Locality in Generic Instance Search from One Example

Ran Tao[1], Efstratios Gavves[1], Cees G.M. Snoek[1], Arnold W.M. Smeulders[1,2]
[1]ISLA, Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
[2]Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Abstract

This paper aims for generic instance search from a single example. Where the state-of-the-art relies on global image representation for the search, we proceed by including locality at all steps of the method. As the first novelty, we consider many boxes per database image as candidate targets to search locally in the picture using an efficient point-indexed representation. The same representation allows, as the second novelty, the application of very large vocabularies in the powerful Fisher vector and VLAD to search locally in the feature space. As the third novelty we propose an exponential similarity function to further emphasize locality in the feature space. Locality is advantageous in instance search as it will rest on the matching unique details. We demonstrate a substantial increase in generic instance search performance from one example on three standard datasets with buildings, logos, and scenes from 0.443 to 0.620 in mAP.

1. Introduction

In instance search the ideal is to retrieve all pictures of an object given a set of query images of that object [2, 11, 21, 28]. Similar to [3, 6, 26, 30, 33], we focus on instance search on the basis of only one example. Different from the references, we focus on *generic* instance search, like [4, 13, 24], in that the method will not be optimized for buildings, logos or another specific class of objects.

The challenge in instance search is to be invariant to appearance variations of the instance while ignoring other instances from the same type of object. With only one example, generic instance search will profit from finding relevant unique details, more than in object categorization, which searches for identifying features shared in the class of objects. The chances of finding relevant unique details will increase when their representation is invariant and the search space is reduced to local and promising areas. From this observation, we investigate ways to improve locality in instance search at two different levels: locality in the picture *and* locality in the feature space.

In the picture, we concentrate the search for relevant unique details to reasonable candidate localizations of the object. Spatial locality has been successfully applied in image categorization [10, 34]. It is likely to be even more successful in instance search considering that there is only one training example and the distinctions to the members of the negative class are smaller. The big challenge here is to keep the number of candidate boxes low while retaining the chance of having the appropriate box. The successful selective search [35] is still evaluating thousands of candidate boxes. Straightforward local picture search requires a demanding 1,000s-fold increase in memory to store the box features. We propose efficient storage and evaluation of boxes in generic instance search. We consider this as the most important contribution of this work.

In the feature space, local concentration of the search is achieved in two ways. The first tactic is using large visual vocabularies as they divide the feature space in small patches. In instance search, large vocabularies have been successfully applied in combination with Bag of Words (BoW), particularly to building search [19, 26, 27]. Without further optimizations to buildings [7, 26], BoW was shown inferior in performance in instance search to VLAD and Fisher vector [13]. Therefore, we focus on the latter two for generic instance search. Yet the use of large vocabularies with these methods is prohibited by the memory it requires. We propose the use of large vocabularies with these modern methods.

As a second tactic in the feature space, we propose a new similarity function, named *exponential similarity*, measuring the relevance of two local descriptors. The exponential similarity enhances locality in the feature space in that the remote correspondences are punished much more than the closer ones. Hence this similarity function emphasizes local search in the feature space.

As the first novelty in this paper, we aim for an efficient evaluation of many boxes holding candidates for the target by a point-indexed representation independent of their number. The representation allows, as the second novelty, the application of very large vocabularies in Fisher vector and VLAD in such a way that the memory use is independent

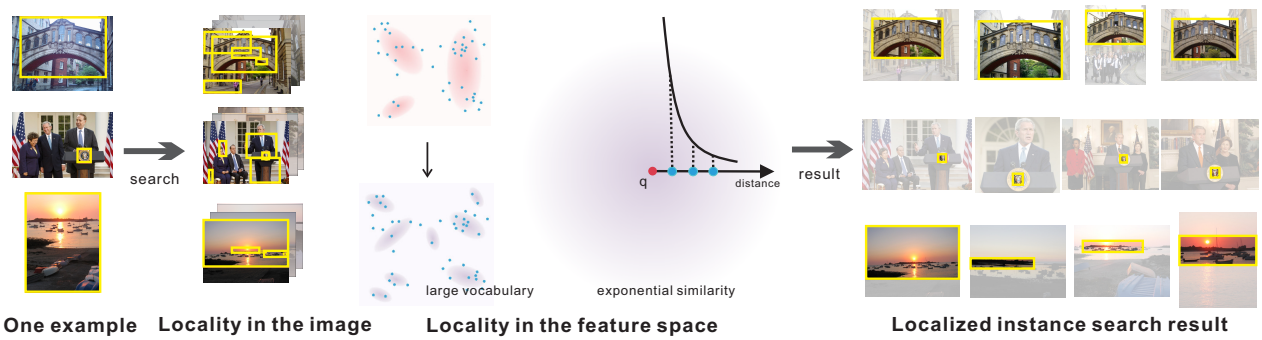

Figure 1. We propose locality in generic instance search from one example. As the first novelty, we consider many boxes as candidate targets to search locally in the picture by an efficient point-indexed representation. The same representation allows, as the second novelty, the application of very large vocabularies in Fisher vector and VLAD to search locally in the feature space. As the third novelty, we propose the exponential similarity to emphasize local matches in feature space. The method does not only improve the accuracy but also delivers a reliable localization.

of the vocabulary size. The large vocabulary enables the distinction of local details in the feature space. Thirdly, we propose the exponential similarity function which emphasizes local matches in the feature space. We summarize our novelties in Figure 1. We demonstrate a drastic increase in performance in generic instance search, enabled by an emphasis on locality in the feature space and the image.

2. Related work

Most of the literature on instance search, also known as object retrieval, focuses on a particular type of object. In [3, 26, 27] the search is focused on buildings, for which vocabularies of 1M visual words successfully identify tiny details of individual buildings. For the same purpose, building search, geometrical verification in [26], improves the precision further, and query expansion in [6, 7] with geometrically verified examples further improves recall. For the topic of logos specifically, in [30], a method is introduced by utilizing the correlation between incorrect keypoint matches to suppress false retrievals. We cover these hard problems on buildings and logos, but at the same time consider the retrieval of arbitrary scenes. To that end, we consider the three standard datasets, Oxford5k [26], BelgaLogos [15] and the Holidays dataset [11] holding 5,062, 10,000 and 1,491 samples each. We do the analysis to evaluate one and the same generic method. Besides, we define a new dataset, TRECVID50k, which is a 50,000 sample of the diverse TRECVID dataset [21] for generic instance search.

BoW quantizes local descriptors to closest words in a visual vocabulary and produces a histogram counting the occurrences of each visual word. VLAD [13] and Fisher vector [23] improve over the performance of BoW by difference encoding, subtracting the mean of the word or a Gaussian fit to all observations respectively. As VLAD and Fisher vector focus on differences in the feature space, their performance is expected to be better in instance search, especially when the dataset grows big. We take the recent application to instance search of VLAD [4, 13] and Fisher vector [13, 24] as our point of reference.

In [19, 20, 26], the feature space is quantized with a large BoW-vocabulary leading to a dramatic improvement in retrieval quality. In VLAD and Fisher vector, storing the local descriptors in a single feature vector has the advantage that the similarity between two examples can readily be compared with standard distance measures. However, such a one-vector-representation stands against the use of large vocabularies in these methods, as the feature dimensionality, and hence the memory footprint, grows linearly with the vocabulary size. Using a vocabulary with 20k visual clusters will produce a vector with 2.56M dimensions for VLAD [4]. In this study, we present a novel representation independent of the vocabulary size in memory usage, effectively enabling large vocabularies.

Spatial locality in the picture has shown a positive performance effect in image categorization [10, 34]. Recent work [1, 9, 35] focuses on generating candidate object locations under a low miss rate. Selective search [35] oversegments the image and hierarchically groups the segments with multiple complementary grouping criteria to generate object hypotheses, achieving a high recall with a reasonable number of boxes. We adopt selective search for instance search, but the method we propose will function for any other location selection method.

Spatial locality has been applied in retrieval [16, 17, 14]. [16] applies BoW on very, very many boxes inserted in a branch and bound algorithm to reduce the number of visits. We reduce their number from the start [35], and we adopt the superior VLAD and Fisher vector representations rather than BoW. [14] randomly splits the image into cells and applies BoW model. [17] proposes a greedy search method for a near-optimal box and uses the score of the box to re-rank the initial list generated based on global BoW

histograms. The reference applies locality after the analysis, relying on the quality of the initial result. The method in the reference is specifically designed for BoW, while we present a generic approach which is applicable to VLAD, Fisher vector and BoW as well. The authors in [4] study the benefits of tiling an image with VLADs when searching for buildings which cover a small portion of an image. In the reference, an image is regularly split into a 3 by 3 grid, and 14 boxes are generated, 9 small ones, 4 medium ones (2 x 2 tiles), and the one covering the entire image. A VLAD descriptor is extracted from each of the boxes and evaluated individually. In this paper, we investigate the effect of spatial locality using the candidate boxes created by the state-of-the-art approach in object localization rather than tiling, and evaluate on a much broader set of visual instances.

The exponential similarity function introduced in this work is similar to the thresholded polynomial similarity function recently proposed in [32] and the query adaptive similarity in [29] in that all pose higher weights on closer matches which are more likely to be true correspondences. However, our proposal has fewer parameters than [32] and does not need the extra learning step of [29].

3. Locality in the image

Given the query instance outlined by a bounding box, relevant details in a positive database image usually occupy only a small portion of the image. Analyzing the entire database image in the search is suboptimal as the real signal on the relevant region will drown in the noise from the rest. The chance of returning an image which contains the target instance is expected to be higher if the analysis is concentrated on the relevant part of the image only. To this end, we propose to search locally in the database image by evaluating many bounding boxes holding candidates for the target and ranking the images based on the per-image maximum scored box. Generating promising object locations has been intensively researched in the field of category-level object detection [1, 9, 35]. We adopt selective search [35] to sample the bounding boxes.

Evaluating many bounding boxes per database image, however, is practically infeasible in combination with VLAD or Fisher vector, since the VLAD or Fisher representations for all the boxes are either too expensive to store or too slow to compute on-the-fly. On the 5,062 images of the Oxford5k dataset [26], selective search will generate over 6 million boxes. With VLAD encoding this will generate over 700 gigabytes even with a small vocabulary consisting of 256 clusters. We therefore propose to decompose the one-vector representations into point-indexed representations, which removes the linear dependence of the memory requirement on the number of sampled boxes. Furthermore, we decompose the similarity function accordingly for efficient evaluation, saving on an expensive online re-composition of the one-vector representation. In the following we first briefly review VLAD and Fisher vector, and then describe the decomposition of the appearance models and the similarity measure, which allows to evaluate boxes efficiently in a memory compact manner.

3.1. Global appearance models

Let $\mathcal{P} = \{\mathbf{p}_t, t = 1...T\}$ be the set of interest points and $\mathcal{X} = \{\mathbf{x}_t, t = 1...T\}$ be the d-dimensional local descriptors quantized by a visual vocabulary $\mathcal{C} = \{\mathbf{c}_i, i = 1...k\}$ to its closest visual word $q(\mathbf{x}) = argmin_{\mathbf{c}\in\mathcal{C}}\|\mathbf{x} - \mathbf{c}\|^2$, where $\|.\|$ is the ℓ_2 norm.

Where BoW counts the occurrences of each visual word into a histogram $\mathbf{V}_B = [w_1, ...w_k]$ with $w_i = \sum_{\mathbf{x}_t\in\mathcal{X}:q(\mathbf{x}_t)=\mathbf{c}_i} 1$, VLAD sums the difference between the local descriptor and the visual word center, which results in a d-dimensional sub-vector per word $\mathbf{v}_i = \sum_{\mathbf{x}_t\in\mathcal{X}:q(\mathbf{x}_t)=\mathbf{c}_i}(\mathbf{x}_t - \mathbf{c}_i)$, concatenated into:$\mathbf{V}_V = [\mathbf{v}_1, ..., \mathbf{v}_k]$. VLAD quantifies differentiation within the visual words and provides a joint evaluation of several local descriptors.

Fisher vector models the local descriptor space by a Gaussian Mixture Model, with parameters $\lambda = \{\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, i = 1, ..., k\}$ where $\omega_i, \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i$ are the mixture weight, mean vector and the standard deviation vector of the i^{th} component. Fisher vector describes how a set of local descriptors deviates from the universal distribution of the local descriptor space via taking the gradient of the set's log likelihood with respect to the parameters of the GMM, first applied to image classification by Perronnin *et al*. [23, 25]. Later the gradient with respect to the mean was applied to retrieval [24, 13]: $\mathbf{g}_i = \frac{1}{\sqrt{\omega_i}}\sum_{t=1}^{T}\gamma_t(i)\frac{\mathbf{x}_t - \boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i}$ where $\gamma_t(i)$ is the assignment weight of $\mathbf{x}_t$ to Gaussian i. We drop T from the denominator as mentioned in [13], as it will be canceled out during normalization. The Fisher vector representation $\mathbf{V}_F$ is the concatenation of $\mathbf{g}_i$ for $i = 1...k$: $\mathbf{V}_F = [\mathbf{g}_1, ..., \mathbf{g}_k]$.

3.2. Decomposition of appearance models

Decomposing a VLAD vector into point-indexed features is straightforward. The description of an interest point $\mathbf{p}_t$ with local descriptor $\mathbf{x}_t$ in VLAD is simply represented by the index of the closest visual word plus the difference vector with the word center

$$\{q_{ind}(\mathbf{x}_t); \mathbf{d}_t = \mathbf{x}_t - q(\mathbf{x}_t)\}. \quad (1)$$

Before we can decompose Fisher vectors, we note that in the original implementation each local descriptor contributes to all k Gaussian components, which imposes a serious memory burden as each point will produce k different representations. We thereby modify the original formulation by allowing association with the largest assignment weights only. A similar idea has been explored for

object detection in [5], where only the components with assignment weights larger than a certain threshold are considered. After rewriting the above equation for $\mathbf{g}_i$ into $\mathbf{g}_i = \sum_{\mathbf{x}_t \in \mathcal{X}: \gamma_t(i) \neq 0} \frac{\gamma_t(i)}{\sqrt{\omega_i}} \frac{\mathbf{x}_t - \boldsymbol{\mu}_i}{\boldsymbol{\sigma}_i}$, the description of a point in the truncated Fisher vector, tFV, is given by the index r_t^j of the Gaussian component with j^{th} largest soft assignment weight, the assignment weight divided by the square root of the mixture weight and similar to the VLAD-case, the difference to the mean. Point $\mathbf{p}_t$ is represented by

$$\{[r_t^j; \frac{\gamma_t(r_t^j)}{\sqrt{\omega_{r_t^j}}}; \mathbf{d}_{tj} = \frac{\mathbf{x}_t - \boldsymbol{\mu}_{r_t^j}}{\boldsymbol{\sigma}_{r_t^j}}], j = 1...m\}. \quad (2)$$

Apparently, the memory consumption of the point-indexed representations is independent of the number of boxes. However, as in VLAD and tFV the difference vectors have the same high dimensionality as the local descriptors, the memory usage of the representations is as yet too large. Hence, we propose to quantize the continuous space of the difference vectors into a discrete set of prototypic elements and store the index of the closest prototype instead of the exact difference vector to arrive at an arbitrarily close approximation of the original representation in much less memory. As in [12], the difference vectors are split into pieces with equal length and each piece is quantized separately. We randomly sample a fixed set of prototypes from real data and use the same set to encode all pieces. Denote the quantization function by $\widetilde{q}$ and the index of the assigned prototype by $\widetilde{q_{ind}}$. Each difference vector $\mathbf{d}_t$ is represented by $[\widetilde{q_{ind}}(\mathbf{d}_{t_s}), s = 1...l]$, where $\mathbf{d}_{t_s}$ is the s^{th} piece of $\mathbf{d}_t$. The quantized point-indexed representations are memory compact, and box independent. To allow the evaluation of bounding boxes, we also store the meta information of the boxes, such as the coordinates, which costs a small extra amount of space.

3.3. Decomposition of similarity measure

Cosine similarity is the de facto similarity measure for VLAD [4, 13] and Fisher vector [13, 24], and hence for tFV. We propose to decompose accordingly the similarity measure into pointwise similarities, otherwise the one-vector-representation of a box has to be re-composed before being able to measure the similarity score of the box.

To explain, first consider the decomposition of the cosine similarity for BoW histograms. Let Q be the query box with $\mathcal{X}^Q = \{\mathbf{x}_1^Q, ..., \mathbf{x}_{n_Q}^Q\}$ local descriptors and let $\mathcal{X}^R = \{\mathbf{x}_1^R, ..., \mathbf{x}_{n_R}^R\}$ be the local descriptors of a test box R. The cosine similarity between histograms $\mathbf{V}_B^Q = [w_1^Q, ..., w_k^Q]$ and $\mathbf{V}_B^R = [w_1^R, ..., w_k^R]$ is:

$$S_B^{QR} = \frac{1}{\|\mathbf{V}_B^Q\| \|\mathbf{V}_B^R\|} \sum_{i=1}^{k} w_i^Q w_i^R. \quad (3)$$

For the sake of clarity, we will drop the normalization term $\frac{1}{\|\mathbf{V}_B^Q\| \|\mathbf{V}_B^R\|}$ in the following elaboration. By expanding w_i^Q, w_i^R with $\sum_{z=1}^{n_Q} q_{ind}(\mathbf{x}_z^Q) == i, \sum_{j=1}^{n_R} q_{ind}(\mathbf{x}_j^R) == i$ and reordering the summations the equation turns to

$$S_B^{QR} = \sum_{j=1}^{n_R} \sum_{z=1}^{n_Q} (q_{ind}(\mathbf{x}_j^R) == q_{ind}(\mathbf{x}_z^Q)) \cdot 1. \quad (4)$$

We define the term $(q_{ind}(\mathbf{x}_j^R) == q_{ind}(\mathbf{x}_z^Q)) \cdot 1$ in Equation 4 as the pointwise similarity between $\mathbf{x}_j^R$ and $\mathbf{x}_z^Q$. Denoting $(q_{ind}(\mathbf{x}_j^R) == q_{ind}(\mathbf{x}_z^Q))$ by δ_{jz} we derive the pointwise similarity for BoW as

$$\hat{S_B}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \delta_{jz} \cdot 1. \quad (5)$$

The VLAD-similarity S_V^{QR} can be decomposed in a similar way into a summation of pointwise similarities, defined as

$$\hat{S_V}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \delta_{jz} < \mathbf{d}_j^R, \mathbf{d}_z^Q >, \quad (6)$$

where $\mathbf{d}_j^R$ and $\mathbf{d}_z^Q$ are the differences with the corresponding visual word centers. Replacing the exact difference vectors with the quantized versions, we derive

$$\hat{S_V}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \delta_{jz} \sum_{i=1}^{l} < \widetilde{q}(\mathbf{d}_{j_i}^R), \widetilde{q}(\mathbf{d}_{z_i}^Q) > . \quad (7)$$

As the space of the difference vectors has been reduced to a set of prototypical elements, the pairwise dot products $D(i, j)$ between prototypes can be pre-computed. Inserting the pre-computed values, we end up with

$$\hat{S_V}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \delta_{jz} \sum_{i=1}^{l} D(\widetilde{q_{ind}}(\mathbf{d}_{j_i}^R), \widetilde{q_{ind}}(\mathbf{d}_{z_i}^Q)). \quad (8)$$

In the same manner, the pointwise similarity measure for tFV approximated up to the m^{th} Gaussian, can be derived as follows:

$$\hat{S_A}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \sum_{f,h=1}^{m} \psi_{jz}^{fh} < \mathbf{d}_{jf}^R, \mathbf{d}_{zh}^Q >, \quad (9)$$

where

$$\psi_{jz}^{fh} = (r_j^f == r_z^h) \frac{\gamma_j(r_j^f) \gamma_z(r_z^h)}{\sqrt{\omega_{r_j^f}} \sqrt{\omega_{r_z^h}}}. \quad (10)$$

Inserting the pre-computed values, we arrive at

$$\hat{S_A}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \sum_{f,h=1}^{m} \psi_{jz}^{fh} \sum_{i=1}^{l} D(\widetilde{q_{ind}}(\mathbf{d}_{jf_i}^R), \widetilde{q_{ind}}(\mathbf{d}_{zh_i}^Q)). \quad (11)$$

The evaluation of sampled bounding boxes is as follows. The approach computes the score of each interest point of

the database image through the pointwise similarity measure described above, and obtains the score of a certain bounding box by summing the scores over the points which locate inside the box. Considering that the pointwise scores only need to be computed once and the box scores are acquired by simple summations, the proposed paradigm is well suited for evaluating a large number of boxes.

4. Locality in the feature space

In this section we continue on localizing the search in the feature space with two different tactics.

4.1. Large vocabularies

We employ large vocabularies in order to shrink the footprint of each word to a local comparison of close observations. This will suppress the confusion from irrelevant observations as they are less likely to reside in the same small cells as the query descriptors. Moreover, small visual clusters can better capture the details in the local feature space, enabling distinction between very similar observations.

It is practically infeasible to apply very large vocabularies directly in the standard VLAD and Fisher vector as the dimensionality of VLAD and Fisher representation grows linearly with the size of the vocabulary. However, the point-indexed representation described in the previous section allows the application of very large vocabularies in VLAD and Fisher vector effortlessly. Its memory consumption is independent of the size of the vocabularies, as for each point it only requires storing m numbers for tFV (and 1 for VLAD) to indicate the associated visual clusters.

4.2. Exponential similarity

In instance search it is reasonable to reward two descriptors with a *disproportionally* high weight when they are close, as we seek exact unique details to match with the detail of the one query instance. The pointwise similarities in equations 6 and 9 do not meet this property. We enhance locality in the feature space by *exponential similarity*.

Without loss of generality, we consider the VLAD case as an example to elaborate. The exponential pointwise similarity for VLAD coding is expressed as

$$S_V^{\hat{exp}}(\mathbf{x}_j^R, \mathbf{x}_z^Q) = \delta_{jz} \cdot exp(\beta \cdot f(\mathbf{d}_j^R, \mathbf{d}_z^Q)), \qquad (12)$$

where $f(\mathbf{d}_j^R, \mathbf{d}_z^Q)$ measures the cosine similarity of the two difference vectors, and β is a parameter which controls the shape of the exponential curve.

The rate of the change is captured by the first-order derivate. The derivate of the above exponential similarity function with respect to the cosine similarity is

$$\frac{\partial S_V^{\hat{exp}}(\mathbf{x}_j^R, \mathbf{x}_z^Q)}{\partial f(\mathbf{d}_j^R, \mathbf{d}_z^Q)} = \delta_{jz} \cdot exp(\beta \cdot f(\mathbf{d}_j^R, \mathbf{d}_z^Q)) \cdot \beta. \qquad (13)$$

Indeed, the rate of similarity change increases as the two observations get closer.

The proposed exponential similarity emphasizes locality in the feature space, putting disproportionally high weight on close matches.

5. Experiments

5.1. Experimental setup

Datasets. We evaluate the proposed methods on 3 datasets, namely Oxford buildings [26], Inria BelgaLogos [15]and Inria Holidays [11]. Oxford buildings contains 5,062 images downloaded from Flickr. 55 queries of Oxford landmarks are specified, each by a query image and a bounding box. BelgaLogos is composed of 10,000 press photographs. 55 queries are defined, each by an image from the dataset and the logo's bounding box. Holidays consists of 1,491 personal holiday pictures, 500 of them used as queries. For all datasets, the retrieval performance is measured in terms of mean average precision (mAP).

Local descriptors. We use the Hessian-Affine detector [18, 22] to extract interest points on Oxford5k and BelgaLogos while the public available descriptors are used for Holidays. The SIFT descriptors are turned into RootSIFT [3], and the full 128D descriptor is used for VLAD as in [4], while for Fisher vector and tFV, the local descriptor is reduced to 64D by PCA, as [13, 31] have shown PCA reduction on the local descriptor is important for Fisher vector, and hence also for tFV.

Vocabularies. The vocabularies for Oxford buildings are trained on Paris buildings [27], and the vocabularies for Holidays are learned from Flickr60k [11], the same as in [4]. For BelgaLogos the vocabularies are trained on a random subset of the dataset.

5.2. Truncated Fisher vector

We first evaluate the performance of tFV with different values of m, which controls the number of Gaussian components each SIFT descriptor is associated with. We compare tFV with the original Fisher vector under the same setting, where a GMM with 256 components is learned to model the feature space and the full database image is used during the search.

As shown in Figure 2, m has little impact on the result. tFV and the original Fisher vector have close performance. In the following experiments, we set $m = 2$ for tFV.

5.3. Spatial locality in the image

In this experiment we test whether adding spatial locality by analyzing multiple bounding boxes in a test image improves the retrieval performance, as compared to the standard global retrieval paradigm where only the full image is evaluated. For the localized search, we use the highest

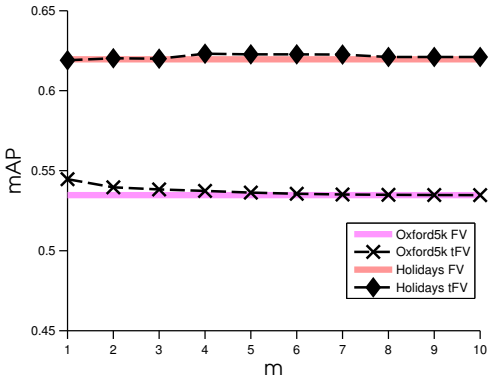

Figure 2. **Impact of the parameter m on the performance of tFV.** The parameter m controls the number of Gaussian components each point is assigned to. The straight line is for $m = 256$, the standard Fisher vector implementation. It is clear that the first assignment is by far the most important one.

scored box as the representative of the image to rank the test examples. We use the same vocabulary with 256 visual clusters for both global retrieval and localized retrieval. In order to ensure a fair comparison and show the influence of spatial locality, we apply ℓ_2 normalization in all cases. The results are shown in Table 1.

Localized search has a significant advantage on Oxford5k (landmarks) and BelgaLogos (small logos), in short for fixed shape things, while on the scene-oriented Holidays dataset, global search works slightly better.

When searching for an object which occupies part of the image, see Figure 3, introducing spatial locality is beneficial, as the signal to noise ratio within the bounding box is much higher than the entire image, especially for small non-conspicuous objects. However, when looking for a specific scene which stretches over the whole picture, adding spatial locality cannot profit. As whether it is an edifice, a logo, an object or alternatively a scene is a property of the query, it can be specified with a simple question at query-time whether to use locality or globality in the search.

5.4. Feature space locality by large vocabularies

In this section we evaluate the effectiveness of large vocabularies which impose locality in feature space by creating small visual clusters. Table 2 lists the retrieval accuracy. It shows increasing the vocabulary size improves the performance in all cases.

Large vocabularies better capture the small details in the feature space, advantageous for instance search where the distinction between close instances of the same category relies on subtle details. However, there is no infinite improvement. We have also tested VLAD200k on Oxford5k and BelgaLogos, and the mAP is 0.723 and 0.266 respectively, no further increase compared to VLAD20k. Creating a GMM with 200k Gaussian components is prohibitively

	VLAD		**tFV**	
	global [13]	*local*	*global* [13]	*local*
Oxford5k	0.505	0.576	0.540	**0.591**
BelgaLogos	0.107	0.205	0.120	**0.219**
Holidays	0.596	0.597	**0.620**	0.610
Generic	0.403	0.460	0.427	**0.473**

Table 1. **The influence of spatial locality.** Localized search evaluates multiple locations in a database image and takes the highest scored box as the representative, while global search [13] evaluates the entire image. To ensure a fair comparison and show the influence of spatial locality, we use the same vocabularies with 256 clusters and ℓ_2 normalization for both localized search and global search. Localized search is advantageous on object-oriented datasets, namely Oxford5k and BelgaLogos, while on scene-oriented Holidays, global search works slightly better. As the average mAP over the three datasets in the last row shows, the proposed localized search is generic, working well on a broad set of instances.

	VLAD			**tFV**		
	256	*2048*	*20k*	*256*	*2048*	*20k*
Oxford5k	0.576	0.670	0.724	0.591	0.673	**0.734**
BelgaLogos	0.205	0.246	0.271	0.219	0.241	**0.280**
Holidays	0.597	0.667	0.727	0.610	0.684	**0.737**
Generic	0.460	0.528	0.574	0.473	0.533	**0.584**

Table 2. **The influence of vocabulary size.** Three sets of vocabularies are evaluated for box search, with 256, 2048 and 20k visual clusters respectively. Increasing the vocabulary size leads to better performance for all datasets.

expensive in terms of computation, but we expect the same behavior as VLAD. The quantified differentiation within the visual clusters will be superfluous or even adverse when the visual cluster is so small that the hosted local descriptors represent the same physical region in the real world. Before reaching the gate, large vocabularies are beneficial.

5.5. Feature space locality by exponential similarity

In this experiment we quantify the add-on value of the proposed exponential similarity, see equation 12, which emphasizes close matches in feature space, as compared to the standard dot product similarity. We set $\beta = 10$ for all datasets without further optimization. We embed the evaluation in the box search framework using 20k-vocabularies. As shown in Table 3, the exponential similarity consistently improves over dot-product similarity by a large margin. Exploring a similar idea, the thresholded polynomial similarity in the concurrent work [32] achieves a close performance. We have also experimented with the adaptive similarity [29]. Giving much higher weights to closer matches has the most important effect on the result. Both [29] and our proposal provide this, where our proposal does not

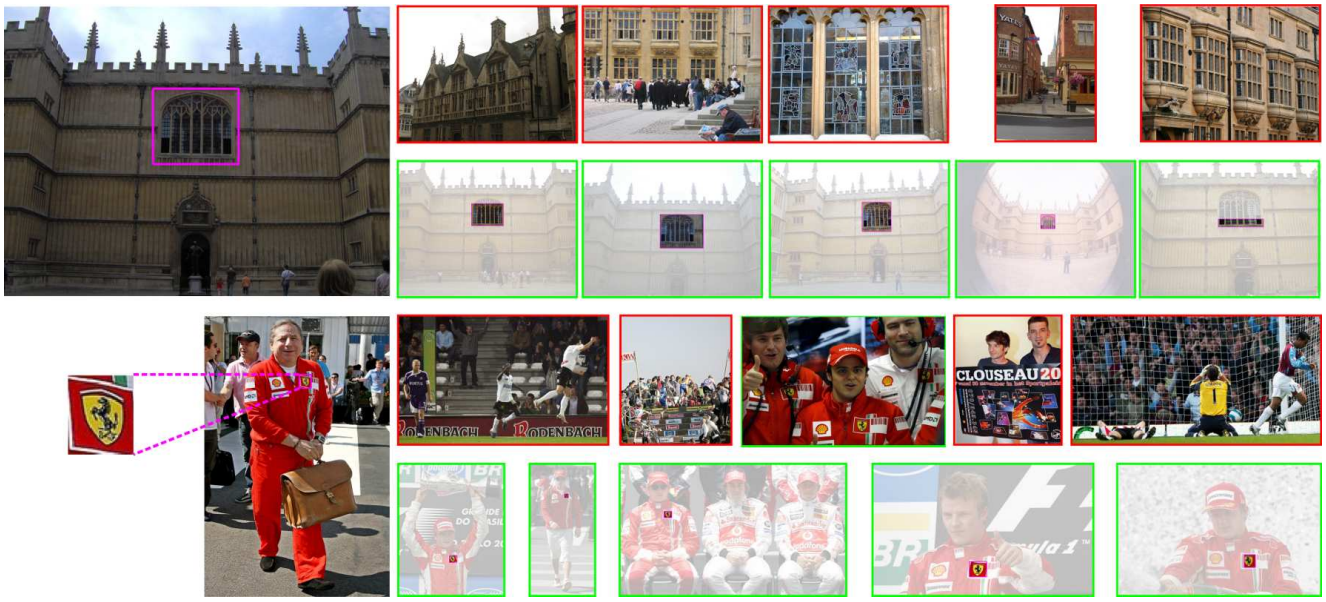

Figure 3. **The effect of spatial locality.** Query instances are shown on the left, delineated by the bounding box. On the right are the top 5 retrieved examples. For each query example, the upper row and lower row are results returned by global search and localized search respectively. Positive (negative) samples are marked with green (red) borders. Focusing on local relevant information, localized search has successfully ranked *and* discovered the instance despite the presence of a noisy background.

	VLAD			tFV		
	dot	*exp*	*poly*	*dot*	*exp*	*poly*
Oxford5k	0.724	0.765	0.773	0.734	0.770	**0.778**
BelgaLogos	0.271	0.291	0.296	0.280	0.302	**0.304**
Holidays	0.727	0.772	0.749	0.737	**0.787**	0.767
Generic	0.574	0.609	0.606	0.584	**0.620**	0.616

Table 3. **The effect of exponential similarity.** The value of the exponential similarity, denoted by *exp*, is evaluated within the box search framework using 20k-vocabularies. As compared to the dot-product similarity, denoted by *dot*, the exponential similarity improves the search accuracy in all cases. *poly* denotes the thresholded polynomial similarity function proposed in the recent work [32].

need the extra learning step. Putting *disproportionally* high weights on close matches in the feature space is advantageous for instance search, which relies on matches of exact unique details.

5.6. State-of-the-art comparison

To compare with the state of the art in generic instance search from one example, in Table 4 we have compiled an overview of the best results from [4, 8, 13, 24] which employ VLAD or Fisher vector. For BelgaLogos where VLAD and Fisher vector have not been applied before, we report results acquired by our implementation. The proposed localized tFV20k with exponential similarity outperforms all other methods by a significant margin. The method is followed by localized VLAD20k^{exp}.

	VLAD			Fisher vector		
	[4]	[8]	20k^{exp}	[13]	[24]	tFV20k^{exp}
Oxford5k	0.555	0.517	0.765	0.418	-	**0.770**
BelgaLogos	0.128*	-	0.291	0.132*	-	**0.302**
Holidays	0.646	0.658	0.772	0.634	0.705	**0.787**
Generic	0.443	-	0.609	0.395	-	**0.620**

Table 4. **State-of-the-art comparison.** The entries indicated with a $*$ are our supplementary runs of the reported methods on that dataset. Our combined novelty, localized tFV20k with exponential similarity outperforms all other methods by a considerable margin.

For the newly defined TRECVID50k dataset, which is a factor of 5 to 30 bigger than the other three datasets, and covering a much larger variety, the performance improvement of our subsequent steps is indicated in the rows of Table 5.

6. Conclusion

We propose locality in generic instance search from one example. As the signal to noise ratio within the bounding box is much higher than in the entire image, localized search in the image for an instance is advantageous. It appears that continuing on the localization in the feature space by using very large vocabularies further improves the results considerably. Finally, localizing the similarity metric by exponen-

	VLAD	tFV
Baseline (*global search*)	0.075	0.096
+ Spatial locality	0.084	0.116
+ 20k vocabulary	0.103	0.131
+ Exponential similarity	0.124	0.144

Table 5. The performance improvement by the three novelties on the TRECVID50k dataset. The dataset is a 50k subset of the TRECVID 2012 instance search dataset [21] with annotations for 21 queries, here applied with 1 example each.

tial weighting, improves the result significantly once more.

The combination of spatial locality and large vocabularies either will pose heavy demands on the memory or on the computation. In the standard implementation even a vocabulary of 256 clusters with box search will require a huge 777 gigabytes and over 2,000s of computation to finish one query for Oxford5k. The implementation of [13] achieves an mAP of 0.490 using PCA and product quantization on a 256 vocabulary with a memory of 1.91 gigabytes. This will explode for larger vocabularies. Our implementation with point-indexed representation requires only 0.56 gigabytes for a 20k vocabulary, achieving a vast increment to an mAP of 0.765 with a computing time of 5s. The computation time can be improved further by the use of hierarchical sampling schemes, a topic of further research.

On the newly proposed TRECVID50k dataset, which contains many diverse instances, we have set an mAP with one query example of 0.144. On the commonly used datasets Oxford5k, BelgaLogos, and Holidays we achieve an average performance increase from 0.395 for the recent [13], and 0.443 [4] to 0.620 for our generic approach to instance search with one example proving the value of locality in the picture and feature space for this type of search. The method does not only improve the accuracy but also delivers a reliable localization, opening other avenues, most notably complex queries asking for spatial relations between multiple instances.

Acknowledgments This research is supported by the Dutch national program COMMIT and the STW STORY project.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 34(11):2189–2202, 2012.

[2] R. Arandjelovic and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

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

[4] R. Arandjelović and A. Zisserman. All about VLAD. In *CVPR*, 2013.

[5] Q. Chen, Z. Song, R. Feris, A. Datta, L. Cao, Z. Huang, and S. Yan. Efficient maximum appearance search for large-scale object detection. In *CVPR*, 2013.

[6] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total recall II: query expansion revisited. In *CVPR*, 2011.

[7] 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 *CVPR*, 2007.

[8] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez. Revisiting the VLAD image representation. In *MM*, 2013.

[9] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[10] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.

[11] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[12] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33(1):117–128, 2011.

[13] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.

[14] Y. Jiang, J. Meng, and J. Yuan. Randomized visual phrases for object search. In *CVPR*, 2012.

[15] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *MM*, 2009.

[16] C. H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *ICCV*, 2009.

[17] Z. Lin and J. Brandt. A local bag-of-features model for large-scale object retrieval. In *ECCV*, 2010.

[18] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[19] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV*, 2010.

[20] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[21] P. Over, G. Awad, J. Fiscus, G. Sanders, and B. Shaw. Trecvid 2012 - an introduction of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2012.

[22] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[23] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[24] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, 2010.

[25] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

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

[27] 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*, 2008.

[28] 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.

[29] D. Qin, C. Wengert, and L. van Gool. Query adaptive similarity for large scale object retrieval. In *CVPR*, 2013.

[30] J. Revaud, M. Douze, and C. Schmid. Correlation-based burstiness for logo retrieval. In *MM*, 2012.

[31] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: theory and practice. *IJCV*, 105(3):222–245, 2013.

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

[33] G. Tolias, Y. Kalantidis, and Y. Avrithis. Symcity: feature selection by symmetry for large scale image retrieval. In *MM*, 2012.

[34] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha. The visual extent of an object: suppose we know the object locations. *IJCV*, 96(1):46–63, 2012.

[35] J. R. R. Uijlings, K. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.