: Aggregating deep convolutional features for image retrieval. In: ICCV (2015)
5. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: ECCV (2014)
6. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: CVPR (2017)
7. Barroso Laguna, A., Riba, E., Ponsa, D., Mikolajczyk, K.: Key. net: Keypoint detection by handcrafted and learned cnn filters. In: ICCV (2019)
8. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: SURF: Speeded up robust features. Computer Vision and Image Understanding **110**(3), 346–359 (May 2008)
9. Benbihi, A., Geist, M., Pradalier, C.: Elf: Embedded localisation of features in pre-trained cnn. In: CVPR (2019)
10. Bhowmik, A., Gumhold, S., Rother, C., Brachmann, E.: Reinforced feature points: Optimizing feature detection and description for a high-level task. In: CVPR (2020)
11. Cao, B., Araujo, A., Sim, J.: Unifying deep local and global features for efficient image search. In: arxiv (2020)
12. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: CVPRW (2018)
13. Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., Sattler, T.: D2-net: A trainable cnn for joint detection and description of local features. In: CVPR (2019)
14. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. IJCV (2017)
15. Gu, Y., Li, C., Jiang, Y.G.: Towards optimal cnn descriptors for large-scale image retrieval. In: ACM Multimedia (2019)
16. Husain, S., Bober, M.: Improving large-scale image retrieval through robust aggregation of local descriptors. PAMI **39**(9), 1783–1796 (Jan 2016)
17. Iscen, A., Tolias, G., Gosselin, P.H., Jégou, H.: A comparison of dense region detectors for image search and fine-grained classification. IEEE Transactions on Image Processing **24**(8), 2369–2381 (2015)
18. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR (Jun 2009)
19. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In: ECCV (Oct 2012)
20. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV **87**(3), 316–336 (Feb 2010)
21. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. PAMI **33**(1), 117–128 (Jan 2011)
22. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local descriptors into compact codes. In: PAMI (Sep 2012)
23. Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. In: ECCVW (2016)

24. Kim, H.J., Dunn, E., Frahm, J.M.: Learned contextual feature reweighting for image geo-localization. In: CVPR (2017)
25. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22**(10), 761–767 (2004)
26. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27**(10), 1615–1630 (2005)
27. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. IJCV **65**(1/2), 43–72 (2005)
28. Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marques, F., Giro-i Nieto, X.: Bags of local convolutional features for scalable instance search. In: ICMR (2016)
29. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: ICCV (2017)
30. Perronnin, F., Liu, Y., Sanchez, J., Poirier, H.: Large-scale image retrieval with compressed Fisher vectors. In: CVPR (2010)
31. Perronnin, F., Liu, Y., Renders, J.M.: A family of contextual measures of similarity between distributions with application to image retrieval. In: CVPR. pp. 2358–2365 (2009)
32. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
33. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (Jun 2008)
34. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: CVPR (2018)
35. Radenović, F., Tolias, G., Chum, O.: Fine-tuning CNN image retrieval with no human annotation. PAMI **41** (Jul 2019)
36. Razavian, A.S., Sullivan, J., Carlsson, S., Maki, A.: Visual instance retrieval with deep convolutional networks. ITE Trans. on Media Technology and Applications (2016)
37. Revaud, J., Almazán, J., de Rezende, R.S., de Souza, C.R.: Learning with average precision: Training image retrieval with a listwise loss. In: ICCV (2019)
38. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: Repeatable and reliable detector and descriptor. In: NeurIPS (2019)
39. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
40. Schönberger, J.L., Radenović, F., Chum, O., Frahm, J.M.: From single image query to detailed 3D reconstruction. In: CVPR (2015)
41. Siméoni, O., Avrithis, Y., Chum, O.: Local features and visual words emerge in activations. In: CVPR (2019)
42. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
43. Teichmann, M., Araujo, A., Zhu, M., Sim, J.: Detect-to-retrieve: Efficient regional aggregation for image search. In: CVPR (2019)
44. Tolias, G., Avrithis, Y., Jégou, H.: Image search with selective match kernels: aggregation across single and multiple images. IJCV (2015)

45. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: ICLR (2016)
46. Vo, N., Jacobs, N., Hays, J.: Revisiting im2gps in the deep learning era. In: CVPR (2017)
47. Wang, Q., Zhou, X., Hariharan, B., Snavely, N.: Learning feature descriptors using camera pose supervision. In: arXiv (2020)
48. Weyand, T., Araujo, A., Cao, B., Sim, J.: Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In: CVPR (2020)
49. Yang, T., Nguyen, D., Heijnen, H., Balntas, V.: Ur2kid: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision. In: arxiv (2020)
50. Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: CVPR (2015)
51. Zhu, C.Z., Jégou, H., ichi Satoh, S.: Query-adaptive asymmetrical dissimilarities for visual object retrieval. In: ICCV (2013)