Local Features and Visual Words Emerge in Activations

Oriane Siméoni\textsuperscript{1}  Yannis Avrithis\textsuperscript{1}  Ondřej Chum\textsuperscript{2}

\textsuperscript{1}Univ Rennes, Inria, CNRS, IRISA \textsuperscript{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 \cite{35, 23}. The matching score was first approximated by a similarity of bag of words \cite{35} or aggregated descriptors \cite{14}, and then re-ranked by efficient spatial verification \cite{26, 25}.

Recently, image retrieval is dominated by convolutional neural networks (CNNs) \cite{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.

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 \cite{18, 4, 31}.

In general, the CNN activation tensor size depends on the number of channels and the image size. It 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 \cite{2}, max pooling \cite{40}, generalized-mean pooling \cite{29}, and others \cite{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
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 approach 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].
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 ROx 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 $P_1 := d(A_1), P_2 := d(A_2)$ respectively. The feature detector $d$, discussed in section 3.3, operates independently per channel and each local feature collection $P$ is a list of sets, one per channel. Local features are represented as discussed in section 3.4.

Then, the two local feature collections $P_1, P_2$ undergo spatial matching, denoted as $g$ and discussed in section 3.5, returning a collection of inliers $M$ and a geometric trans-
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 \( P_1, P_2 \) extracted by detector \( d \) undergo spatial matching \( g \), resulting in a collection of inliers \( M \). 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 \( P_1, P_2 \) are stored and \( g \) applies directly at re-ranking.

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 intensity (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.

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 \( v := \text{pool}_{r \in R} A^{(j)}(r) \). 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 \( 2 \times 1 \) mean (position) vector \( \mu \) and \( 2 \times 2 \) covariance matrix (local shape) \( \Sigma \). 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.
by tuple \( p := (\mu, \Sigma, v) \). Finally, we collect local features \( P = (P^{(1)}, \ldots, P^{(k)}) \) where \( P^{(j)} \) contains the local features \( p \) found in channel \( j \). The entire operation is denoted by \( P := d(A) \).

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 \( P_1, P_2 \) 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 \( P_1, P_2 \) that are consistent with this transformation. Matching is based on correspondences, i.e., pairs of local features \( e = (p_1, p_2) \) from the two images. We allow pairs only between local features of the same channel, that is, \( p_1, p_2 \) are in \( P_1^{(j)}, P_2^{(j)} \) 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, \( C := (P_1^{(1)} \times P_2^{(1)}, \ldots, P_1^{(k)} \times P_2^{(k)}) \).

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 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 \( M \) is returned. The operation is denoted by \( \langle M, T \rangle := g(P_1, P_2) \) and \( M = (M^{(1)}, \ldots, M^{(k)}) \) is a list of sets of inliers, one per channel.

### 3.6. Retrieval and re-ranking

In an image retrieval scenario, \( n \) database images \( X = \{x_1, \ldots, x_n\} \) are given in advance. For each image \( x_i \) with feature tensor \( A_i \), its local features \( P_i := d(A_i) \) 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 \( P := d(A) \) and global descriptor \( z \), we first rank \( \{z_1, \ldots, z_n\} \) by cosine similarity to \( z \), and then the top-ranking images undergo spatial matching against \( P \) according to \( \langle M_i, T_i \rangle := g(P, P_i) \) and are re-ranked according to similarity function \( s(M_i) \). The most common choice, which we also follow in this work, is the number of inliers found, \( s(M_i) := \sum_{j=1}^{k} |M_i^{(j)}| \).

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 ROxf and RPar benchmarks [27]. We also use the large-scale benchmarks ROxf+R1M and RPar+R1M, 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 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/√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.

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 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 a

<table>
<thead>
<tr>
<th>Method</th>
<th>Medium ROxf RPar</th>
<th>Hard ROxf RPar</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-MAC*</td>
<td>64.0 75.5</td>
<td>36.7 53.2</td>
</tr>
<tr>
<td>R-MAC*↑</td>
<td>63.9 75.5</td>
<td>35.6 53.3</td>
</tr>
<tr>
<td>R-GeM[29]</td>
<td>64.7 77.2</td>
<td>38.5 56.3</td>
</tr>
<tr>
<td>R-GeM[29]↑</td>
<td>65.3 77.3</td>
<td>39.6 56.6</td>
</tr>
<tr>
<td>R-MAC*+D</td>
<td>73.7 89.5</td>
<td>45.8 80.5</td>
</tr>
<tr>
<td>R-MAC*↑+D</td>
<td>73.9 89.9</td>
<td>45.6 81.0</td>
</tr>
<tr>
<td>R-GeM[29]+D</td>
<td>69.8 88.9</td>
<td>40.5 78.5</td>
</tr>
<tr>
<td>R-GeM[29]↑+D</td>
<td>70.1 89.1</td>
<td>41.5 78.9</td>
</tr>
</tbody>
</table>

Table 1. Impact of ResNet (R) activation upsampling (↑) on mAP in ROxf and RPar [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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Medium ROxf RPar</th>
<th>Hard ROxf RPar</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>44.8 63.3</td>
<td>18.4 31.2</td>
</tr>
<tr>
<td>V+DSM</td>
<td>51.1 77.3</td>
<td>25.3 40.3</td>
</tr>
<tr>
<td>R↑</td>
<td>44.4 64.2</td>
<td>17.7 31.2</td>
</tr>
<tr>
<td>R↑+DSM</td>
<td>49.6 74.0</td>
<td>21.7 37.6</td>
</tr>
<tr>
<td>V+D</td>
<td>48.4 65.2</td>
<td>24.8 37.1</td>
</tr>
<tr>
<td>V+DSM+D</td>
<td>61.6 81.0</td>
<td>35.5 48.1</td>
</tr>
<tr>
<td>R↑+D</td>
<td>53.8 69.0</td>
<td>29.8 38.1</td>
</tr>
<tr>
<td>R↑+DSM+D</td>
<td>60.2 78.9</td>
<td>33.1 42.0</td>
</tr>
</tbody>
</table>

Table 2. Impact of the proposed deep spatial matching (DSM) on mAP and mP@10 on ROxf and RPar [27] with off-the-shelf (pre-trained on Imagenet) VGG (V) and ResNet (R). ↑: upsampling; D: diffusion [12]. DSM: this work. All results with GeM pooling and supervised whitening.
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 performance increase by up to 1 mAP point on the hard setup of R0xf, 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 R0xf and RPar medium and hard setup. We improve results with and without diffusion. The gain is significant on R0xf, up to 13 mAP points on VGG-GeM with diffusion, medium setup. It is much smaller on RPar, where the performance is already 20 to 40 mAP points higher than on R0xf.

**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 R0xf. 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 R0xf 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.

### 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 R0xf, R0xf+R1M, RPar and RPar+R1M. We collect all results in Table 4.

Most baselines are improved by re-ranking, and all experiments on R0xf show consistent 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 RPar, 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-

<table>
<thead>
<tr>
<th>Method</th>
<th>Medium</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R0xf</td>
<td>RPar</td>
</tr>
<tr>
<td></td>
<td>mAP</td>
<td>mP@10</td>
</tr>
<tr>
<td>V*</td>
<td>55.2</td>
<td>78.1</td>
</tr>
<tr>
<td>V*+DSM</td>
<td>58.2</td>
<td>83.4</td>
</tr>
<tr>
<td>V*+W</td>
<td>59.1</td>
<td>81.3</td>
</tr>
<tr>
<td>V*+W+DSM</td>
<td>60.0</td>
<td>84.3</td>
</tr>
<tr>
<td>R*↑</td>
<td>54.0</td>
<td>75.7</td>
</tr>
<tr>
<td>R*↑+DSM</td>
<td>57.4</td>
<td>80.4</td>
</tr>
<tr>
<td>R*↑+W</td>
<td>63.9</td>
<td>85.2</td>
</tr>
<tr>
<td>R*↑+W+DSM</td>
<td>62.7</td>
<td>83.7</td>
</tr>
</tbody>
</table>

Table 3. Impact of the supervised whitening (W) [21] on mAP and mP@10 on R0xf and RPar [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 R0xf, using VGG-MAC.
prove on most datasets, networks and pooling options in this case. The gain is more pronounced on ROxf; 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 and matching, and local descriptors, 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 OPVVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”.

<table>
<thead>
<tr>
<th>Method</th>
<th>Medium ROxf</th>
<th>Medium ROxf+K1M</th>
<th>Medium KPar</th>
<th>Medium KPar+K1M</th>
<th>Hard ROxf</th>
<th>Hard ROxf+K1M</th>
<th>Hard KPar</th>
<th>Hard KPar+K1M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
<td>mAP mP@10</td>
</tr>
<tr>
<td>&quot;DELF-ASMK*+SP&quot; [27]</td>
<td>67.8 87.9</td>
<td>53.8 81.1</td>
<td>76.9 99.3</td>
<td>57.3 98.3</td>
<td>43.1 62.4</td>
<td>51.2 50.7</td>
<td>55.4 93.4</td>
<td>26.4 75.7</td>
</tr>
<tr>
<td>R-RMAC [10] [27]</td>
<td>60.9 78.1</td>
<td>39.3 62.1</td>
<td>78.9 96.9</td>
<td>54.8 93.9</td>
<td>32.4 50.0</td>
<td>12.5 24.9</td>
<td>59.4 86.1</td>
<td>28.0 70.0</td>
</tr>
<tr>
<td>V-MAC [29]</td>
<td>58.4 81.1</td>
<td>39.7 68.6</td>
<td>66.8 97.7</td>
<td>42.4 92.6</td>
<td>30.5 48.0</td>
<td>17.9 27.9</td>
<td>42.0 82.9</td>
<td>17.7 63.7</td>
</tr>
<tr>
<td>V-MAC*</td>
<td>59.1 81.3</td>
<td>40.2 68.1</td>
<td>66.8 97.7</td>
<td>42.1 92.0</td>
<td>31.5 49.0</td>
<td>17.8 28.4</td>
<td>41.7 82.3</td>
<td>17.4 63.6</td>
</tr>
<tr>
<td>V-MAC*+DMS</td>
<td>60.0 84.3</td>
<td>42.2 71.0</td>
<td>67.0 98.6</td>
<td>42.5 94.7</td>
<td>32.5 53.1</td>
<td>19.4 31.6</td>
<td>42.0 82.3</td>
<td>17.7 66.0</td>
</tr>
<tr>
<td>R-MAC*↑</td>
<td>63.9 85.2</td>
<td>43.2 69.6</td>
<td>75.5 98.4</td>
<td>50.1 95.3</td>
<td>35.6 52.6</td>
<td>17.7 31.4</td>
<td>53.3 89.6</td>
<td>22.4 71.6</td>
</tr>
<tr>
<td>R-MAC*↑+DMS</td>
<td>62.7 83.7</td>
<td>44.4 72.3</td>
<td>75.7 98.7</td>
<td>50.4 96.4</td>
<td>35.4 51.6</td>
<td>20.6 32.3</td>
<td>53.1 88.6</td>
<td>22.7 72.1</td>
</tr>
<tr>
<td>V-GeM [29]</td>
<td>61.9 82.7</td>
<td>42.6 68.1</td>
<td>69.3 97.9</td>
<td>45.4 94.1</td>
<td>33.7 51.0</td>
<td>19.0 29.4</td>
<td>44.3 83.7</td>
<td>19.1 64.9</td>
</tr>
<tr>
<td>V-GeM [29]+DMS</td>
<td>63.0 85.5</td>
<td>43.9 72.9</td>
<td>69.2 98.4</td>
<td>45.4 94.7</td>
<td>34.5 54.0</td>
<td>19.9 32.9</td>
<td>43.9 82.7</td>
<td>19.5 67.6</td>
</tr>
<tr>
<td>R-GeM [29]</td>
<td>64.7 84.7</td>
<td>45.2 71.7</td>
<td>77.2 98.1</td>
<td>52.3 95.3</td>
<td>38.5 53.0</td>
<td>19.9 34.9</td>
<td>56.3 89.1</td>
<td>24.7 73.3</td>
</tr>
<tr>
<td>R-GeM [29]↑</td>
<td>65.3 86.3</td>
<td>46.1 73.4</td>
<td>77.3 98.3</td>
<td>52.6 95.4</td>
<td>39.6 54.6</td>
<td>22.2 36.4</td>
<td>56.6 88.4</td>
<td>24.8 73.6</td>
</tr>
<tr>
<td>V-GeM [29]↑+DSM</td>
<td>65.3 87.1</td>
<td>47.6 76.4</td>
<td>77.4 98.1</td>
<td>52.8 96.7</td>
<td>39.2 55.3</td>
<td>23.2 37.9</td>
<td>56.2 89.9</td>
<td>25.0 74.6</td>
</tr>
</tbody>
</table>

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 on the graph obtained by [27]. 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.
References


