Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, Ondrej Chum

Efficient Diffusion on Region Manifolds: Recovering Small Objects with Compact CNN Representations

Ahmet Iscen[1] Giorgos Tolias[2] Yannis Avrithis[1] Teddy Furon[1] Ondřej Chum[2]
[1]Inria Rennes [2]VRG, FEE, CTU in Prague
{ahmet.iscen,ioannis.avrithis,teddy.furon}@inria.fr
{giorgos.tolias,chum}@cmp.felk.cvut.cz

Abstract

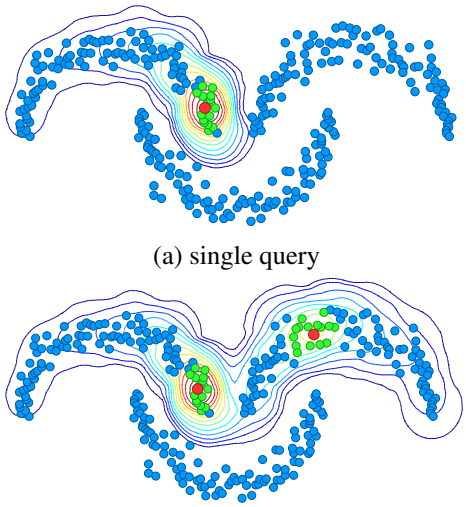
Query expansion is a popular method to improve the quality of image retrieval with both conventional and CNN representations. It has been so far limited to global image similarity. This work focuses on diffusion, a mechanism that captures the image manifold in the feature space. The diffusion is carried out on descriptors of overlapping image regions rather than on a global image descriptor like in previous approaches. An efficient off-line stage allows optional reduction in the number of stored regions. In the on-line stage, the proposed handling of unseen queries in the indexing stage removes additional computation to adjust the precomputed data. We perform diffusion through a sparse linear system solver, yielding practical query times well below one second.

Experimentally, we observe a significant boost in performance of image retrieval with compact CNN descriptors on standard benchmarks, especially when the query object covers only a small part of the image. Small objects have been a common failure case of CNN-based retrieval.

1. Introduction

Object search is a key tool behind a number of applications like content based image collection browsing [56, 34], visual localization [46, 1], and 3D reconstruction [22, 47]. Many applications benefit from retrieving images taken from various viewing angles and under different illumination, *e.g.* more information for the user while browsing, localization in day and night, and complete 3D models. Each image is represented by one or more descriptors designed or learned to exhibit a certain degree of invariance to imaging conditions. Retrieval is formulated as a nearest neighbor search in the descriptor space, performed by approximate methods [36, 25, 29, 5].

While collections of local descriptors provide good invariance, global descriptors like VLAD [26] have smaller memory footprint, but are more prone to locking onto the clutter. This mainly holds when the queried object covers only a small part of the image. In case of global CNN descriptors, the invariance is partially designed by global max [3, 52] or sum [30, 4] pooling layers or multi-scale querying [19], and partially learned by the choice of the training data. Robustness to background clutter is improved by computing descriptors over object proposals [35, 18, 57] or over a fixed grid of regions [52]. Better performance is observed at a cost of increased memory footprint [44].



(a) single query

(b) multiple queries

Figure 1. Diffusion on a synthetic dataset in $\mathbb{R}^2$. Dataset points, query points and their k-nearest neighbors are shown in blue, red, and green respectively. Contour lines correspond to ranking scores after diffusion. In this work, points are region descriptors.

In image collections, objects are depicted in various conditions. As a consequence, query and relevant images are often connected by a sequence of images, where consecutive images are similar. The descriptors of these images form a manifold in the descriptor space. Even though the images of the sequence contain the same object, the descriptors may be completely unrelated after a certain point.

This idea has been first exploited by Chum et al. [8] who introduce query expansion. The average query expansion

(AQE) is now used as a standard tool in image retrieval, due to its efficiency and significant performance boost. However, AQE only explores the neighborhood of very similar images. Recursive and scale-band recursive methods [8] further improve the results by explicitly crawling the image manifold. This is at a cost of increased query time.

Query expansion exploits the manifold of images at query time—starting from nearest neighbors of the query and using these neighbors to issue new queries. On the other hand, *diffusion* [39, 64, 13] is based on a neighborhood graph of the dataset that is constructed off-line and efficiently uses this information at query time to search on the manifold in a principled way.

We make the following contributions:

- We introduce a *regional diffusion mechanism*, which handles one or more query vectors at the same cost. There is one vector per region and a few regions per image so that constructing and storing the graph is tractable. This approach significantly improves retrieval of small objects and cluttered scenes.
- In diffusion mechanisms [39, 64, 13], query vectors are usually part of the dataset and available at the indexing stage. A novel approach to *unseen queries* with no computational overhead is proposed.
- Though a closed form solution is known to exist, it has been explicitly avoided so far [13]. We show that the commonly used alternative is in fact a well known iterative linear system solver. Since the relevant matrix is sparse and positive definite, the *conjugate gradient* method is more efficient resulting in practical query times well below one second.
- To study the dependence of performance on relative object size, we experiment on INSTRE dataset [55], which has not received much attention so far. We propose a new evaluation protocol that is in line with other well known datasets and provide a rich set of baselines to facilitate future comparisons.

Searching in parallel in more than one manifolds via diffusion and using the nearest neighbors of unseen queries are illustrated in Figure 1.

The remaining text is structured as follows. Sections 2 and 3 discuss related work and background respectively, focusing on diffusion mechanisms. Sections 4 and 5 present our contributions in detail and the experimental body.

2. Related work

This section discusses existing query expansion or reranking methods. We also review the concept of diffusion in computer vision and image retrieval in particular. Apart from AQE [8], none of these methods has been applied to retrieval in the context of convolutional features.

Query expansion. A variety of methods [8, 7, 51] employ local features and are well adapted to the Bag-of-Words model [49]. Others are generic and applicable on any global image representation [27, 42, 2, 48, 10]. In both cases, ranking is performed on the image level. Extension to regional level is not always straightforward. If even possible, such an extension would come at a significant cost, as each query region would need to be treated independently. This is unlike our regional diffusion mechanism, which has a fixed cost with respect to the number of query regions.

Diffusion. We are focusing on diffusion mechanisms, which propagate similarities through a pairwise affinity matrix [13, 39]. They are applied to many computer vision problems, such as semi-supervised classification [63], seeded image segmentation [20], saliency detection [33, 6], clustering [12] and image retrieval [28, 14, 60, 13, 58].

The power of such methods lies in capturing the intrinsic manifold structure of the data [63]. The popular PageRank algorithm [39] was originally used to estimate the importance of web pages by exploiting their links in a graph structure. Our retrieval scenario comes closer to its so called personalized [39] or query dependent versions [45], where the final ranking both respects the data manifold and the similarity to a number of query vectors.

Diffusion is used for retrieval of general scenes or shapes of particular objects [28, 14, 60, 13]. It can also fuse multiple feature modalities [61, 59] by jointly modeling them on the same graph. In these approaches, images are the nodes of the graph with edges established given a pairwise similarity measure. We differentiate by defining a graph of image *regions* linked based on region similarities while performing a single pseudo random walk for multiple query regions. Diffusion with regional similarity has been investigated before, but only to define image level affinity [62], to aggregate local features [15], or to handle bursts [16].

Donoser and Bischof [13] review a number of diffusion mechanisms for retrieval. They focus on iterative solutions arguing that closed form solutions, when existing, are impractical due to inversion of large matrices. We rather focus on a closed form solution computed approximatively with an iterative method that is particularly designed for this problem and show that this approach is faster.

3. Ranking with diffusion

Diffusion in the work of Donoser and Bischof [13] denotes a mechanism spreading the query similarities over the manifolds composing the dataset. This is only weakly related to continuous time diffusion process or random walks on graph. We mainly follow Zhou et al. [64] below.

Affinity matrix. Given a dataset $\mathcal{X} := \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$, we define the *affinity matrix* $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ having as elements the pairwise similarities between points of $\mathcal{X}$:

$$a_{ij} := s(\mathbf{x}_i, \mathbf{x}_j), \quad \forall (i,j) \in [n]^2, \tag{1}$$

where $[n] := \{1, \dots, n\}$ and $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a similarity measure assumed to be symmetric ($A = A^\top$), positive ($A > 0$), and with zero self-similarities ($\mathrm{diag}(A) = \mathbf{0}$).

Matrix A is the adjacency matrix of a weighted undirected graph G with vertices $\mathcal{X}$. The degree matrix of the graph is $D := \mathrm{diag}(A\mathbf{1}_n)$, *i.e.* a diagonal matrix with the row-wise sum of A. The Laplacian of the graph is defined as $L := D - A$. It is usual to symmetrically normalize these matrices, for instance,

$$S := D^{-1/2} A D^{-1/2}, \tag{2}$$

for the affinity matrix and $\mathcal{L} := I_n - S$ for the Laplacian, where I_n denotes the identity matrix of size n. Matrices $L, \mathcal{L}$ are positive-semidefinite [9].

Diffusion. In the work of Zhou et al. [64], a vector $\mathbf{y} = (y_i) \in \mathbb{R}^n$ specifies a set of query points in $\mathcal{X}$, with $y_i = 1$ if $\mathbf{x}_i$ is a query and $y_i = 0$ potherwise. The objective is to obtain a ranking score f_i for each point $\mathbf{x}_i \in \mathcal{X}$, represented as vector $\mathbf{f} = (f_i) \in \mathbb{R}^n$.

We focus on a particular diffusion mechanism that, given an initial vector $\mathbf{f}^0$, iterates according to

$$\mathbf{f}^t = \alpha S \mathbf{f}^{t-1} + (1 - \alpha)\mathbf{y}. \tag{3}$$

If S is a transition matrix and $\mathbf{y}$ a ℓ^1 unit vector, this defines the following 'random walk' on the graph: with probability α one jumps to an adjacent vertex according to distribution stated in S, and with $1 - \alpha$ to a query point uniformly at random. In this fashion, points spread their ranking score to their neighbors in the graph. The benefit is the ability to capture the intrinsic manifold structure represented by the affinity matrix and to combine multiple query points.

Assuming $0 < \alpha < 1$, Zhou et al. [63, 64] show that sequence $\{\mathbf{f}^t\}$ defined by (3) converges to

$$\mathbf{f}^\star = (1 - \alpha)\mathcal{L}_\alpha^{-1}\mathbf{y} \tag{4}$$

where $\mathcal{L}_\alpha := I_n - \alpha S$ is positive-definite. This follows since $\mathcal{L}_\alpha = \alpha\mathcal{L} + (1 - \alpha)I_n \succ \alpha\mathcal{L} \succeq 0$. In this work, we focus on the *closed form* solution (4) rather than its intuitive derivation from iterative process (3).

Relation to other approaches. A diffusion mechanism also appears in seeded image segmentation [20], where query points correspond to labeled pixels (seeds) and database points to the remaining unlabeled pixels. This problem is equivalent to semi-supervised classification [63]. In our context, the approach of Grady [20] decomposes $\mathbf{f} = (\mathbf{f}_d^\top, \mathbf{f}_q^\top)^\top$ for the scores of the query (fixed $\mathbf{f}_q$) and database (unknown $\mathbf{f}_d$) points. Diffusion interpolates $\mathbf{f}_d$ from $\mathbf{f}_q$ by minimizing, w.r.t. $\mathbf{f}_d$, the quadratic cost $\sum_{i,j} a_{ij}(f_i - f_j)^2 = \mathbf{f}^\top L \mathbf{f}$ to enforce that neighboring points should have similar scores. By decomposing $L = [L_d, -S_{qd}; -S_{qd}^\top, L_q]$, it is shown [20] that the solution fulfills $L_d \mathbf{f}_d = \mathbf{y}$ with $\mathbf{y} = S_{qd}^\top \mathbf{f}_q$. In our setup, L_d would be singular, preventing us to single out a solution $\mathbf{f}_d^\star$. Yet, it is easy to show that the minimizer of the cost $\alpha\mathbf{f}^\top L\mathbf{f} + (1 - \alpha)\|\mathbf{f}\|^2$ has a similar expression to (4). The regularization term singles out a solution by forcing $\mathbf{f}$ to be zero in subgraphs not connected to any query point. The details are omitted for brevity.

Local constraints. Donoser and Bischof have extensively investigated various constructions of affinity matrices in the context of image retrieval [13]. Our work uses matrix (2), which is found to be the most effective in their work, and is also used by Zhou et al. [63]. Further, to handle noise and outliers, we adopt a locally constrained random walk [31] where only pairs of points that are reciprocal (mutual) nearest neighbors are kept as edges in the graph. In particular, given $\mathbf{z} \in \mathbb{R}^d$, let

$$s_k(\mathbf{x}|\mathbf{z}) = \begin{cases} s(\mathbf{x}, \mathbf{z}), & \text{if } \mathbf{x} \in \mathrm{NN}_k(\mathbf{z}) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

be the similarity of $\mathbf{x} \in \mathcal{X}$ given $\mathbf{z}$, that is, restricted to the k nearest neighbors $\mathrm{NN}_k(\mathbf{z})$ of $\mathbf{z}$ in $\mathcal{X}$. Then,

$$s_k(\mathbf{x}, \mathbf{z}) = \min\{s_k(\mathbf{x}|\mathbf{z}), s_k(\mathbf{z}|\mathbf{x})\} \tag{6}$$

equals $s(\mathbf{x}, \mathbf{z})$ if $\mathbf{x}, \mathbf{z}$ are the k-nearest neighbors of each other in $\mathcal{X}$, and zero otherwise. We use similarity function s_k to construct affinity matrix A like in (1).

4. Method

This section describes our contributions on image retrieval: handling new query points not in the dataset, searching for multiple regions with a single diffusion mechanism, and efficiently computing the solution.

4.1. Handling new queries

In prior work on diffusion, a query point $\mathbf{q}$ is considered to be contained in the dataset $\mathcal{X}$ [63, 13]. This does not hold in a retrieval scenario, but a query can be included in the dataset graph at query time [61] as follows. The k nearest neighbors $\mathrm{NN}_k(\mathbf{q})$ of $\mathbf{q}$ in $\mathcal{X}$ are found and reciprocity is checked. The rows and columns of the affinity matrix A corresponding to $\mathrm{NN}_k(\mathbf{q})$ are updated to maintain (6) in the presence of $\mathbf{q}$, and A is augmented by appending an extra row and column for $\mathbf{q}$. Matrix S is computed by normalizing A (2). Finally, vector $\mathbf{y}$ indicates that $\mathbf{q}$ is a query. Generalizing to multiple query points is straightforward.

Even if we ignore the time needed for the above computation, we argue that locking, modifying and augmenting the entire affinity matrix for each query is not acceptable in terms of space requirements[1]. We introduce here an alternative method which defines vector $\mathbf{y}$ in a new way rather

[1] Imagine the case of multiple users querying at the same time; a different matrix per query is required. Also, updating mutual neighbors requires k-NN lists which are not available any longer.

than modifying A. Qualitatively, instead of searching for $\mathbf{q}$, we are searching for its neighbors $\mathrm{NN}_k(\mathbf{q})$, appropriately weighted. In particular, we define $\mathbf{y}$ as

$$y_i = s_k(\mathbf{x}_i|\mathbf{q}), \quad \forall i \in [n]. \tag{7}$$

Our motivation for this choice is detailed in Section 4.2 including the more general case of multiple query points. Figure 1 shows a toy 2-dimensional example of diffusion, where the k-nearest neighbors to each query point taken into account in (7) are depicted. It is evident that multiple manifolds are captured when multiple queries are issued. Section 5 experimentally shows improved performance compared to the conventional approach.

4.2. Regional diffusion

The diffusion mechanism described so far is applicable to image retrieval when database and query images are globally represented by single vectors. We call this *global diffusion* in the rest of the paper. Unlike the traditional representation with local descriptors [49, 40], global diffusion fits perfectly with the early CNN-based global features [4, 30, 43].

Global features still fail under severe occlusion or when the object of interest is small. Local CNN features from multiple image regions have been investigated for this purpose, either aggregated [17, 52] or represented as a set [44]. Given a query image, the latter means that one searches for each query feature individually.

Fortunately, diffusion as defined in section 3 can already handle multiple queries. In the following, an image is represented by a set $X_i \subset \mathbb{R}^d$ of m points, one for each region. Dataset $\mathcal{X}$ is the union of such sets over all images; n still denotes its size. The query image is also represented by a set Q of m points. Each region feature is a point possibly lying on a different manifold. We discuss below the new definition of vector $\mathbf{y}$ and the combination of individual region ranking scores into a single score per image. We call this mechanism *regional diffusion*.

Specifying queries. In the conventional approach where query points are in the dataset, one directly applies (3) with $\mathbf{y} \in \{0,1\}^{n+m}$ with m non-zero elements indicating the query points. This situation resembles the personalized PageRank [39]. However, it is simpler to keep A as an $n \times n$ affinity matrix and to set $\mathbf{y} \in \mathbb{R}^n$ as

$$y_i := \sum_{\mathbf{q} \in Q} s_k(\mathbf{x}_i|\mathbf{q}), \quad \forall i \in [n]. \tag{8}$$

Each dataset point $\mathbf{x}_i$ is assigned a scalar that is the sum of similarities over all query points $\mathbf{q}$ for which $\mathbf{x}_i$ appears in the corresponding k-nearest neighbor set $\mathrm{NN}_k(\mathbf{q})$, and zero if it appears in no such set.

Derivation. Our work is inspired by the analysis in the work of Grady [20] that we apply to the diffusion mechanism of Zhou et al. [64], where query points Q are in the dataset. We decompose the quantities in (3) as $\mathbf{f} = (\mathbf{f}_d^\top, \mathbf{f}_q^\top)^\top$, with $\mathbf{f}_d \in \mathbb{R}^n$ and $\mathbf{f}_q \in \mathbb{R}^m$,

$$S = \begin{pmatrix} S_d & B_{dq} \\ B_{qd} & S_q \end{pmatrix}, \tag{9}$$

and $\mathbf{y} = (\mathbf{0}_n^\top, \mathbf{1}_m^\top)^\top$. Subscripts d, q denote data and query respectively. Then, (3) is written as

$$\mathbf{f}_d^t = \alpha S_d \mathbf{f}_d^{t-1} + \alpha B_{dq} \mathbf{f}_q^{t-1} \tag{10}$$

$$\mathbf{f}_q^t = \alpha B_{qd} \mathbf{f}_d^{t-1} + \alpha S_q \mathbf{f}_q^{t-1} + (1-\alpha)\mathbf{1}_m. \tag{11}$$

Provided this system converges, the data part satisfies

$$\mathbf{f}_d^\star \propto \mathcal{L}_\alpha^{-1} B_{dq} \mathbf{1}_m \tag{12}$$

if $\mathbf{f}_q^\star \propto \mathbf{1}_m$, $S_q = \mathbf{0}_{m \times m}$ and $B_{qd} = \mathbf{0}_{m \times n}$. In words, the query points are perfectly retrieved, they are dissimilar to each other, and the graph is indeed directed with query regions pointing to dataset regions, but the reverse is not allowed. Comparing (12) with (4), it follows that $B_{dq}\mathbf{1}_m$ is a good choice for $\mathbf{y}$. Since B_{dq} stores the similarities between the dataset and the query points, this analysis justifies the single query (7) and the multiple queries (8) cases.

Diffusion. Given this definition of $\mathbf{y}$, diffusion is now performed on dataset $\mathcal{X}$, jointly for all query points in Q. Affinities of multiple query points are propagated in the graph in a single process at no additional cost compared to the case of a single query point. Here we are excluding the additional cost of computing $\mathbf{y}$ itself in (8) compared to (7). This search takes place in all related work. We also do not discuss how to make this search more efficient in space and time [5], which is beyond the scope of this work.

Figure 1 illustrates the diffusion on single and multiple query points. The contour lines show the ranking score any point on the plane would be assigned given the query point(s). It is evident that multiple manifolds are captured when multiple queries are issued.

Pooling. After diffusion, each image is associated with several elements of the ranking score vector $\mathbf{f}^\star$, one for each point $\mathbf{x}$ in $X \subset \mathcal{X}$. A simple way to combine these scores is to define the score of image X as

$$f(X) = \sum_{j \in [m]} w_j f^\star_{i_X(j)}, \tag{13}$$

where $i_X(j)$ is the index of the j-th point of X in the dataset $\mathcal{X}$ and $\mathbf{w} = (w_j)$ a weighting vector. The latter is defined as $\mathbf{w} = \mathbf{1}_m$ for *sum pooling* and, assuming $m < d$,

$$\mathbf{w} = (\Phi\Phi^\top + \lambda I_m)^{-1} \mathbf{1}_m \tag{14}$$

for *generalized max pooling* (GMP) [37, 23], where $\Phi = (\mathbf{x}_{i_X(1)}^\top, \ldots, \mathbf{x}_{i_X(m)}^\top)^\top$ and $\lambda \in \mathbb{R}^+$ is a regularization parameter. Our experiments show that GMP always outperforms sum pooling.

4.3. Efficient solution

Iteration (3) works well in practice but is slow at large scale. Taking the closed-form solution (4) literally, one may be tempted to compute the inverse $\mathcal{L}_\alpha^{-1}$ offline, but this matrix is not sparse like $\mathcal{L}_\alpha$. We propose a more efficient solution by making the connection to linear system solvers.

Diffusion is an iterative solver. Eq. (3) can be seen as an iteration of the *Jacobi* solver [21]. Given a linear system $A\mathbf{x} = \mathbf{b}$[2], Jacobi decomposes A as $A = \Delta + R$ where $\Delta = \mathrm{diag}(A)$. It then iterates according to

$$\mathbf{x}^t = \Delta^{-1}(\mathbf{b} - R\mathbf{x}^{t-1}). \tag{15}$$

In our case, $\mathbf{x} = \mathbf{f}$, $\mathbf{b} = (1-\alpha)\mathbf{y}$, and $A = \mathcal{L}_\alpha = I - \alpha S$. It follows that $\Delta = I_n$ and $R = -\alpha S$, so that

$$\mathbf{f}^t = \alpha S\mathbf{f}^{t-1} + (1-\alpha)\mathbf{y}. \tag{16}$$

We have just re-derived (3). Note that a sufficient condition for Jacobi's convergence is that matrix A is strictly diagonally dominant, *i.e.* $|a_{ii}| > \sum_{j\neq i} a_{ij}$ for $i \in [n]$. It is easily checked that $\mathcal{L}_\alpha$ does satisfy this condition by construction, given that $0 < \alpha < 1$. This provides an alternative proof of the main result of Zhou et al. [63].

Conjugate gradient (CG) [38] is the method of choice for solving linear systems like ours

$$\mathcal{L}_\alpha \mathbf{f} = (1-\alpha)\mathbf{y}, \tag{17}$$

where $\mathcal{L}_\alpha$ is positive-definite, and in particular for graph-related problems [54]. It has been used for random walk problems [20], but not diffusion-based retrieval according to our knowledge. In fact, the linear system formulation has been explicitly avoided in this context [13].

Here we argue, as in [32], that it is the solution of (17) that we seek, rather than the path followed by iteration (3). However, we use CG to approximate this solution, since matrix $\mathcal{L}_\alpha$ is indeed positive-definite. At each iteration, CG minimizes the quadratic function $\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{x}^\top\mathbf{b}$ in a particular direction by analytically computing the optimal step length. More importantly, the direction chosen at each iteration is conjugate to previous ones. Thus, any update of $\mathbf{x}$ along this direction does not destroy the optimality reached in the entire subspace considered thus far.

Contrary to other iterative methods including (16), CG is guaranteed to terminate in n steps. Remarkably, it provides good approximations in very few steps.

Normalization is preconditioning. Finally, a standard improvement is preconditioning, *i.e.* , solving a related system with A replaced by $C^{-1}AC^{-\top}$, a matrix satisfying a weak condition like its eigenvalues being clustered. Unfortunately, finding an appropriate matrix C can be quite complex [54]. We observe that normalization (2) *is* preconditioning.Indeed, we could equally consider matrix $L_\alpha = D - \alpha A = \alpha L + (1-\alpha)I \succ 0$ and solve the linear system

$$L_\alpha(D^{-1/2}\mathbf{f}) = (1-\alpha)(D^{1/2}\mathbf{y}) \tag{18}$$

instead, which is equivalent to (17). By normalizing L_α into $\mathcal{L}_\alpha$, we are actually performing preconditioning with $C = \mathrm{diag}(L_\alpha)^{1/2}$. This is a simple form of symmetric preconditioning, known as *diagonal scaling* or *Jacobi* [53]. It improves convergence, be it for CG or diffusion (3).

4.4. Scaling up

Despite the efficient solution described in the previous section, there are still issues concerning space and offline pre-processing at large scale. We address these issues here.

Compact representation. At large scale, the number of region features per database image should be kept as low as possible. For this reason, we learn a Gaussian Mixture Model (GMM) on the original features of each database image and represent the image by the unit normalized means. This is an even more natural choice when dealing with overlapping regions (see Section 5). As a result, it decreases the number of region features and their redundancy.

The off-line construction of the affinity matrix is quadratic in the number of vectors in the database and might not be tractable at large scale. We employ the efficient and approximate k-NN graph construction method by Dong et al. [11]. Section 5 shows that it is orders of magnitude faster than exhaustive search and has almost no effect on performance.

Truncating the affinity matrix. Instead of ranking the full dataset, diffusion re-ranks an initial search. This baseline in our experiments is done with global descriptors and kNN search. Then we apply diffusion only on the top ranked images. We truncate the affinity matrix keeping only the rows and columns related to the regions of the top ranked images and re-normalize it according to (2). The cost of this step is not significant compared to the actual diffusion.

5. Experiments

This section presents the experimental setup and investigates the accuracy of our methods for image retrieval compared with the state-of-the-art approaches.

5.1. Experimental Setup

Datasets. We use three datasets. Two are well-known image retrieval benchmarks: Oxford Buildings [40] and Paris [41]. We refer to them as Oxford5k and Paris6k. We experiment at large-scale by adding 100k distractor images from Flickr [40], forming Oxford105k and Paris106k datasets. The third corpus is the recently introduced instance search dataset called INSTRE [55]. It contains various everyday 3D or planar objects from buildings to logos

[2] We adopt the standard linear system notation in this section; matrix A is not to be confused with our affinity matrix defined in (1).

Pooling	INSTRE	Oxf5k	Oxf105k	Par6k	Par106k
sum	79.1	92.2	90.6	96.1	94.4
GMP	80.0	93.2	91.6	96.5	94.6

Table 1. Retrieval performance (mAP) of regional diffusion with sum and generalized max pooling (GMP), with $\lambda = 1$ in (14).

with many variations such as different scales, rotations, and occlusions. Some objects cover a small part of the image, making it a challenging dataset. It consists of 28,543 images from 250 different object classes. In particular, 100 classes with images retrieved from on-line sources, 100 classes with images taken by the dataset creators, and 50 classes consisting of pairs from the second category. We differentiate from the original protocol [55], which uses all database images as queries. We randomly split the dataset into 1250 queries, 5 per class, and 27293 database images, while a bounding box defines the query region[3]. The query and the database sets have no overlap. We use mean average precision (mAP) as a performance measure in all datasets.

Representation. We employ a CNN that is fine-tuned for image retrieval [43] to extract global and regional representation. In particular, this fine-tuned VGG produces 512 dimensional descriptors. We extract regions at 3 different scales as in R-MAC [52], while we additionally include the full image as a region. In this fashion, each image has on average 21 regions. The regional descriptors are aggregated and re-normalized to unit norm in order to construct the global descriptors, which is exactly as in R-MAC. We apply supervised whitening [43] to both global and regional descriptors. We use this network to perform all our initial experiments. In Section 5.4, we also report scores with higher dimensional descriptors derived from the fine-tuned ResNet101 [19] using the same fixed grid.

Implementation details. We define the affinity function using a monomial kernel [50] as $s(\mathbf{x}, \mathbf{z}) = \max(\mathbf{x}^\top \mathbf{z}, 0)^3$. The diffusion parameter α is always 0.99, as in the work of Zhou et al. [63]. The k-NN search required by (8) is assumed to access all database vectors exhaustively. Our work does not investigate how approximate search methods [36, 25, 29, 5, 24] could improve time and space consumed by this process. After computing (8), we only keep the largest k values of $\mathbf{y}$ and set the rest to zero.

5.2. Impact of different components

Neighbors. We vary the number of nearest neighbors k for constructing the affinity matrix and evaluate performance for both global and regional diffusion. The global baseline method is k-NN search with R-MAC, while the regional one is the method by Razavian et al. [44], where image regions

[3] http://people.rennes.inria.fr/Ahmet.Iscen/diffusion.html

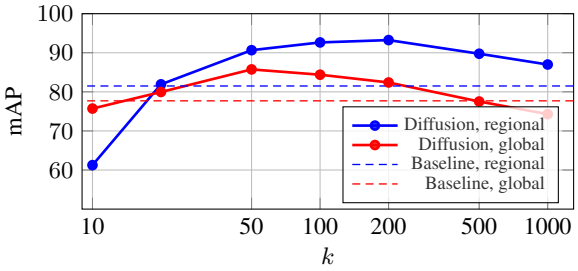


Figure 2. Impact of the number of nearest neighbors k in the affinity matrix. mAP performance for global and regional diffusion on Oxford5k; baselines are R-MAC and R-match respectively.

are indexed and cross-matched. We refer to the latter as R-match in the rest of our experiments.

Results for Oxford5k are presented in Figure 2, and are consistent in other datasets. The performance stays stable over a wide range of k. The drop for low k is due to very few neighbors being retrieved (where regional diffusion is more sensitive), whereas for high k, it is due to capturing more than the local manifold structure (where regional diffusion is superior). This behavior is consistent with the fact that small patterns appear more frequently than entire images.

We set $k = 200$ for regional diffusion, and $k = 50$ for global diffusion for the rest of our paper. Since only mutual neighbors are linked, the actual number of edges per element is less: The average number of edges per image (resp. region) is 25 (resp. 75) for global (resp. regional) diffusion, measured on INSTRE. We set $k = 200$ for the query as well in the case of the regional diffusion, while for the global one $k = 10$ is needed to achieve good performance.

Pooling. We evaluate the two pooling strategies after regional diffusion in Table 1. Generalized max pooling has a small but consistent benefit in all datasets. We use this strategy for the rest of our experiments. Weights (14) are computed off-line and only one scalar per region is stored.

Efficient diffusion with conjugate gradient. We compare the iterative diffusion (3) to our conjugate gradient solution. We iterate each method until convergence. Performance is presented in Figure 3 with timings measured on a machine with a 4-core Intel Xeon 2.00GHz CPU. CG converges in as few as 20 iterations, which are also faster, while (3) reaches the same performance as CG only after 110 iterations.

The average query time on Oxford5k including all stages for global baseline, regional baseline, global diffusion and regional diffusion without truncation is 0.001s, 0.321s, 0.02s, and 0.664s, respectively.

Handling new queries. We compare our new way of handling new queries to the conventional approach that assumes queries to be part of the dataset. Our method achieves 80.0 mAP on INSTRE compared to 77.7 achieved by the conventional approach. We therefore not only offer space

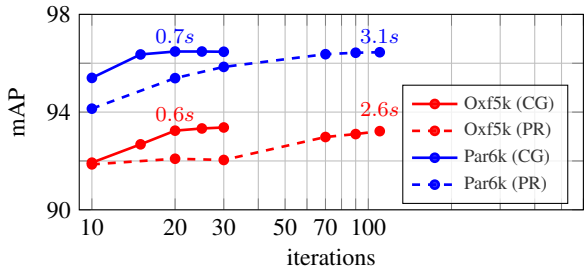


Figure 3. mAP performance of regional diffusion *vs.* number of iterations for conjugate gradient (CG) and iterative diffusion (3). Labels denote diffusion time.

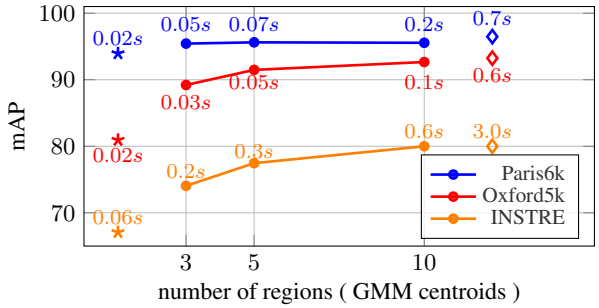


Figure 4. mAP performance for varying number of regional descriptors after learning a GMM per image. Symbol ⋆ denotes global diffusion, and ⋄ to the default number of regions (21) per image. Average diffusion time in seconds is shown in text labels.

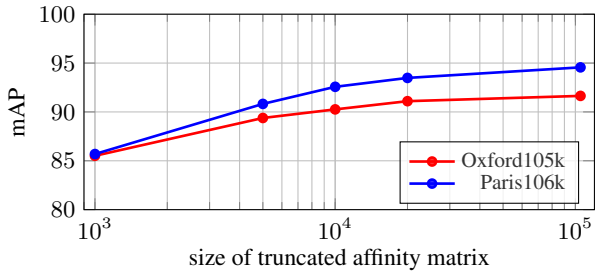


Figure 5. Retrieval performance (mAP) versus the shortlist size used for affinity matrix truncation.

improvements but also better performance,mainly in the case of regional diffusion. The main difference is that k nonzero elements are kept both per query region (8) and for the entire vector $\mathbf{y}$. This, due to the overlapping nature of the CNN regions, may filter out incorrect neighbors.

5.3. Large scale diffusion

We now focus on the large scale solutions of Section 4.4.

Reduced number of regions. Figure 4 shows the impact of reducing the number of regions with Gaussian mixture models. Having as few as 5 descriptors per image already achieves competitive performance, while reducing the online search complexity. We decrease the number of neighbors k to 50 when GMM reduction is used, as there are now less positive neighbors.

Method	$m \times d$	INSTRE	Oxf5k	Oxf105k	Par6k	Par106k
Global descriptors - nearest neighbor search						
CroW [30]†	512	-	68.2	63.2	79.8	71.0
R-MAC [43]	512	47.7	77.7	70.1	84.1	76.8
R-MAC [19]	2,048	62.6	83.9	80.8	93.8	89.9
NetVLAD [1]†	4,096	-	71.6	-	79.7	-
Global descriptors - query expansion						
R-MAC [43]+AQE [8]	512	57.3	85.4	79.7	88.4	83.5
R-MAC [43]+SCSM [48]	512	60.1	85.3	80.5	89.4	84.5
R-MAC [43]+HN [42]	512	64.7	79.9	-	92.0	-
Global diffusion	512	70.3	85.7	82.7	94.1	92.5
R-MAC [19]+AQE [8]	2,048	70.5	89.6	88.3	95.3	92.7
R-MAC [19]+SCSM [48]	2,048	71.4	89.1	87.3	95.4	92.5
Global diffusion	2,048	80.5	87.1	87.4	96.5	95.4
Regional descriptors - nearest neighbor search						
R-match [44]	21×512	55.5	81.5	76.5	86.1	79.9
R-match [44]	21×2,048	71.0	88.1	85.7	94.9	91.3
Regional descriptors - query expansion						
HQE [51]	2.4k×128	74.7	89.4†	84.0†	82.8†	-
R-match [44]+AQE [8]	21×512	60.4	83.6	78.6	87.0	81.0
Regional diffusion⋆	5×512	77.5	91.5	84.7	95.6	93.0
Regional diffusion⋆	21×512	80.0	93.2	90.3	96.5	92.6
R-match [44]+AQE [8]	21×2,048	77.1	91.0	89.6	95.5	92.5
Regional diffusion⋆	5×2,048	88.4	95.0	90.0	96.4	**95.8**
Regional diffusion⋆	21×2,048	**89.6**	**95.8**	**94.2**	**96.9**	95.3

Table 2. Performance comparison to the state of the art. Results from original publications are marked with †, otherwise they are based on our implementation. Our methods are marked with ⋆. Points at 512D are extracted with VGG [43] and at 2048D with ResNet101 [19]. Regional diffusion with 5 regions uses GMM.

Affinity matrix with Dong's algorithm [11]. We compare the exhaustive construction of matrix A to Dong's efficient k-NN graph algorithm [11]. Exhaustive search for Oxford105k composed of 2.2M regions takes 96 hours on a machine with a 12-core Intel Xeon 2.30GHz CPU. The approximate graph only takes 45 minutes and affects the final retrieval performance only slightly. It achieves 91.6 mAP on Oxford105k and 94.6 on Paris106k, while the exhaustive construction yields 92.5 and 95.2 respectively.

Truncation is a means to handle large scale datasets, *i.e.* more than 100k images. Regional diffusion on the full dataset takes 13.9s for Oxford105k, which is not practical. We therefore rank images according to the aggregated regional descriptors, which is equivalent to the R-MAC representation [52], and then perform diffusion on a short-list.

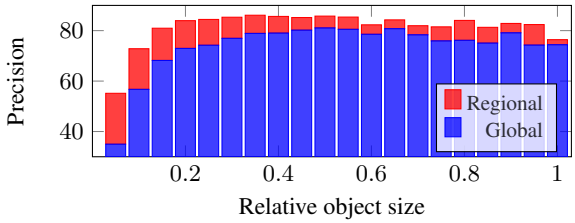


Figure 6. Precision of each positive image measured at the position where it was retrieved, averaged over positive images according relative object size. Statistics computed on INSTRE over all queries for global and regional diffusion.

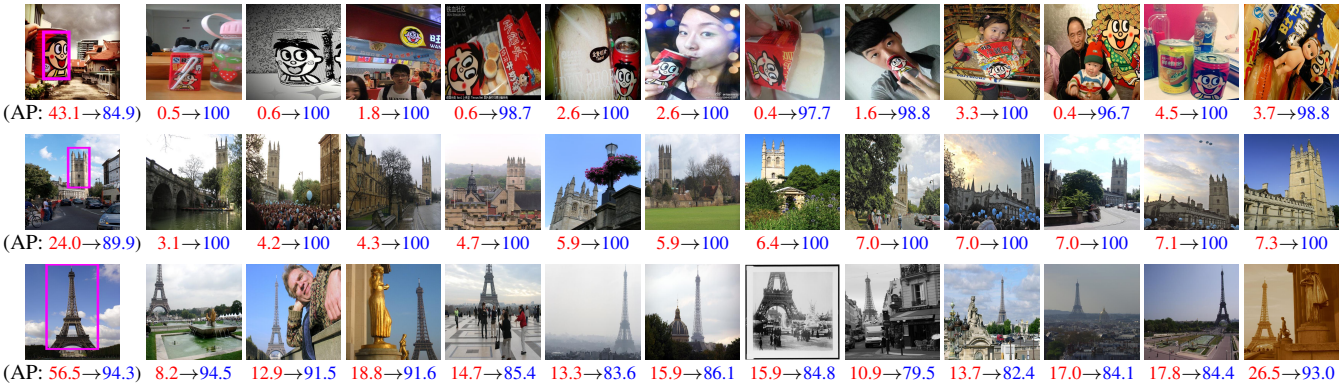


Figure 7. Query examples from INSTRE, Oxford, and Paris datasets and retrieved images ranked by decreasing order of ranking difference between global and regional diffusion. We measure precision at the position where each image is retrieved and report this under each image for global and regional diffusion. Average Precision (AP) is reported per query for the two methods.

Figure 5 reports results with truncation. The performance of the full database diffusion is nearly attained by re-ranking less than 10% of the database. The entire truncation and diffusion process on Oxford105k takes 1s, with truncation and re-normalization taking only a small part of it. In the following, search on Oxford105k and Paris105k is performed by truncating the top 10k images. This choice results in an affinity matrix A of around 200k regions. When GMM reduction is used, our short-list size is chosen so that A has 2M regions too, keeping re-ranking complexity fixed.

Our approach is scalable thanks to truncation: the short-list length is fixed and so is the re-ranking time, regardless of the database size and the dimensionality of the descriptors. Although this shortlist contains a small fraction of the database, its significantly outperforms the baseline.

Small objects. We present quantitative and qualitative results revealing that images benefit from our method mainly when the depicted object is small and the scene is cluttered. Figure 7 shows that the retrieved images with the highest increase of precision of regional compared to global diffusion contain small objects that the latter cannot see. Since the bounding boxes are available for all images of INSTRE, we quantitatively measure precision for all positive images: Figure 6 shows that the highest improvement indeed comes for objects with small relative size.

5.4. Comparison to other methods

We compare with the state-of-the-art approaches with global or regional representation, with or without query expansion. Table 2 summarizes the results. We implement three methods typically combined with BoW, namely Average Query Expansion (AQE) [8], Spatially Constrained Similarity Measure (SCSM) [48] and Hello Neighbor (HN) [42]. AQE is also effective with CNN global representation [52, 30, 18]. A baseline for the regional scenario is R-match [44]. We additionally extend AQE to regional representation[4] combined with the similarity used in R-match. Hamming Query Expansion[5] (HQE) [51] is the only method not using CNNs, but local descriptors.

Regional diffusion significantly outperforms all other methods in all datasets. Global diffusion performs well on Paris because query objects almost fully cover the image in most of the database entries. This does not hold on INSTRE, which contains a lot of small objects. The improvements of regional diffusion are in this case much larger.

6. Conclusion

We propose a retrieval approach capturing distinct manifolds in the description space at no additional cost compared to a single query. We experimentally show that it significantly improves retrieval of small objects and cluttered scenes. The conclusion is that as few as 5-10 regional CNN descriptors can convey important information on small objects while thousands of conventional local descriptors are typically needed. Thus, a regional affinity matrix becomes possible. Regional diffusion was not possible before. In contrast to prior work, we use the closed form solution of the diffusion iteration, obtained by the conjugate gradient method. Combined with our contributions on space efficiency, this achieves large scale search at reasonable query times. Using recent CNN architectures, we achieve state-of-the-art and near optimal performance on two popular benchmarks and a recent more challenging dataset.

Acknowledgments The authors were supported by the MSMT LL1303 ERC-CZ grant. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

[4] AQE has not been proposed in a regional scenario. We extend it as competitive baseline derived from prior work.

[5] We evaluated HQE on INSTRE for the purposes of this work.

References

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 1, 7

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, June 2012. 2

[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPRW*, 2014. 1

[4] A. Babenko and V. Lempitsky. Aggregating deep convolutional features for image retrieval. In *ICCV*, 2015. 1, 4

[5] A. Babenko and V. Lempitsky. Efficient indexing of billion-scale datasets of deep descriptors. In *CVPR*, 2016. 1, 4, 6

[6] S. Chen, L. Zheng, X. Hu, and P. Zhou. Discriminative saliency propagation with sink points. *Pattern recognition*, 60:2–12, 2016. 2

[7] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, June 2011. 2

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, October 2007. 1, 2, 7, 8

[9] F. R. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997. 3

[10] A. Delvinioti, H. Jégou, L. Amsaleg, and M. Houle. Image retrieval with reciprocal and shared nearest neighbors. In *VISAPP*, 2014. 2

[11] W. Dong, M. Charikar, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW*, March 2011. 5, 7

[12] M. Donoser. Replicator graph clustering. In *BMVC*, 2013. 2

[13] M. Donoser and H. Bischof. Diffusion processes for retrieval revisited. In *CVPR*, 2013. 2, 3, 5

[14] A. Egozi, Y. Keller, and H. Guterman. Improving shape retrieval by spectral matching and meta similarity. IEEE *Transactions on Image Processing*, 19(5):1319–1327, 2010. 2

[15] T. Furuya and R. Ohbuchi. Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. In *ICMR*, 2015. 2

[16] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian. Democratic diffusion aggregation for image retrieval. *IEEE Trans. on Multimedia*, 18:1661 – 1674, 2016. 2

[17] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 4

[18] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. *ECCV*, 2016. 1, 8

[19] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. In *arXiv*, 2016. 1, 6, 7

[20] L. Grady. Random walks for image segmentation. *IEEE Trans. PAMI*, 28(11):1768–1783, 2006. 2, 3, 4, 5

[21] W. Hackbusch. *Iterative solution of large sparse systems of equations*. Springer Verlag, 1994. 5

[22] J. Heinly, J. L. Schonberger, E. Dunn, and J.-M. Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *CVPR*, 2015. 1

[23] A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. In *arXiv*, 2014. 4

[24] A. Iscen, M. Rabbat, and T. Furon. Efficient large-scale similarity search using matrix factorization. In *CVPR*, 2016. 6

[25] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. PAMI*, 33(1):117–128, January 2011. 1, 6

[26] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, June 2010. 1

[27] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007. 2

[28] Y. Jing and S. Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Trans. PAMI*, 30(11):1877–1890, 2008. 2

[29] Y. Kalantidis and Y. Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *CVPR*, 2014. 1, 6

[30] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*, 2016. 1, 4, 7, 8

[31] P. Kontschieder, M. Donoser, and H. Bischof. Beyond pairwise shape similarity analysis. In *ACCV*, 2009. 3

[32] A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004. 5

[33] S. Lu, V. Mahadevan, and N. Vasconcelos. Learning optimal seeds for diffusion-based salient object detection. In *CVPR*, 2014. 2

[34] A. Mikulik, O. Chum, and J. Matas. Image retrieval for online browsing in large image collections. In *International Conference on Similarity Search and Applications*, 2013. 1

[35] K. R. Mopuri and R. V. Babu. Object level deep feature pooling for compact image representation. *CVPRW*, 2015. 1

[36] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. PAMI*, 36, 2014. 1, 6

[37] N. Murray and F. Perronnin. Generalized max-pooling. In *CVPR*, June 2014. 4

[38] J. Nocedal and S. Wright. *Numerical optimization*. Springer, 2006. 5

[39] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999. 2, 4

[40] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, June 2007. 4, 5

[41] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, June 2008. 5

[42] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011. 2, 7, 8

[43] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. *ECCV*, 2016. 4, 6, 7

[44] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4:251–258, 2016. 1, 4, 6, 7, 8

[45] M. Richardson and P. M. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *NIPS*, 2001. 2

[46] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt. Image retrieval for image-based localization revisited. In *BMVC*, 2012. 1

[47] J. L. Schonberger, F. Radenovic, O. Chum, and J.-M. Frahm. From single image query to detailed 3d reconstruction. In *CVPR*, 2015. 1

[48] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Spatially-constrained similarity measure for large-scale object retrieval. *IEEE Trans. PAMI*, 36(6):1229–1241, 2014. 2, 7, 8

[49] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2, 4

[50] G. Tolias, Y. Avrithis, and H. Jégou. To aggregate or not to aggregate: Selective match kernels for image search. In *ICCV*, December 2013. 6

[51] G. Tolias and H. Jégou. Visual query expansion with or without geometry: refining local descriptors by feature aggregation. *Pattern recognition*, 47(10):3466–3476, 2014. 2, 7, 8

[52] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *ICLR*, 2016. 1, 4, 6, 8

[53] L. N. Trefethen and D. Bau III. *Numerical linear algebra*. SIAM, 1997. 5

[54] N. K. Vishnoi. Laplacian solvers and their algorithmic applications. *Theoretical Computer Science*, 8(1-2):1–141, 2012. 5

[55] S. Wang and S. Jiang. INSTRE: a new benchmark for instance-level object retrieval and recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11:37, 2015. 2, 5, 6

[56] T. Weyand and B. Leibe. Discovering favorite views of popular places with iconoid shift. In *ICCV*, 2011. 1

[57] L. Xie, R. Hong, B. Zhang, and Q. Tian. Image classification and retrieval are one. In *ICMR*, 2015. 1

[58] L. Xie, Q. Tian, W. Zhou, and B. Zhang. Fast and accurate near-duplicate image search with affinity propagation on the imageweb. *CVIU*, 124, 2014. 2

[59] F. Yang, B. Matei, and L. S. Davis. Re-ranking by multi-feature fusion with diffusion for image retrieval. In *WACV*, 2015. 2

[60] X. Yang, S. Koknar-Tezel, and L. J. Latecki. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, 2009. 2

[61] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, 2012. 2, 3

[62] W. Zhang, C.-W. Ngo, and X. Cao. Hyperlink-aware object retrieval. IEEE *Transactions on Image Processing*, 25(9):4186–4198, 2016. 2

[63] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *NIPS*, 2003. 2, 3, 5, 6

[64] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *NIPS*, 2003. 2, 3, 4