Local Features and Visual Words Emerge in Activations

Oriane Siméoni[1] Yannis Avrithis[1] Ondřej Chum[2]
[1]Univ Rennes, Inria, CNRS, IRISA [2]VRG, FEE, CTU in Prague

Abstract

We propose a novel method of deep spatial matching *(DSM) for image retrieval. Initial ranking is based on image descriptors extracted from convolutional neural network activations by global pooling, as in recent state-of-the-art work. However, the same sparse 3D activation tensor is also approximated by a collection of local features. These local features are then robustly matched to approximate the optimal alignment of the tensors. This happens without any network modification, additional layers or training. No local feature detection happens on the original image. No local feature descriptors and no visual vocabulary are needed throughout the whole process.*

We experimentally show that the proposed method achieves the state-of-the-art performance on standard benchmarks across different network architectures and different global pooling methods. The highest gain in performance is achieved when diffusion on the nearest-neighbor graph of global descriptors is initiated from spatially verified images.

1. Introduction

Image and specific object retrieval is commonly addressed as large scale image matching: a query is matched against the database images and the final ranking is given by the matching score. In the early retrieval days, methods based on local features were dominating [35, 23]. The matching score was first approximated by a similarity of bag of words [35] or aggregated descriptors [14], and then re-ranked by efficient spatial verification [26, 25].

Recently, image retrieval is dominated by convolutional neural networks (CNNs) [10, 29]. Image representation is derived from the output of the CNN, which can be interpreted as a collection of 2D response maps of pattern detectors. The position of the response indicates the location of the pattern in the image, the size of the pattern is limited by the receptive field, and the value of the response indicates the confidence in the presence of the pattern.

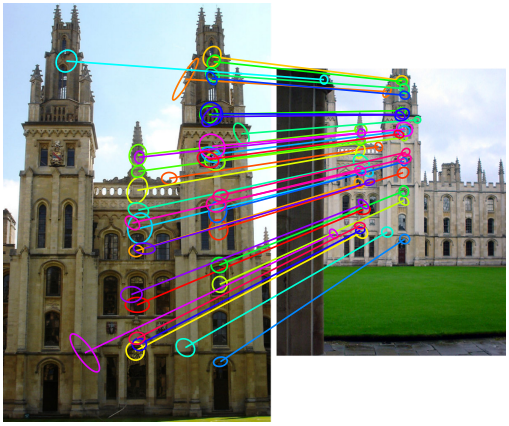

Figure 1. Fast spatial matching [26] finds a linear geometric transformation between two views of an object based on a local feature representation. This is used for spatial verification in large-scale image retrieval. Inlier correspondences shown, colored by visual word. What is the underlying representation?[1]

(a) *SIFT [19] descriptors on Hessian-affine [22] local features.*
(b) *Descriptors on detected patches by an end-to-end differentiable pipeline using patch pair labels [42].*
(c) *A subset of convolutional features at locations selected by an attention mechanism learned on image-level labels [24].*
(d) *Local maxima on each channel of a vanilla feature map. No vocabulary needed.*

Images of corresponding objects or object parts have similar response in all channels. It is known that the image-to-image mapping can be recovered by correlating the response tensors of the two images [18, 4, 31].

In general, the CNN activation tensor size depends on the number of channels and the image size. It is is too large to be stored, especially for large-scale applications. To construct a descriptor of a fixed and reasonable size, vectors obtained by global pooling are extracted instead, for instance mean pooling [2], max pooling [40], generalized-mean pooling [29], and others [15, 40]. If the CNN-response tensors are matching, the statistics obtained after the global pooling should be matching too.

Global pooling not only reduces the size of the descriptor, but also injects view-point invariance. In fact, the global pooling is, similarly as bag of features, invariant to a very

[1] Answer: (b) [this work].

broad class of transformations. Thus, some information, namely geometric consistency, is lost.

In this work we introduce a very simple way of extracting from the CNN activations a representation that is suitable for geometric verification, which we apply to re-ranking. Ideally, one would estimate the geometric transformation to align the activation tensors and compare. Nevertheless, as stated previously, this would be impractical. We propose to approximate this process, exploiting two properties of the activations: high values are more important and the activations are sparse. Therefore each channel can be well approximated by a small number of extremal regions.

After discussing related work in section 2, we develop our method, called *deep spatial matching* (DSM), in section 3. Experimental results are reported in section 4 and conclusions are drawn in section 5.

2. Related work

Shortly after the popularization of AlexNet and the illustration of image retrieval using the output vector of its last *fully connected* layer [17], it was found that convolutional layers possessed much more discriminative power and were better adapted to new domains [3]. However, just flattening the 3D *convolutional activation* tensor into a vector yields a non-invariant representation. The next obvious attempt was to split images into patches, apply *spatial max-pooling* and match them exhaustively pairwise, which could beat conventional pipelines [37] for the first time, but is expensive [30]. Is was then shown more efficient to apply regional max-pooling on a single convolutional activation of the entire image [40]. Combined with integral image computation, [40] also allowed fast sliding window-style spatial matching, still requiring to store a tensor of the same size as the entire convolutional activation tensor.

Network *fine-tuning* of globally pooled representations like MAC and R-MAC [40] using metric learning loss functions for the retrieval task followed, giving state of the art performance [10, 29]. The power of CNN representations of one or very few regional descriptors per image allowed reducing image retrieval to nearest neighbor search and extending previous query expansion [6, 39] into efficient online exploration of the entire nearest neighbor graph of the dataset by *diffusion* [12]. This progress nearly solved previous benchmarks and necessitated revisiting them [27]. The main drawback of these compact representations is that they are not compatible with spatial verification, which would ensure accuracy of the top ranking results as was the case with conventional representations [26, 36]. In fact, given enough memory, such representations are still the state of the art [27].

Most notable in the latter benchmark was the performance of *deep local features* (DELF) [24], which combined the power of CNN features with the conventional pipeline of hundreds of local descriptors per image, followed by encoding against a vocabulary and search by inverted files. The DELF apprach does allow spatial verification at the cost of more memory and incompatibility with global representations, which on the other hand, allow nearest neighbor search. In this work, we attempt to reduce this gap by introducing a new representation that encodes geometric information allowing spatial verification, yet it has a trivial relation to the global representation used for nearest neighbor search.

At this point, it is worth looking at the geometric alignment of two views shown in Figure 1 and reflecting on what could be the underlying representation and what would be the advantages of each choice. In terms of geometric correspondence, most recent efforts have focused on either dense registration [18, 4, 31, 32], which would not apply to retrieval due to the storage limitation, or imitating conventional pipelines [19, 22]. In the latter case, two dominating paradigms are *detect-then-describe* [42] and *describe-then-detect* [24], both of which result in a large set of visual descriptors. We break this dilemma by "reading off" information directly from feature maps.

3. Deep spatial matching

We begin by motivating our approach, and then present the proposed architecture, followed by the main ideas, including feature detection and representation from CNN activations, spatial matching and re-ranking.

3.1. Motivation

Given a convolutional neural network ending in global average pooling, objects of a given class can be localized by *class activation maps* (CAM) [43], even if the network has only been trained for classification on image-level labels. These maps are linear combinations of individual feature maps (channels) of the last convolutional layer. Grad-CAM [33] generalizes this idea to any network architecture and allows visualization at any layer by a similar linear combination on the gradient signal instead. Without any class semantics, another linear combination produces a saliency map used for spatial pooling in *cross-dimensional weighting* (CroW) [15]. The latter weighs channels according to *sparsity*, but in all cases the linear combinations only provide coarse localization of objects of a given class or class-agnostic salient regions.

Experiments in [40] have shown *max-pooling* of convolutional activations (MAC) to be superior to other spatial pooling schemes, at least for image retrieval. This can be connected to the sparsity of the activations. More interestingly, looking at the positions of the maxima in channels contributing most to image similarities, one can readily identify correspondences between two images [40]. The same has been observed in person re-identification [1].

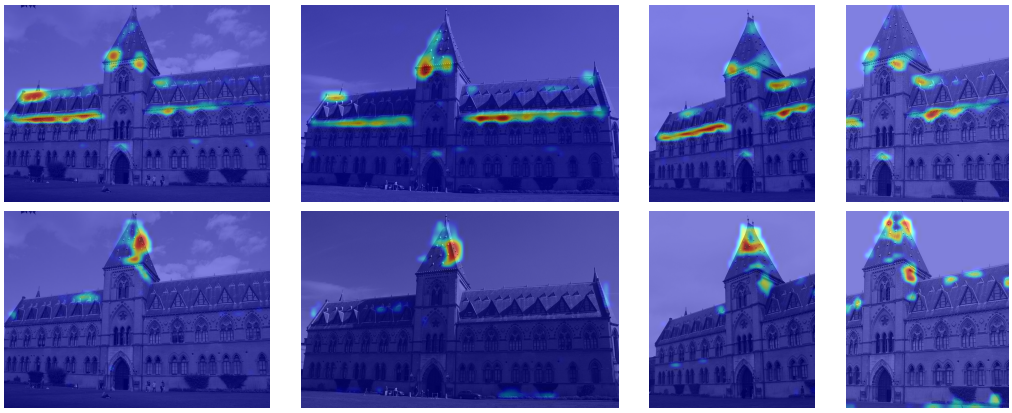

Figure 2. Four views (columns) of the Museum of natural history in the $\mathcal{R}$Oxf dataset, overlaid with two different feature maps (rows) of the last convolutional layer of the VGG16 [34] network. The filter kernel in each channel is responding to similar image structures in all images. All activations are naturally sparse and nonzero responses agree in both location and local shape between all images.

Later, *generalized mean pooling* (GeM) [29] was shown to outperform max-pooling. This can be attributed to the fact that it allows for more than one locations contributing to the representation, while still being more selective than average pooling.

Following the above observations, we investigate the responses of the last convolutional layer of VGG on several matching images of the $\mathcal{R}$Oxf dataset. This time we do not limit ourselves to the channels that are contributing most to image similarity (assuming *e.g.* global max-pooling and cosine similarity), but we rather observe all channels. We find out that, as illustrated in Figure 2, for two example channels, in most cases the responses to all images are not just sparse but consistent too: the filters respond to the same structures in the images, and there are responses at consistent locations with consistent local shape. The responses exhibit translation and scale covariance to some extent. The deep spatial matching proposed in this work is motivated by the following ideas.

Instead of just reducing each channel to a single scalar, why not keep all the peaks of the responses in each channel along with geometric information (coordinates and local shape)? Instead of attaching an entire descriptor to each such geometric entity, why not just attach the channel it was extracted from, as if it was a visual word?

We propose a method in-between two commonly used approaches, taking the best of the two worlds. One is conventional representations of thousands of local features per image, each with its own descriptor, suitable for inverted files and spatial verification. The other relies on a single global or few regional descriptors per image, leading to compact storage, efficient nearest neighbor search, and graph-based re-ranking. The proposed approach is applicable to any network fine-tuned for retrieval, without requiring any network adaptation, even without any training. It needs no vocabulary and it is trivially related to the global descriptors that dominate the state of the art.

3.2. Method overview

The preceding ideas give rise to the *deep spatial matching* (DSM) network architecture that we introduce in this work, illustrated in Figure 3. We consider a fully convolutional backbone network architecture that maintains as much as possible spatial resolution. We denote by f the *network function* that maps an input image to the feature tensor of the last convolutional layer. We assume that the backbone network f, when followed by a pooling mechanism *e.g.* MAC [40] or GeM [29], extracts a global descriptor that is used *e.g.* for retrieval [10, 29].

As shown in Figure 3, two input images x_1, x_2 are processed by a network into 3-dimensional *feature tensors* $A_1 := f(x_1), A_2 := f(x_2)$ where $A_i \in \mathbb{R}^{w_i \times h_i \times k}$, $w_i \times h_i$ is the spatial resolution of A_i for $i = 1, 2$ and k is the number of channels (features). Using the two feature tensors is standard practice in image registration [18, 4], optical flow [8] or semantic alignment [16, 31], but here we use an entirely different way of working with the tensors.

In particular, similarly to local feature detection from a single feature tensor [24], most registration/flow/alignment methods see a feature tensor $A \in \mathbb{R}^{w \times h \times k}$ as a $w \times h$ array of k-dimensional vector descriptors. Then, given two feature tensors, most consider the correlation of the two 2-dimensional arrays, seeking dense correspondences. By contrast, from each feature tensor A_1, A_2 we extract a sparse collection of *local features* $\mathcal{P}_1 := d(A_1), \mathcal{P}_2 := d(A_2)$ respectively. The feature detector d, discussed in section 3.3, operates independently per channel and each local feature collection $\mathcal{P}$ is a list of sets, one per channel. Local features are represented as discussed in section 3.4.

Then, the two local feature collections $\mathcal{P}_1, \mathcal{P}_2$ undergo *spatial matching*, denoted as g and discussed in section 3.5, returning a collection of inliers $\mathcal{M}$ and a geometric trans-

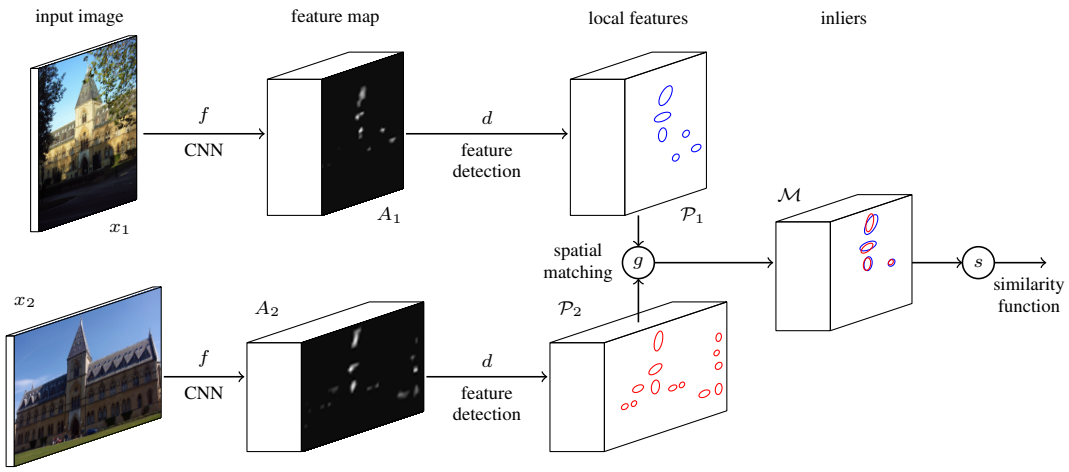

Figure 3. *Deep spatial matching* (DSM) network architecture. Two input images x_1, x_2 are mapped by network f to feature tensors A_1, A_2 respectively. Sparse *local features* $\mathcal{P}_1, \mathcal{P}_2$ extracted by *detector* d undergo *spatial matching* g, resulting in a collection of inliers $\mathcal{M}$. Similarity function s applies to this collection. Local features are detected and matched independently per channel, with channels playing the role of *visual words*. This takes place without any additional learning and without adapting the backbone network. In retrieval, only local features $\mathcal{P}_1, \mathcal{P}_2$ are stored and g applies directly at re-ranking.

formation T. We fit a linear motion model to a collection of tentative *correspondences*, *i.e.*, pairs of local features from the two images, which are formed again independently per channel. This implicitly assumes that the "appearance" of each local feature is *quantized* according to channel where it was detected, hence channels play the role of *visual words*, without any descriptor vectors ever being computed. The output collection of *inlier* correspondences $\mathcal{M}$ is again given as a list of sets, one per channel. Finally, *similarity function* s applies to $\mathcal{M}$.

The entire feature detection and matching mechanism takes place without adapting the backbone network in any way and without any additional learning. When applied to image retrieval, this architecture assumes that local features have been precomputed and are the representation of the database images, that is, feature tensors are discarded. Based on this representation, spatial matching g applies directly for geometric verification and *re-ranking*.

3.3. Local feature detection

To detect local features in each feature channel we use *maximally stable extremal regions* (MSER) by Matas *et al*. [20]. MSERs are defined over a 2-dimensional input, in our case over feature map $A^{(j)}$ of feature tensor A independently for each channel $j = 1, \ldots, k$. The extractor finds continuous regions R with all interior points having strictly higher response value than neighboring outer points. Regions satisfying a stability criterion [20] and passing location non-maxima suppression are selected. These features are appropriate for regions of arbitrary shape, including localized peaks, blobs, elongated or even nested regions.

When MSERs are used as image features, the response value is either the image intesity (MSER$^+$) or the reverse intensity (MSER$^-$). In our case, only regions of high CNN activations in sparse feature maps are of interest, and hence only one type of MSERs are extracted directly over the feature map responses.

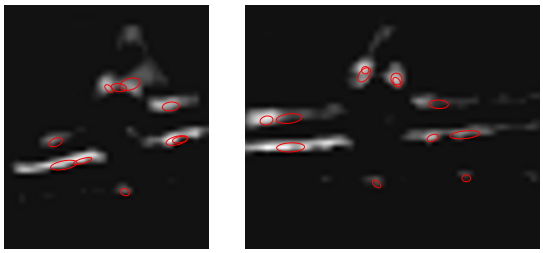

Figure 4. Feature maps from one channel for two different views of a building in the $\mathcal{R}$Oxf dataset. Ellipses are fitted to the local features detected by MSER.

3.4. Local feature representation

For each MSER R detected in channel j we compute a scalar value v representing strength. It is pooled over the spatial support of R in feature map $A^{(j)}$ as $v := \mathrm{pool}_{r \in R} A^{(j)}(r)$. Here pool can be any pooling operation like max, mean, or generalized mean. We also fit an ellipse by matching its first and second moments, *i.e.* its 2×1 mean (position) vector μ and 2×2 covariance matrix (local shape) Σ. For instance, Figure 4 shows an example of ellipses fitted to the MSER detected on feature maps of one channel for two views of the Oxford Museum of Natural History. Ellipses are well aligned in the two views. The local feature corresponding to R is then represented

Figure 5. Examples of our *deep spatial matching* (DSM) between images from $\mathcal{R}$Oxf and $\mathcal{R}$Par benchmarks. Inlier features (ellipses) and correspondences (lines) shown in different colors.

by tuple $p := (\mu, \Sigma, v)$. Finally, we collect local features $\mathcal{P} = (P^{(1)}, \dots, P^{(k)})$ where $P^{(j)}$ contains the local features p found in channel j. The entire operation is denoted by $\mathcal{P} := d(A)$.

To treat feature channels as visual words, we assume that features are uncorrelated, which does not hold in practice as indicated by the fact that whitening boosts performance. The same filter may respond to a variety of input patterns and worse, several filters may respond to the same pattern. This can increase the level of interference in negative image pairs. For this reason we apply *non-maximum suppression* (NMS) over all channels on the detected regions of each database image. Because local features are often small, we set a low IoU threshold. We do not apply NMS to the query image in order to allow matches from any channel.

3.5. Spatial matching

Given the local features $\mathcal{P}_1, \mathcal{P}_2$ of two images x_1, x_2, we use *fast spatial matching* (FSM) [26] to find the geometric transformation T between the two images and the subsets of $\mathcal{P}_1, \mathcal{P}_2$ that are consistent with this transformation. Matching is based on *correspondences*, *i.e.* pairs of local features $c = (p_1, p_2)$ from the two images. We allow pairs only between local features of the same channel, that is, p_1, p_2 are in $\mathcal{P}_1^{(j)}, \mathcal{P}_2^{(j)}$ respectively for some channel j. We thus treat channels as *visual words*, as if local features were assigned descriptors that were vector-quantized against a vocabulary and matched with the discrete metric. We begin with the *tentative correspondences* that is the set of all such pairs, $\mathcal{C} := (\mathcal{P}_1^{(1)} \times \mathcal{P}_2^{(1)}, \dots, \mathcal{P}_1^{(k)} \times \mathcal{P}_2^{(k)})$.

FSM is a variant of RANSAC [9] that generates a transformation hypothesis from a single correspondence. We adopt the linear 5-dof transformation which allows for translation, anisotropic scale and vertical shear but no rotation, assuming images are in "upright" orientation. Given a correspondence of two features $p_1 = (\mu_1, \Sigma_1, v_1)$ and $p_2 = (\mu_2, \Sigma_2, v_2)$, one finds from the two ellipses $(\mu_1, \Sigma_1), (\mu_2, \Sigma_2)$ the transformations T_1, T_2 that map them to the unit circle while maintaining the y-direction, and defines the transformation hypothesis $T = T_2^{-1} T_1$.

A hypothesis is evaluated based on the number of *inliers*, that is, correspondences that are consistent with it. Because tentative correspondences are not too many, all possible hypotheses are enumerated. Following [26], we are using LO-RANSAC [5], which iteratively evaluates promising hypotheses by fitting a full transformation to inliers by least squares. The transformation T with the most inliers $\mathcal{M}$ is returned. The operation is denoted by $(\mathcal{M}, T) := g(\mathcal{P}_1, \mathcal{P}_2)$ and $\mathcal{M} = (\mathcal{M}^{(1)}, \dots, \mathcal{M}^{(k)})$ is a list of sets of inliers, one per channel.

3.6. Retrieval and re-ranking

In an image retrieval scenario, n database images $X = \{x_1, \dots, x_n\}$ are given in advance. For each image x_i with feature tensor A_i, its local features $\mathcal{P}_i := d(A_i)$ are computed along with a global descriptor z_i spatially pooled directly from A_i again *e.g.* by max or GeM pooling; A_i is then discarded. At query time, given query image x with feature tensor A, local features $\mathcal{P} := d(A)$ and global descriptor z, we first rank $\{z_1, \dots, z_n\}$ by cosine similarity to z, and then the top-ranking images undergo spatial matching against $\mathcal{P}$ according to $(\mathcal{M}_i, T_i) := g(\mathcal{P}, \mathcal{P}_i)$ and are re-ranked according to *similarity function* $s(\mathcal{M}_i)$. The most common choice, which we also follow in this work, is the number of inliers found, $s(\mathcal{M}_i) := \sum_{j=1}^{k} |\mathcal{M}_i^{(j)}|$.

In order to improve the performance, we follow a *multi-scale* approach where we compute feature tensors and local features from each input image at 3 different scales,

but still keeping a fixed number of local features from all scales according to strength. During re-ranking, we then perform spatial matching on all 9 combinations of query and database image scales and keep the combination with maximum similarity. Matching examples are shown in Figure 5. As post-processing, we apply *supervised whitening* to global descriptors as in [29] and query-time *diffusion* [12]. The latter is based on a nearest neighbor graph of the entire dataset X and is a second re-ranking process applied after spatial re-ranking. The precision of top-ranking images is important for diffusion [27], so spatial re-ranking is expected to help more its presence.

4. Experiments

In this section we evaluate the benefits of different parts of our *deep spatial matching* (DSM) and compare our results with the state of the art on standard benchmarks.

4.1. Experimental setup

Test sets. We use the medium and hard setups of the revisited $\mathcal{R}$Oxf and $\mathcal{R}$Par benchmarks [27]. We also use the large-scale benchmarks $\mathcal{R}$Oxf+$\mathcal{R}$1M and $\mathcal{R}$Par+$\mathcal{R}$1M, which are a combination of a set of 1M distractor images with the two small ones. We resize all images to a maximum size of 1024×1024. We evaluate performance by *mean average precision* (mAP) and *mean precision at 10* (mP@10), as defined by the protocol [27].

Networks. We use VGG16 [34] and Resnet101 [11], denoted simply as VGG (ResNet), or V (R) for short. In particular we use the versions trained by Radenovic *et al*. [29] with GeM pooling. We also re-train them with max-pooling, on the same dataset of 120k Flickr images and the same structure-from-motion pipeline [29]. Max-pooling is denoted by MAC [40] and re-training by *. ResNet has a resolution 4 times smaller than VGG. Therefore we remove the stride in the first $conv5$ convolutional layer and add a dilation factor of 2 in all following layers. We thus preserve the feature space while upsampling by 2. This upsampling requires no re-training and is denoted by ↑.

Global image representation. To rank images based on cosine similarity, we compute the multi-scale global representation described in section 3.5. We extract descriptors at three different scales, related by factors 1, $1/\sqrt{2}$, and $1/2$, and pooled from the last activation maps using max-pooling [40] (MAC) or generalized mean-pooling [29] (GeM). The descriptors are pooled over scales into a single representation by either GeM for networks using GeM, or average for networks using MAC.

Local feature detection. We use the MSER implementation of VLFEAT [41] to detect regions in the last activation map of the network. We set the minimum diversity to 0.7 and maximum variation to 0.5. We observed that the step Δ needs adjusting according to the network/dataset used. We do this by setting Δ to 60% of the cumulative histogram of the activation values over the dataset.

Method	Medium		Hard	
	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
R-MAC*	64.0	75.5	36.7	53.2
R-MAC*↑	63.9	75.5	35.6	53.3
R-GeM[29]	64.7	77.2	38.5	56.3
R-GeM[29]↑	65.3	77.3	39.6	56.6
R-MAC*+D	73.7	89.5	45.8	80.5
R-MAC*↑+D	73.9	89.9	45.6	81.0
R-GeM[29]+D	69.8	88.9	40.5	78.5
R-GeM[29]↑+D	70.1	89.1	41.5	78.9

Table 1. Impact of ResNet (R) activation upsampling (↑) on mAP in $\mathcal{R}$Oxf and $\mathcal{R}$Par [27]. MAC: max-pooling [40]; GeM: generalized-mean pooling [29]; D: diffusion [12]. All results with supervised whitening [21]. Citation specifies the origin of the network or *: our re-training.

Method	Medium				Hard			
	$\mathcal{R}$Oxf		$\mathcal{R}$Par		$\mathcal{R}$Oxf		$\mathcal{R}$Par	
	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
V	44.8	63.3	65.7	95.0	18.4	31.2	41.0	79.1
V+DSM	51.1	77.3	66.2	96.9	25.3	40.3	41.0	81.7
R↑	44.4	64.2	69.0	96.4	17.7	31.2	46.5	85.3
R↑+DSM	49.6	74.0	69.7	98.4	21.7	37.6	46.7	87.0
V+D	48.4	65.2	81.4	95.6	24.8	37.1	67.1	93.0
V+DSM+D	61.6	81.0	82.8	97.6	35.5	48.1	68.7	95.9
R↑+D	53.8	69.0	85.6	96.3	29.8	38.1	72.1	94.1
R↑+DSM+D	60.2	78.9	86.3	96.9	33.1	42.0	72.8	95.0

Table 2. Impact of the proposed *deep spatial matching* (DSM) on mAP and mP@10 on $\mathcal{R}$Oxf and $\mathcal{R}$Par [27] with *off-the shelf* (pre-trained on Imagenet [7]) VGG (V) and ResNet (R). ↑: upsampling; D: diffusion [12]. DSM: this work. All results with GeM pooling and supervised whitening.

Local image representation. To spatially verify images, we compute the multi-scale local representation introduced in section 3.5. We fit an ellipse to each MSER region and for each ellipse we keep the covariance matrix, center position, channel id and maximum value. We discard activation maps with more than 20 features detected on query images, and 10 on database images. We apply NMS to features of database images with IoU threshold 0.2, which is restrictive enough even for small features. We rank features over all scales according to activation value and we select the top-ranking 512 features on VGG and 2048 on ResNet.

Re-ranking. After initial ranking by cosine similarity, we perform spatial matching between the query and the 100 top-ranked images as described in section 3.5. Tentative correspondences originate from the same channels. We set the error threshold to 2 pixels (in the activation channel, not

the image) and the maximal scale change to 3. Finally, we use the number of inliers to re-rank the top 100 images.

Spatially verified diffusion. We use diffusion [12], denoted by D, as a second post-processing step after spatial verification. It is based on a nearest neighbor graph of the global descriptors of the entire dataset, which is computed off-line. It starts from the top ranked images and finds more similar images according to manifold similarity. Diffusion is very powerful but sensitive to the quality of the initial top-ranked results. Thanks to our spatial matching, these results are more accurate. We take our 10 top-ranking spatially verified images and we compute a new score that is the product of the number of inliers and the descriptor similarity scores. We select the top 5 of them to initiate diffusion.

4.2. Ablation experiments

Upsampling. Table 1 shows the effect of upsampling on retrieval. This is not significant on MAC pooling. On GeM however, it results in perfomance increase by up to 1 mAP point on the hard setup of $\mathcal{R}$Oxf, both with and without diffusion. This can be explained by the higher resolution of the activation maps.

Off-the-shelf networks. Our re-ranking can be applied to any network, even as pre-trained on Imagenet [7] (*off-the-shelf*). We use GeM pooling, which is better than MAC on such networks [27]. Table 2 shows the effect of DSM on $\mathcal{R}$Oxf and $\mathcal{R}$Par medium and hard setup. We improve results with and without diffusion. The gain is significant on $\mathcal{R}$Oxf, up to 13 mAP points on VGG-GeM with diffusion, medium setup. It is much smaller on $\mathcal{R}$Par, where the perfomance is already 20 to 40 mAP points higher than on $\mathcal{R}$Oxf.

Whitening. We investigate the efficiency of our re-ranking with multi-scale global descriptors that are whitened or not. We use supervised whitening as in [21, 28], denoted by W. This is more powerful than PCA whitening [13]. As shown in Table 3, we improve significantly on non-whitened descriptors with both networks on $\mathcal{R}$Oxf. We gain 3 to 4 mAP points, as well as increasing mP@10. On the other hand, whitening boosts cosine similarity search, and gains 5 to 10 mAP points. Our improvement is more marginal or we lose up to one mAP point in this case.

Inliers. To evaluate the quality of matching, we check how many inliers are found for positive and negative images. In particular, Fig. 6 shows the distribution of the number of inliers to all queries of $\mathcal{R}$Oxf with VGG-MAC for both positive and negative images. The distribution is similar over different networks and datasets. Negative images can be easily discriminated by having few inliers, but this may result in loosing positive ones. Contrary to conventional spatial matching, we do not use local descriptors. This is positive in terms of memory, but comes necessarily with lower quality of matches. However, the top-ranking spatially verified images *per query* are indeed accurate as indicated by mP@10, which is enough to initiate a better diffusion.

Method	Medium				Hard			
	$\mathcal{R}$Oxf		$\mathcal{R}$Par		$\mathcal{R}$Oxf		$\mathcal{R}$Par	
	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
V*	55.2	78.1	61.3	96.1	25.0	38.6	35.8	77.4
V*+DSM	58.2	83.4	61.9	98.9	28.4	46.6	36.2	80.4
V*+W	59.1	81.3	66.8	97.7	31.5	49.0	41.7	82.3
V*+W+DSM	60.0	84.3	67.0	98.6	32.5	53.1	42.0	82.3
R*↑	54.0	75.7	70.6	97.0	24.2	36.6	44.4	84.6
R*↑+DSM	57.4	80.4	70.9	98.7	28.4	42.6	44.3	84.9
R*↑+W	63.9	85.2	75.5	98.4	35.6	52.6	53.3	89.6
R*↑+W+DSM	62.7	83.7	75.7	98.7	35.4	51.6	53.1	88.6

Table 3. Impact of the *supervised whitening* (W) [21] on mAP and mP@10 on $\mathcal{R}$Oxf and $\mathcal{R}$Par [27]. Results with VGG (V) and ResNet (R), both with MAC pooling; ↑: upsampling; D: diffusion [12]; DSM: this work. Citation specifies the origin of the network or *: our re-training.

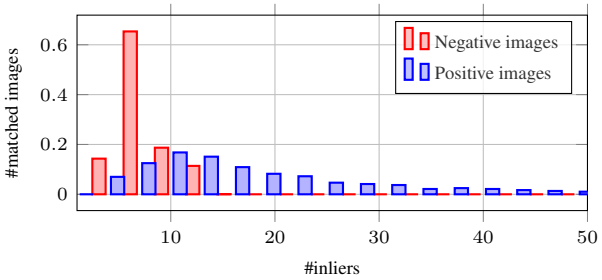

Figure 6. Distribution of number of inliers for positive and negative database images over all queries of $\mathcal{R}$Oxf, using VGG-MAC.

4.3. Comparison with the state-of-the-art

We conduct an extensive comparison of our method with baselines and additional 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. We collect all results in Table 4.

Most baselines are improved by re-ranking, and all experiments on $\mathcal{R}$Oxf show consitent increase in performance. However, re-ranking is not perfect, as seen in Fig. 6. In few cases the performance drops after re-ranking by up to one mAP point on $\mathcal{R}$Par, in particular with the upsampled ResNet-GeM. We attribute the loss to two factors. One is a limited "vocabulary", based only on 512 or 2048 activation maps. The other is the fact that activation maps are highly correlated. This is exploited by whitening of the global descriptors, but tends to create correlated features.

The performance is improved significantly when diffusion is initiated from top-ranked spatially verified images. Diffusion only needs few relevant images, and we are able to provide these images thanks to spatial matching. We im-

Method	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	
	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10	mAP	mP@10
"DELF-ASMK*+SP" [27]	67.8	87.9	53.8	81.1	76.9	99.3	57.3	98.3	43.1	62.4	31.2	50.7	55.4	93.4	26.4	75.7
R-RMAC[10] [27]	60.9	78.1	39.3	62.1	78.9	96.9	54.8	93.9	32.4	50.0	12.5	24.9	59.4	86.1	28.0	70.0
V-MAC[28]	58.4	81.1	39.7	68.6	66.8	97.7	42.4	92.6	30.5	48.0	17.9	27.9	42.0	82.9	17.7	63.7
V-MAC*	59.1	81.3	40.2	68.1	66.8	97.7	42.1	92.0	31.5	49.0	17.8	28.4	41.7	82.3	17.4	63.6
V-MAC*+DSM	60.0	84.3	42.2	71.0	67.0	98.6	42.5	94.7	32.5	53.1	19.4	31.6	42.0	82.3	17.7	66.0
R-MAC*↑	63.9	85.2	43.2	69.6	75.5	98.4	50.1	95.3	35.6	52.6	17.7	31.4	53.3	89.6	22.4	71.6
R-MAC*↑+DSM	62.7	83.7	44.4	72.3	75.7	98.7	50.4	96.4	35.4	51.6	20.6	32.3	53.1	88.6	22.7	72.1
V-GeM[29]	61.9	82.7	42.6	68.1	69.3	97.9	45.4	94.1	33.7	51.0	19.0	29.4	44.3	83.7	19.1	64.9
V-GeM[29]+DSM	63.0	85.5	43.9	72.9	69.2	98.4	45.4	94.7	34.5	54.0	19.9	32.9	43.9	82.7	19.5	67.6
R-GeM[29]	64.7	84.7	45.2	71.7	77.2	98.1	52.3	95.3	38.5	53.0	19.9	34.9	56.3	89.1	24.7	73.3
R-GeM[29]↑	65.3	86.3	46.1	73.4	77.3	98.3	52.6	95.4	39.6	54.6	22.2	36.4	56.6	89.4	24.8	73.6
R-GeM[29]↑+DSM	65.3	87.1	47.6	76.4	77.4	99.1	52.8	96.7	39.2	55.3	23.2	37.9	56.2	89.9	25.0	74.6
Diffusion																
"DELF-HQE+SP" [27]	73.4	88.2	60.6	79.7	84.0	98.3	65.2	96.1	50.3	67.2	37.9	56.1	69.3	93.7	35.8	69.1
"DELF-ASMK*+SP"→D† [27]	75.0	87.9	68.7	83.6	**90.5**	98.0	**86.6**	98.1	48.3	64.0	39.4	55.7	**81.2**	95.6	**74.2**	94.6
V-MAC*+D	67.7	86.1	56.8	78.6	85.6	97.6	78.6	96.4	39.8	51.1	29.4	46.0	73.9	94.1	62.4	91.9
V-MAC*+DSM+D	72.0	90.6	59.2	80.1	86.4	98.9	79.3	97.1	43.9	56.0	32.0	47.4	75.1	95.4	63.4	92.9
R-MAC*↑+D	73.9	87.9	61.3	80.6	89.9	96.1	83.0	95.1	45.6	62.2	31.9	48.4	81.0	94.3	68.6	91.9
R-MAC*↑+DSM+D	**76.9**	90.7	65.7	83.9	90.1	96.4	84.0	95.3	**49.4**	64.7	35.7	51.3	**81.2**	93.3	70.1	92.6
V-GeM[29]+D	69.6	84.7	60.4	79.4	85.6	97.1	80.7	97.1	41.1	51.1	33.1	49.6	73.9	93.7	65.3	93.1
V-GeM[29]+DSM+D	72.8	89.0	63.2	83.7	85.7	96.1	80.1	95.7	45.4	57.1	35.4	53.7	74.2	93.3	65.2	91.9
R-GeM[29]+D	69.8	84.0	61.5	77.1	88.9	96.9	84.9	95.9	40.5	54.4	33.1	48.2	78.5	94.6	71.6	93.7
R-GeM[29]↑+D	70.1	84.3	67.5	79.0	89.1	97.3	85.0	96.6	41.5	54.4	39.6	53.0	78.9	95.1	72.0	94.1
R-GeM[29]↑+DSM+D	75.0	89.6	**70.2**	84.5	89.3	97.1	84.8	95.3	46.2	60.6	**41.9**	54.9	79.3	95.1	72.0	93.4

Table 4. mAP and mP@10 *state-of-the-art* on the full benchmark [27]. We use VGG (V) and ResNet (R), with MAC or GeM pooling. ↑: upsampling; *: our re-training; D: diffusion [12]. DSM: this work. Results citing [27] are as reported in that work and are combining DELF [24], ASMK* [38] and HQE [39]. SP: spatial matching [26]; D†: diffusion on the graph obtained by [10]. The remaining citations specify where we took the trained network from.

prove on most datasets, networks and pooling options in this case. The gain is more pronounced on $\mathcal{R}$Oxf, and is up to 5 mAP or 6 mP@10 points.

Finally, the proposed method with spatially verified diffusion outperforms approaches based on deep local features in a number of cases. In particular, we compare with the best performing and expensive version of DELF [24] proposed and evaluated by [27]. Apart from spatial verification by [26] on the 100 top-ranking images, this version is using two independent representations. One is ASMK* [38], based on 128-dimensional descriptors of 1000 DELF features per image, and used for initial ranking. Another is a global descriptor obtained by ResNet-RMAC [10], and used for diffusion (D†) after spatial verification as in this work. By contrast, our global and local representations are obtained from the same activation tensor, and we do not use any local descriptors or their quantized versions.

5. Discussion

Our experiments validate that the proposed representation for spatial verification achieves state-of-art performance across a number of different datasets, networks and pooling mechanisms. This representation arises naturally in the existing convolutional activations of off-the-shelf or fine-tuned networks, without any particular effort to detect local features or extract local descriptors on image patches. It does not require any network modification or retraining. It is a significant step towards bridging the gap between global descriptors, which are efficient for initial ranking using nearest neighbor search, and local representations, which are compatible with spatial verification.

Of course, the activation channels are not the most appropriate by construction to replace a visual vocabulary. This means that our representation, while being very compact, is not as powerful as storing *e.g.* hundreds of local descriptors per image. Nonetheless, we still demonstrate that it is enough to provide high-quality top-ranking images to initiate diffusion, which then brings excellent results.

Acknowledgments This work was supported by the GAČR grant 19-23165S and the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

References

[1] Jon Almazan, Bojana Gajic, Naila Murray, and Diane Larlus. Re-id done right: towards good practices for person re-identification. *arXiv preprint arXiv:1801.05339*, 2018. 2

[2] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015. 1

[3] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *ECCV*, 2014. 2

[4] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *NIPS*, pages 2414–2422, 2016. 1, 2, 3

[5] Ondřej Chum, Jiri Matas, and Josef Kittler. Locally optimized ransac. In *DAGM Symposium on Pattern Recognition*, page 236. Springer Verlag, 2003. 5

[6] Ondřej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, October 2007. 2

[7] Wei Dong, Richard Socher, Li Li-Jia, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, June 2009. 6, 7

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *CVPR*, pages 2758–2766, 2015. 3

[9] Martin A. Fischler and Robert C. Bolles. Random sample consensus. *Communications of ACM*, 6(24):381–395, 1981. 5

[10] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 124(2):237–254, Sep 2017. 1, 2, 3, 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[12] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*, 2017. 2, 6, 7, 8

[13] Hervé Jégou and Ondrej Chum. Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening. In *ECCV*, October 2012. 7

[14] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, June 2010. 1

[15] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 685–701, Cham, 2016. Springer International Publishing. 1, 2

[16] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *CVPR*, 2017. 3

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 2

[18] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*. 2014. 1, 2, 3

[19] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 1, 2

[20] Jiri Matas, Ondrej Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, September 2002. 4

[21] Krystian Mikolajczyk and Jiri Matas. Improving descriptors for fast tree matching by optimal linear projection. In *CVPR*, 2007. 6, 7

[22] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 1, 2

[23] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, June 2006. 1

[24] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Largescale image retrieval with attentive deep local features. In *ICCV*, 2017. 1, 2, 3, 8

[25] Michal Perdoch, Ondrej Chum, and Jiri Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, June 2009. 1

[26] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, June 2007. 1, 2, 5, 8

[27] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, 2018. 2, 6, 7, 8

[28] Filip Radenović, Giorgos Tolias, and Ondřej Chum. CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 7, 8

[29] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Trans. PAMI*, 2018. 1, 2, 3, 6, 8

[30] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. Visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574*, 2014. 2

[31] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, 2018. 1, 2, 3

[32] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood Consensus Networks. In *NIPS*, Montréal, Canada, December 2018. 2

[33] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, Oct 2017. 2

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014. 3, 6

[35] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1

[36] Giorgos Tolias and Yannis Avrithis. Speeded-up, relaxed spatial matching. In *ICCV*, 2011. 2

[37] Giorgios Tolias, Yannis Avrithis, and Hervé Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, December 2013. 2

[38] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: aggregation across single and multiple images. *IJCV*, 2016. 8

[39] Giorgos Tolias and Herve Jegou. Visual query expansion with or without geometry: Refining local descriptors by feature aggregation. *Pattern Recognition*, 2014. 2, 8

[40] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *ICLR*, 2016. 1, 2, 3, 6

[41] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 6

[42] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1, 2

[43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, June 2016. 2