Aggregating Deep Convolutional Features for Image Retrieval

Artem Babenko
Yandex
Moscow Institute of Physics and Technology
artem.babenko@phystech.edu

Victor Lempitsky
Skolkovo Institute of Science and Technology
(Skoltech)
lempitsky@skoltech.ru

Abstract

Several recent works have shown that image descriptors produced by deep convolutional neural networks provide state-of-the-art performance for image classification and retrieval problems. It has also been shown that the activations from the convolutional layers can be interpreted as local features describing particular image regions. These local features can be aggregated using aggregation approaches developed for local features (e.g. Fisher vectors), thus providing new powerful global descriptors.

In this paper we investigate possible ways to aggregate local deep features to produce compact global descriptors for image retrieval. First, we show that deep features and traditional hand-engineered features have quite different distributions of pairwise similarities, hence existing aggregation methods have to be carefully re-evaluated. Such re-evaluation reveals that in contrast to shallow features, the simple aggregation method based on sum pooling provides arguably the best performance for deep convolutional features. This method is efficient, has few parameters, and bears little risk of overfitting when e.g. learning the PCA matrix. Overall, the new compact global descriptor improves the state-of-the-art on four common benchmarks considerably.

1. Introduction

Image descriptors based on the activations within deep convolutional neural networks (CNNs) [13] have emerged as state-of-the-art generic descriptors for visual recognition [18, 21, 4]. Several recent works [2, 21, 7] proposed to use the outputs of last fully-connected network layers as global image descriptors and demonstrate their advantage over prior state-of-the-art when the dimensionality of descriptors is limited.

Recently, research attention shifted from the features extracted from the fully-connected layers to the features from the deep convolutional layers of CNNs [5, 22, 14] (below we refer to these features as *deep convolutional features*). These features possess very useful properties, e.g. they can be extracted straightforwardly and efficiently from an image of any size and aspect ratio. Also, features from the convolutional layers have a natural interpretation as descriptors of local image regions corresponding to receptive fields of the particular features. Such features can thus be considered as an analogy of "shallow" hand-crafted features such as dense SIFT [16, 26]. Perhaps inspired by this analogy, [15] suggested to use such features to identify meaningful object parts, while [5] proposed to use Fisher vector [23] constructed on these *local* features to produce a *global* image descriptor that provides state-of-the-art classification performance on external datasets.

The focus of this paper is image retrieval and in particular the construction of global descriptors for image retrieval. Following recent papers [2, 7, 21, 22], we consider descriptors based on activations of pretrained deep CNNs, and specifically deep convolutional layers of CNNs. Given the emerging perception of the features in the convolutional layers as "new dense SIFT" [15, 22, 5, 14], it seems natural to reuse state-of-the-art embedding-and-aggregation frameworks for dense SIFT such as VLAD [9], Fisher vectors [19] or triangular embedding [10], and apply them to deep convolutional features. Our first contribution is the evaluation of these approaches (specifically, Fisher vectors and triangular embeddings) alongside simpler aggregation schemes such as sum pooling and max pooling.

Perhaps surprisingly, we have found that the relative performance of the aggregation methods for deep convolutional features is rather different from the case of shallow descriptors. In particular, a simple global descriptor based on sum pooling aggregation without high-dimensional embedding and with simple postprocessing performs remarkably well. Such descriptors based on sum-pooled convolutional features (*SPoC* descriptors) improve considerably the state-of-the-art for compact global descriptors on standard retrieval datasets, and perform much better than deep global descriptors for retrieval previously suggested in [2, 7, 22]. In addition to the excellent retrieval accuracy, SPoC features are efficient to compute, simple to implement and have al-

most no hyperparameters to tune.

Importantly, SPoC features perform better than Fisher vector and triangular embeddings of deep convolutional features. This is in sharp contrast to the dense SIFT case, where sum pooling of raw features does not produce a competitive global descriptor. We further investigate why the performance of deep convolutional features is different from shallow features (SIFT), and show that the preliminary embedding step is not needed for deep convolutional features because of their higher discriminative ability and different distribution properties. Both qualitative explanation and experimental confirmations for this claim are provided.

Overall, this paper introduces and evaluates a new simple and compact global image descriptor and investigates the reasons underlying its success. The descriptor outperforms the existing methods on the common retrieval benchmarks. For example, the performance of 0.66 mAP on the Oxford dataset with 256-dimensional representation (when using entire images during query process) is achieved.

2. Related work

Descriptor aggregation. The problem of aggregating a set of local descriptors (such as SIFT) into a global one has been studied extensively. The best known approaches are VLAD [9], Fisher Vectors [19], and, more recently, triangular embedding [10], which constitutes state-of-the-art for "hand-crafted" features like SIFT.

Let us review the ideas behind these schemes (using the notation from [10]). An image I is represented by a set of features $\{x_1, \dots, x_n\} \subset \mathbf{R}^d$. The goal is to combine these features into a discriminative global representation $\psi(I)$. Discriminativity here means that the representations of two images with the same object or scene are more similar (e.g. w.r.t. cosine similarity) than the representations of two unrelated images. Apart from discriminativity, most applications have a preference towards more compact global descriptors, which is also a focus of our work here. Consequently, the dimensionality of $\psi(I)$ is reduced by PCA followed by certain normalization procedures.

The common way to produce a representation $\psi(I)$ includes two steps, namely *embedding* and *aggregation* (optionally followed by PCA). The embedding step maps each individual feature x into a higher dimensional vector $\phi(x) \in \mathbf{R}^D$. Then the aggregation of mapped features $\{\phi(x_1), \dots, \phi(x_n)\} \subset \mathbf{R}^D$ is performed. One possible choice for this step is a simple summation $\psi(I) = \sum \phi(x_i)$ but more advanced methods (e.g. *democratic kernel* [10]) are possible.

The existing frameworks differ in the choice of the mapping ϕ. For example, VLAD precomputes a codebook of K centroids $\{c_1, \dots, c_K\}$ and then maps x to vector $\phi_{\text{VL}}(x) = [0 \ 0 \ \dots, (x - c_k) \ \dots, 0] \in \mathbf{R}^{K \times d}$, where k is the number of the closest centroid to x. The pipeline for Fisher vector embedding is similar except that it uses the soft probabilistic quantization instead of hard quantization in VLAD. It also includes the second-order information about the residuals of individual features into embedding. Triangulation Embedding [10] also uses cluster centroids and embeds an individual feature x by a concatenation of normalized differences between it and cluster centroids $\phi_{\text{TE}}(x) = \left[\frac{x-c_1}{||x-c_1||}, \dots, \frac{x-c_K}{||x-c_K||}\right]$. Then the embeddings $\phi_{\text{TE}}(x)$ are centered, whitened and normalized.

The rationale behind the embedding step is to improve the discriminative ability of individual features. Without such embedding, a pair of SIFT features x_i, x_j coming from unrelated images have a considerable chance of having a large value of the scalar product $\langle x_i, x_j \rangle$. This becomes a source of accidental false positive matches between local features, and, if the dataset is big enough, between images (as the similarity between resulting global descriptors is aggregated from similarities between pairs of local features [3, 25]). The embedding methods $\phi(\cdot)$ are typically designed to suppress such false positives. For instance, VLAD embedding suppresses all matches between pairs of features that are adjacent to different centroids in the codebook (making the corresponding scalar product zero). Similar analysis can be performed for other embeddings.

Suppressing false positives with high-dimensional mappings has certain drawbacks. First, such mapping can also suppress true positive matches between local features. Second, the embedding usually includes learning a lot of parameters that can suffer from overfitting if the statistics of training and test sets differ. Likewise, as the representations $\psi(I)$ can be very high-dimensional, it may require hold-out data with similar statistics to learn reliable PCA and whitening matrices. For this reason [10] proposes to use PCA rotation and power-normalization instead of whitening. Finally, high-dimensional embeddings are computationally intense compared to simpler aggregation schemes.

Despite these drawbacks, high-dimensional embeddings are invariably used with features like SIFT, since without them the discriminativity of the resulting global descriptors is unacceptingly low. In this paper, we demonstrate that in contrast to SIFT, the similarities of raw deep convolutional features are reliable enough to be used without embedding. Simple sum-pooling aggregation performed on unembedded features thus provides the performance which is comparable with high-dimensional embeddings. Eliminating the embedding step simplifies the descriptor, leads to faster computation, avoids problems with overfitting, and overall leads to a new state-of-the-art compact descriptor for image retrieval.

Deep descriptors for retrieval. Several prior works have considered the use of deep features for image retrieval. Thus, the seminal work [12] have presented qualitative examples of retrieval using deep features extracted from fully-

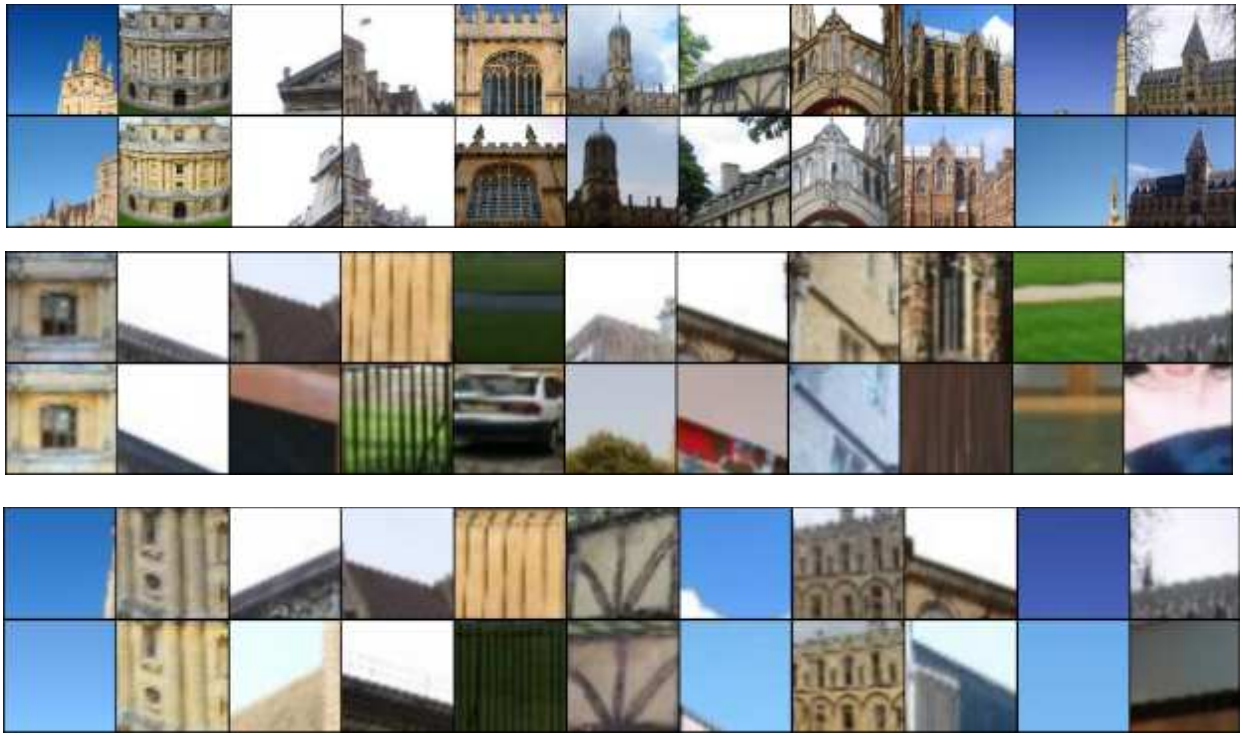

Figure 1. Randomly selected examples of image patches that are matched by individual deep features (top row), by original SIFT features (middle row) or by Fisher Vector-embedded SIFT features (bottom row). For deep features only the centers of corresponding receptive fields are shown. Overall, the matches produced by deep features has much lower false positive rate.

connected layers. After that, [2] has extensively evaluated the performance of such features with and without fine-tuning on related dataset, and overall reported that PCA-compressed deep features can outperform compact descriptors computed on traditional SIFT-like features.

Simultaneously, in [7] an even more performant desriptors were suggested based on extracting different fragments of the image, passing them through a CNN and then using VLAD-embedding [9] to aggregate the activations of a fully-connected layer. Related to that, the work [21] reported very good retrieval results using sets of few dozen features from fully-connected layers of a CNN, without aggregating them into a global desriptor.

Finally, the recent works [1, 22] evaluated image retrieval descriptors obtained by the max pooling aggregation of the last convolutional layer. Here, we show that using sum pooling to aggregate features on the last convolutional layer leads to much better performance. This is consistent with the interpretation of sum pooling aggregation as an implementation of the simplest match kernel [3], which is lacking in the case of max pooling.

Overall, compared to previous works [2, 7, 21, 1, 22] we show that a number of design choices within our descrptor (SPoC) lead to a big boost in descriptor accuracy and efficiency. Compared to those works, we also discuss and analyze the connection to the body of work on descriptor aggregation and evaluate several important aggregation alternatives.

3. Deep features aggregation

In this section, we first compare the distribution properties of deep convolutional features and SIFTs and highlight their differences. Based on these differences, we propose a new global image descriptor that avoids the embedding step necessary for SIFTs and discuss several design choices associated with this descriptor.

In our experiments, deep convolutional features are extracted by passing an image I through a pretrained deep network, and considering the output of the last convolutional layer. Let this layer consist of C feature maps each having height H and width W. Then the input image I is represented with a set of $H \times W$ C-dimensional vectors, which are the deep convolutional features we work with.

3.1. Properties of local feature similarities

As was analysed in e.g. [10] individual similarities of raw SIFT features are not reliable, i.e. unrelated image patches can result in very close SIFT features. Deep features are expected to be much more powerful as they are learned from massive amount of data in a supervised man-

ner. To confirm this, we have performed a comparison of the properties of similarities computed on the features in the form of two experiments.

Experiment 1 looks at patches matched by the three types of descriptors (Figure 1). To find these patches we proceed as follows:

- For each image in the Oxford Buildings dataset we extract both deep features and dense SIFT features.
- We embed SIFT features via Fisher vector embedding with 64 components.
- For each feature type (deep convolutional, original SIFT, embedded SIFT), for each query image we compute cosine similarity between its features and the features of all other images in the dataset.
- We consider random feature pairs from the top ten list for each image in terms of their similarities and visualize the corresponding image patches (full receptive field for original and embedded SIFT features, the center of the receptive field for deep convolutional features).

Figure 1 shows the random subset of the feature pair selected with such procedure (one randomly-chosen feature pair per Oxford building), with the top row corresponding to matching based on deep convolutional features, the middle to original dense SIFT, and the bottom to embedded SIFT. As expected, matches produced by deep features have much fewer obvious false positives among them, as they often correspond to the same object with noticeable tolerance to illumination/viewpoint changes and small shifts. SIFT-based matches are significantly worse and many of them correspond to unrelated image patches. The embedding of SIFT features by Fisher vector improves the quality of matches but still performs worse than deep features.

Experiment 2. We also investigate the statistics of high-dimensional distributions for deep convolutional features and dense SIFTs. Most of all we are interested in the distribution of deep features with largest norms as these features contribute most to a global descriptor. We also observe them to be the most discriminative by the following experiment. We performed retrieval by sum pooling descriptor but we aggregated only (1) 1% random features (2) 1% of features which had the largest norm. The mAP score for the Oxford Buildings dataset [20] for (1) was only 0.09, which was much smaller than mAP for (2), 0.34. This verifies that features with large norms are much more discriminative than random features.

For different types of features we want to investigate the reliability of matches produced by their individual similarities. To do this, we compare distances from each point to its closest neighbors with distances to random points in the dataset. In more details, we perform the following. From each query image, we extract ten deep features with maximum norms and for each of them compute the distances to all deep convolutional features of other images. Then we plot a graph which demonstrates how the distance to the k-th neighbor depends on its index k. For every query feature, distances are normalized by dividing by a median of all distances between the given feature and all features from other images.

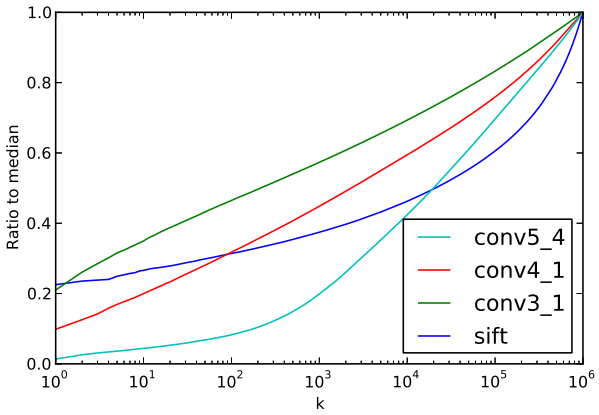

Figure 2. The average ratio between the distances to the kth neighbor and the median distance to all features for dense SIFT and deep convolutional features with the highest norm from three convolutional layers. The features from the last convolutional layer tend to have much closer neighbors (hence much smaller ratios) despite having higher dimensionality thus reflecting the differences in the spatial distribution of the two types of features in the corresponding high-dimensional spaces.

We perform this procedure for three types of convolutional features extracted from the layers with different level of depth: "conv3_1", "conv4_1" and "conv5_4" from the OxfordNet [24]. We also perform this experiment for dense SIFTs, though in this case random features from each image were taken as all SIFT features are normalized to have the same norm. For all types of features we use a subset of two million features as the reference set and about a thousand of features per image.

The curves averaged by all queries are shown in Figure 2. They demonstrate that the high-norm deep convolutional features from "conv5_4" layer have a small amount of "very close" neighbors, which are considerably closer than other points. This is in contrast to SIFTs, where typical distances to the closest neighbors are much closer to the distances to random descriptors in the dataset. This fact indicates that closeness of SIFT features is much less informative, and their strong similarities are unreliable and prone to accidental false positive matches. Interestingly, the individual similarities of features from "conv3_1" and "conv4_1" are less reliable than from "conv5_4" (deeper layers produce features with more reliable similarities).

Note that the second experiment is unsupervised, in the sense that we do not take correctness of matches into account when computing the distances. Rather, the second experiment highlights the substantial differences in the distribution of deep convolutional features and SIFT features in high-dimensional spaces.

The results of both experiments suggest that the individual similarities of deep features from the last convolutional layer are significantly more discriminative and the amount of false positives in matches produced by these similarities should be smaller compared to SIFTs, both because the matching is more accurate (experiment 1) and because higher-norm deep features have fewer close neighbors (experiment 2). This motivates bypassing the high-dimensional embedding step when such features need to be encoded into a global descriptor.

3.2. SPoC design

We describe the *SPoC* descriptor, which is based on the aggregation of raw deep convolutional features without embedding. We associate each deep convolutional feature f computed from image I with the spatial coordinates (x, y) corresponding to the spatial position of this feature in the map stack produced by the last convolutional layer.

Sum pooling. The construction of the SPoC descriptor starts with the sum pooling of the deep features:

$$\psi_1(I) = \sum_{y=1}^{H} \sum_{x=1}^{W} f_{(x,y)} \tag{1}$$

The scalar product of resulting descriptors corresponds to the simplest match kernel [3] between a pair of images:

$$\text{sim}(I_1, I_2) = \langle \psi(I_1), \psi(I_2) \rangle = \sum_{f_i \in I_1} \sum_{f_j \in I_2} \langle f_i, f_j \rangle \tag{2}$$

Centering prior. For most retrieval datasets, objects of interest tend to be located close to the geometrical center of an image. SPoC descriptor can be modified to incorporate such centering prior via a simple weighting heuristic. This heuristics assigns larger weights to the features from the center of the feature map stack, changing the formula (1) to:

$$\psi_2(I) = \sum_{y=1}^{H} \sum_{x=1}^{W} \alpha_{(x,y)} f_{(x,y)} \tag{3}$$

Coefficients $\alpha_{(w,h)}$ depend only on the spatial coordinates h and w. In particular, we use the Gaussian weighting scheme:

$$\alpha_{(x,y)} = \exp\left\{ -\frac{\left(y - \frac{H}{2}\right)^2 + \left(x - \frac{W}{2}\right)^2}{2\sigma^2} \right\}, \tag{4}$$

where we set σ to be one third of the distance between the center and the closest boundary (the particular choice is motivated from the "three sigma" rule of thumb from statistics, although it obviously is not directly related to our use). While very simple, this centering prior provides substantial boost in performance for some datasets as will be shown in the experiments.

Post-processing. The obtained representation $\psi(I)$ is subsequently l_2-normalized, then PCA compression and whitening are performed:

$$\psi_3(I) = \text{diag}\left(s_1, s_2, \ldots, s_N\right)^{-1} M_{\text{PCA}}\ \psi_2(I) \tag{5}$$

where M_{PCA} is the rectangular $N \times C$ PCA-matrix, N is the number of the retained dimensions, and s_i are the associated singular values.

Finally, the whitened vector is l_2-normalized:

$$\psi_{SPOC}(I) = \frac{\psi_3(I)}{\|\psi_3(I)\|_2} \tag{6}$$

Note, that the uncompressed $\psi_2(I)$ has a dimensionality C which equals to the number of output maps in the corresponding convolutional layer. Typical values for C are several hundred hence $\psi(I)$ has moderate dimensionality. Thus, when computing a compact descriptor, it takes much less data to estimate the PCA matrix and associated singular values for SPoC than for Fisher vector or triangulation embedding, since their corresponding descriptors are much higher-dimensional and the risk of overfitting is higher. The experiments below as well as the reports in e.g. [10] suggest that such overfitting can be a serious issue.

4. Experimental comparison

Datasets. We evaluate the performance of SPoC and other aggregation algorithms on four standard datasets.

INRIA Holidays dataset [8] *(Holidays)* contains 1491 vacation snapshots corresponding to 500 groups each having the same scene or object. One image from each group serves as a query. The performance is reported as mean average precision over 500 queries. Similarly to e.g. [2], we manually fix images in the wrong orientation by rotating them by ± 90 degrees.

Oxford Buildings dataset [20] *(Oxford5K)* contains 5062 photographs from Flickr associated with Oxford landmarks. 55 queries corresponding to 11 buildings/landmarks are fixed, and the ground truth relevance of the remaining dataset w.r.t. these 11 classes is provided. The performance is measured using mean average precision (mAP) over the 55 queries.

Oxford Buildings dataset+100K [20] *(Oxford105K)* contains the Oxford Building dataset and additionally 100K distractor images from Flickr.

Method	Holidays	Oxford5K (full)	Oxford105K (full)	UKB
Fisher vector, k=16	0.704	0.490	—	—
Fisher vector, k=256	0.672	0.466	—	—
Triangulation embedding, k=1	0.775	0.539	—	—
Triangulation embedding, k=16	0.732	0.486	—	—
Max pooling	0.711	0.524	0.522	3.57
Sum pooling (SPoC w/o center prior)	0.802	0.589	0.578	3.65
SPoC (with center prior)	0.784	0.657	0.642	3.66

Table 1. Detailed comparison of feature aggregation methods for deep convolutional features (followed by PCA compression to 256 dimensions and whitening/normalization). Sum pooling (SPoC) consistently outperforms other aggregation methods. Full (uncropped) query images are used for Oxford datasets. *See text for more discussions.*

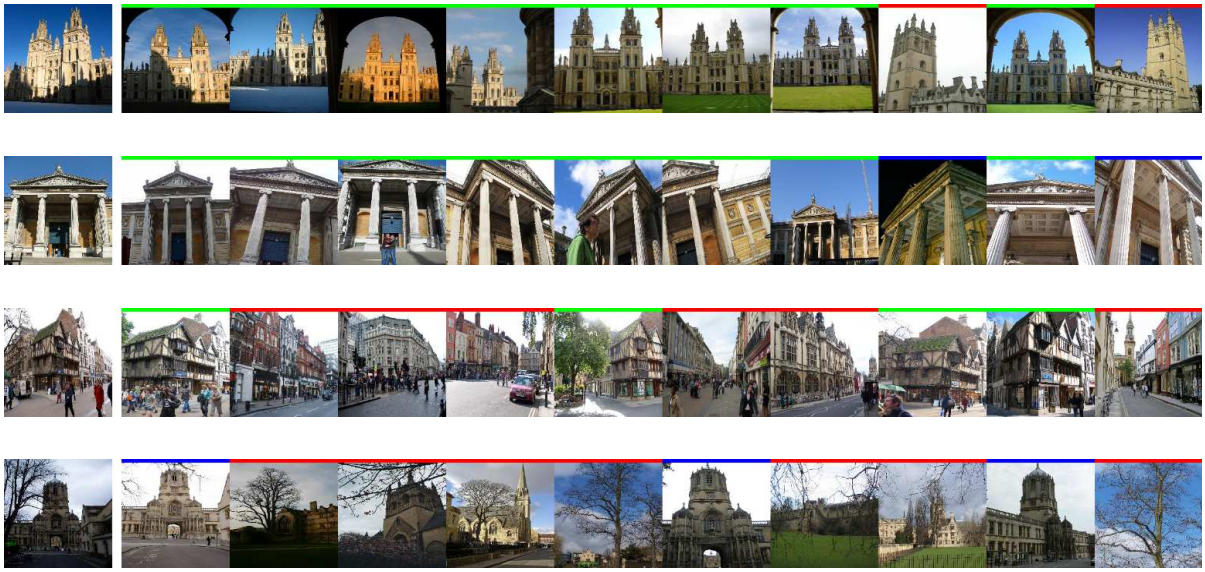

Figure 3. Retrieval examples (queries and top-ten matches) using SPoC descriptor on the Oxford Buildings dataset (Oxford5K). Red color marks false positives, green color marks true positives and blue color marks images from "junk" lists. Two top examples demonstrate that SPoC is robust to changes in viewpoint, cropping and scale. Two bottom rows are the cases where SPoC fails. In these cases SPoC "is distracted" by irrelevant objects such as the pavement or the tree.

University of Kentucky Benchmark dataset [17] *(UKB)* contains $10,200$ indoor photographs of 2550 objects (four photos per object). Each image is used to query the rest of the dataset. The performance is reported as the average number of same-object images within the top four results.

Experimental details. We extract deep convolutional features using the very deep CNN trained by Simonyan and Zisserman [24]. `Caffe` [11] package for CNNs is used. For this architecture, the number of maps in the last convolutional layer is $C = 512$. All images are resized to the size 586×586 prior to passing through the network. As a result the spatial size of the last layer is $W \times H = 37 \times 37$. The final dimensionality for SPoC and, where possible, for other methods is fixed at $N = 256$.

Aggregation methods. The emphasis of the experiments is on comparing different aggregation schemes for deep convolutional features.

We consider simple sum pooling and max pooling aggregation. In addition, we consider the two more sophisticated aggregation methods, namely Fisher Vectors [19] (Yael [6] implementation) and Triangulation embedding [10] (authors implementation). We have carefully tweaked the design choices of these methods in order to adapt them to new kind of features.

Thus, for Fisher vectors it was found beneficial to PCA-compress the features to 32 dimensions before embedding. For the triangulation embedding several tweaks that had strong impact for SIFTs had relatively small impact in the case of deep features (this includes square rooting of initial features and removing highly-energetic components). We have not used democratic kernel [10] in the systematic comparisons as it can be applied to all embedding methods, while its computationally complexity can be prohibitive in some scenarios. We observed that for Holidays it consistently improved the performance of triangulation embedding by 2 percent (measured prior to PCA).

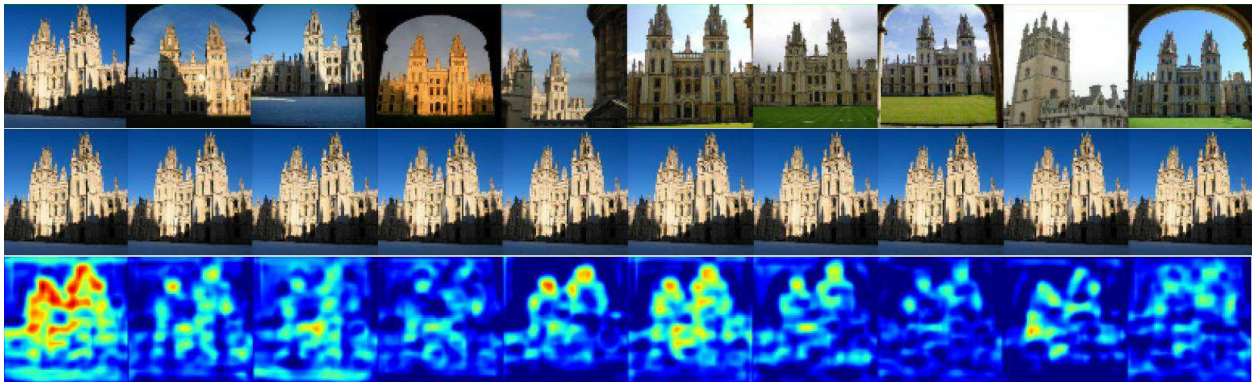

Figure 4. The examples of similarity maps between the local features of a query image and the SPoC descriptors of top-ten matches. The local features are compressed by the same PCA+whitening matrices as were used for SPoC descriptors and the cosine similarity between each local feature of a query and the SPoC descriptor of a dataset image is computed. The similarity maps allow to localize the regions of a query which are "responsible" for the fact that the particular image is considered similar to a query. For instance, for the query above, the spires of the two towers are "responsible" for most of the top matches.

All embedding methods were followed by PCA reduction to 256 dimensions. For sum pooling (SPoC) this was followed by whitening, while for Fisher vectors and Triangulation embedding we used power normalization in order to avoid overfitting (as suggested in [10]). While [1] recommends to use whitening with max pooling aggregation, we observed that it reduces retrieval performance and we do not use whitening for max pooling. In the end, all representations were $l2$-normalized and the scalar product similarity (equivalent to Euclidean distance) was used during retrieval. The parameters of PCA (and whitening) were learned on hold-out datasets (Paris buildings for Oxford Buildings, 5000 Flickr images for Holidays) unless noted otherwise.

Results. The comparison of different aggregation methods as well as different variants of SPoC are shown in Table 1 and Table 2. Several things are worth noting:

- For deep convolutional features sum pooling emerges as the best aggregation strategy by a margin. It is better than equally simple max pooling, but also better than Fisher vectors and Triangulation embedding even with handicaps discussed below, which is in sharp contrast with SIFT features.

- We demonstrate the amenability to the overfitting for different methods in Table 2. One can see that despite replacing whitening with power normalization, Fisher vectors and Triangulation embedding suffer from the overfitting of the final PCA. When learning PCA on the test dataset their performance improves very considerably. Because of this overfitting effect, it is actually beneficial to use simpler aggregation models: 16 vs 256 mixture components for Fisher vectors, 1 vs 16 cluster centers in the triangulation embedding. For SPoC and max-pooling overfitting is very small.

Method	Holidays	Oxford5K
Fisher vector, k=16	0.704	0.490
Fisher vector, PCA on test, k=16	0.747	0.540
Fisher vector, k=256	0.672	0.466
Fisher vector, PCA on test, k=256	0.761	0.581
Triang. embedding, k=1	0.775	0.539
Triang. embedding, PCA on test, k=1	0.789	0.551
Triang. embedding, k=16	0.732	0.486
Triang. embedding, PCA on test, k=16	0.785	0.576
Max pooling	0.711	0.524
Max pooling, PCA on test	0.728	0.531
SPoC w/o center prior	0.802	0.589
SPoC w/o center prior, PCA on test	0.818	0.593
SPoC (with center prior)	0.784	0.657
SPoC (with center prior), PCA on test	0.797	0.651

Table 2. Comparison of overfitting effect arose from PCA matrix learning for SPoC and other methods. Dimensionalities of all descriptors were reduced to 256 by PCA. Overfitting is much smaller for SPoC and max pooling than for the state-of-the-art high-dimensional aggregation methods.

- For triangulation embedding, degenerate configuration with one centroid performs best (more exhaustive search was performed than reported in the table). Even without PCA compression of the final descriptor to 256 dimensions, we observed that the performance of uncompressed descriptor benefitted very little from using more than one centroid, which is consistent with our observations about the statistics of deep convolutional features.

- Center prior helps for Oxford (a lot), Oxford105K (a lot) and UKB (very little) datasets and hurts (a little) for the Holidays dataset.

Method	D	Holidays	Oxford5K (full query)	Oxford5K (crop query)	Oxford105K (full query)	Oxford105K (crop query)	UKB
SIFT + Triang. + Democr. aggr.[10]	1024	0.720	–	0.560	–	0.502	3.51
SIFT + Triang. + Democr. aggr.[10]	128	0.617	–	0.433	–	0.353	3.40
Deep fully connected [2]	256	0.749	0.435	–	0.386	–	3.42
Deep fully connected + fine-tuning [2]	256	0.789	0.557	–	0.524	–	3.56
Deep convolutional + Max pooling [22]	256	0.716	0.533	–	0.489	–	–
Deep fully connected + VLAD [7]	512	0.783	–	–	–	–	–
Sum pooling (SPoC w/o center prior)	256	0.802	0.589	0.531	0.578	0.501	3.65

Table 3. Comparison with state-of-the-art for compact global descriptors. For the recent works we report results for dimensionality 256 or for the closest dimensionalities reported in those papers. Despite their simplicity, SPoC features considerably improve state-of-the-art on all four datasets.

- Whitening is much more beneficial for sum pooling than for max pooling (e.g. max pooling with whitening achieves 0.48 mAP on Oxford while 0.52 without whitening). Apparently some popular features that are both common across images and bursty and their contribution to SPoC are suppressed by whitening. For max-pooling burstiness of popular features are less of an issue.

- PCA compression benefits deep descriptors, as was observed in [2]. The uncompressed (but still whitened) SPoC features achieve mAP 0.55 on Oxford (0.59 with compression) and 0.796 on Holidays (0.802 with compression).

Some qualitative examples of good and bad retrieval examples using SPoC descriptors are shown in Figure 3. We also demonstrate some examples of similarity maps between local features of a query image and a global SPoC descriptors of dataset images. To produce these maps we compress the local features by the same PCA+whitening transformation as was used for SPoC construction. Then cosine similarities between local features of the query image and the SPoC descriptor of the dataset image are calculated and visualized as a heatmap. Such heatmaps allow to localize the regions of a query image which are similar to a particular image in the search results.

Comparison with state-of-the-art for compact global descriptors is given in Table 3. Existing works use different evaluation protocols for Oxford datasets, e.g. [10, 25] crop query images before retrieval, while recent works [22, 2, 1, 21] use uncropped query images. Here, we evaluate our SPoC descriptor in both protocols. In the crop case, for a query image we aggregate only features which have the centers of their receptive fields inside a query bounding box (as it usually done in SIFT-based approaches). As some information about context is discarded by cropping, the results with croped queries are lower.

It turns out that the gap between Oxford5K and Oxford105K performance is quite small for all evaluated settings (especially when queries are not cropped). It seems that the 100K Flickr distractor images while "distracting enough" for hand-crafted features, do not really "distract" deep convolutional features as they are too different from the Oxford Buildings images.

SPoC features provide considerable improvement over previous state-of-the-art for compact descriptors including deep descriptors in [2, 7, 22]. There are several ways how the results can be further improved. First, a mild boost can be obtained by pooling together features extracted from multiple scales of the same image (about 2 percent mAP in our preliminary experiments). Similar amount of improvement can be obtained by fine-tuning the original CNN on a specially collected dataset (in the same vein to [2]).

5. Summary and Discussion

We have investigated several alternatives for aggregating deep convolutional features into compact global descriptors, and have suggested a new descriptor (SPoC) based on simple sum-pooling aggregation. While the components of SPoC are simple and well-known, we show that the combination of our design choices results in a descriptor that provides a substantial boost over previous global image descriptors based on deep features and, in fact, over previous state-of-the-art for compact global image descriptors.

Apart from suggesting a concrete descriptor, we have evaluated advanced aggregation strategies proposed for the previous generation of local features (SIFT), and analyzed why sum pooling provides a viable alternative to them for deep convolutional features. In particular, we have highlighted the differences between local convolutional features and dense SIFT. Our experience suggests that deep convolutional features should not be treated as "new dense SIFT" in the sense that the relative performance of different computer vision techniques suggested for features like SIFT has to be reevaluated when switching to new features.

References

[1] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations

for visual recognition. *CoRR*, abs/1406.5774, 2014.

[2] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky. Neural codes for image retrieval. In *European Conference on Computer Vision - ECCV*, pages 584–599, 2014.

[3] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. In *Advances in Neural Information Processing Systems (NIPS).*, pages 135–143, 2009.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference, BMVC*, 2014.

[5] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] M. Douze and H. Jégou. The yael library. In *Proceedings of the ACM International Conference on Multimedia, MM*, pages 687–690, 2014.

[7] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *13th European Conference on Computer Vision (ECCV)*, pages 392–407, 2014.

[8] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, 2008.

[9] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3304–3311, 2010.

[10] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3310–3317, 2014.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia, MM*, pages 675–678, 2014.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[13] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 396–404, 1989.

[14] L. Liu, C. Shen, and A. van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. *CoRR*, abs/1411.7466, 2014.

[15] J. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems (NIPS)*, pages 1601–1609, 2014.

[16] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1717–1724, 2014.

[19] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3384–3391, 2010.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[21] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops*, pages 512–519, 2014.

[22] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Visual instance retrieval with deep convolutional networks. *CoRR*, abs/1412.6574, 2014.

[23] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[25] G. Tolias, Y. S. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *IEEE International Conference on Computer Vision, ICCV*, pages 1401–1408, 2013.

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