# A Comprehensive Study over VLAD and Product Quantization in Large-scale Image Retrieval

Eleftherios Spyromitros-Xioufis, Symeon Papadopoulos, Ioannis (Yiannis) Kompatsiaris, *Senior Member, IEEE*, Grigorios Tsoumakas and Ioannis Vlahavas, *Member, IEEE*

***Abstract*—This paper deals with content-based large-scale image retrieval using the state-of-the-art framework of VLAD and Product Quantization proposed by Jegou et al. [1] as a starting point. Demonstrating an excellent accuracy-efficiency trade-off, this framework has attracted increased attention from the community and numerous extensions have been proposed. In this work, we make an in-depth analysis of the framework that aims at increasing our understanding over its different processing steps and boosting its overall performance. Our analysis involves the evaluation of numerous extensions (both existing and novel) as well as the study of the effects of several unexplored parameters. We specifically focus on a) *employing more efficient and discriminative local features*, b) *improving the quality of the aggregated representation*, and c) *optimizing the indexing scheme*. Our thorough experimental evaluation provides new insights into extensions that consistently contribute and others that do not to performance improvement, and sheds light into the effects of previously unexplored parameters of the framework. As a result, we develop an enhanced framework that significantly outperforms the previous best reported accuracy results on standard benchmarks and is more efficient.**

***Index Terms*—Image retrieval, indexing, image classification.**

## I. INTRODUCTION

THIS paper deals with large-scale image retrieval that is defined as the problem of finding in a large database of images (e.g. 100 million), those that depict the same object or scene with a query image under variations such as 3D viewpoint and lighting changes, object deformations, or the presence of occlusions and clutter. This definition includes but is not limited to near- and partial-duplicate images. Near-duplicates are edited (scaled, format changed, etc.) versions of the same image and partial-duplicates are padded or cropped near-duplicates [2]. A large-scale image retrieval system has many important applications, ranging from object retrieval [3], [4] to location and landmark detection [5], copyright violation detection [6], representative image selection [7], and more recently *visual meme* discovery in social media [8].

Until recently, state-of-the-art methods in large-scale image retrieval relied on the bag-of-words (BoW) representation [3], [9]. According to this, local features (usually SIFT [10]) are extracted from each image and each feature is assigned to the nearest visual word from a visual vocabulary. The result of this process is a high-dimensional and sparse histogram vector for each image. Such vectors are compared with standard similarity measures (e.g. cosine) and can be searched efficiently using established text-retrieval techniques such as inverted list structures. Several attempts have been made to improve the accuracy of BoW, with soft-assignment [11] and Hamming-Embedding (HE)[12] being among the most successful methods. In soft-assignment each feature is mapped to a weighted set of visual words, thus enriching the representation at the cost of increased memory requirements. HE augments the inverted list structure with a binary signature that encodes the approximate location of each feature in the Voronoi cell. This approach not only increases BoW's accuracy but is also faster. Motivated by the facts that BoW ignores spatial information and the cost of full geometric verification is prohibitive, [12], [13] derived methods that embed spatial information for each feature in BoW's inverted list structure. Such methods can boost the accuracy of BoW, especially for partial duplicate image retrieval [14] but this comes again at the cost of increased complexity and memory requirements.

Despite their success, BoW-based methods cannot scale to more than few million images (on a single machine) due to computational and memory constraints [1], [15]. In an attempt to address the scalability issues, prior art has focused on compressing the BoW vectors [16]–[18]. However, to approximate the initial accuracy of a high-dimensional BoW vector, these methods require thousands of bytes per image.

Recently, a number of more scalable approaches have been developed [1], [19], [20] that employ more discriminative vector representations than BoW and combine them with powerful compression techniques. One of the most successful frameworks of this type, with respect to the accuracy-efficiency trade-off, is presented in [1]. This framework still relies on SIFT features but replaces BoW with the highly discriminative Fisher Vector [20] representation or its simpler variant, named VLAD (Vector of Locally Aggregated Descriptors) [19]. Using these optimized vector representations, significantly better results are obtained, compared to a BoW vector with similar dimensionality. Beyond the optimized vector representation, the success of the framework is due to a powerful indexing

This work was supported by the SocialSensor FP7 project, partially funded by the EC under contract number 287975.

E. Spyromitros-Xioufis is with the Information Technologies Institute / Centre for Research and Technology Hellas (ITI/CERTH), Greece, and with the Department of Informatics, Aristotle University of Thessaloniki (AUTH), Greece, E-mail: espyromi@iti.gr, espyromi@csd.auth.gr.

S. Papadopoulos, I. Kompatsiaris are with ITI/CERTH, Greece, E-mail: {papadop,ikom}@iti.gr.

G. Tsoumakas and I. Vlahavas are with the Department of Informatics, AUTH, Greece, E-mail: {greg,vlahavas}@csd.auth.gr.

This paper has supplementary downloadable material (a pdf file with additional experimental results) available at http://ieeexplore.ieee.org, provided by the authors.

scheme that jointly optimizes dimensionality reduction and indexing. First, Principal Component Analysis (PCA) is applied to significantly reduce the dimensionality of the vectors while having a negligible impact in accuracy for a moderate amount of reduction. Subsequently, the reduced vectors are indexed and searched efficiently using the recently proposed Product Quantization (PQ) method [21], which was shown to outperform a number of state-of-the-art indexing methods including Spectral Hashing [22] and FLANN [23].

The framework of [1], hence denoted as *VLAD+PQ*, has demonstrated very good results in terms of search accuracy, significantly outperforming previous state-of-the-art approaches [18]–[20] when a similar efficiency setup is employed. For instance, the performance of BoW with a vocabulary of 20K visual words can be obtained using only 128 bits per image. With such a small code size, 100 million images can fit in approximately 2 GB of RAM and be searched in 250 ms on a single core. However, such an aggressive compression results in a significant decrease of the initial search accuracy. In this paper, we attempt to further improve the accuracy of this framework, with special focus on extremely efficient setups in terms of memory usage and query response times. To this end, we perform an in-depth, end-to-end analysis of the framework and show that its performance can be significantly improved by incorporating extensions and optimizing the individual processing steps of its pipeline.

We initially focus on employing better local features and compare (in Subsection V-A) the performance of VLAD vectors generated using SURF [24], SIFT, RootSIFT [25] and CSURF features. CSURF is a new SURF-based color feature presented in Subsection III-A. The results indicate that SURF and CSURF are not only more efficient but also lead to significantly increased accuracy. Next, we attempt to improve the quality of the vectorized representation by applying two types of feature filtering methods described in Subsection III-B. Methods of the first type are inspired from [2] and perform *filtering based on the richness of feature structure*, while methods of the second type are inspired from [26] and perform *filtering based on a feature-vocabulary relation*. As shown in SubsectionV-B, while some of these methods can offer small improvements when full-dimensional VLAD vectors are employed, this is not the case with PCA-projected vectors. These results are interesting as they contradict the results presented in [26].

Subsequently, we move our focus towards improving the vector aggregation and dimensionality reduction steps. In Subsection V-C we evaluate the *mean-aggregation* strategy proposed in [27] for VLAD and find that it results in a significant performance degradation compared to the standard aggregation method of VLAD. Subsections III-E and III-F study the recently proposed *whitening* and *multiple vocabulary aggregation* methods [15]. Both methods were mainly studied within the context of BoW vectors with particular success but several details remain unclear regarding their applicability on VLAD. The thorough empirical analysis presented in Subsections V-E and V-F sheds light into these issues and shows that both methods can offer significant performance improvements. We also study the effects of important vector generation parameters such as vocabulary size and projection length in Subsection V-D.

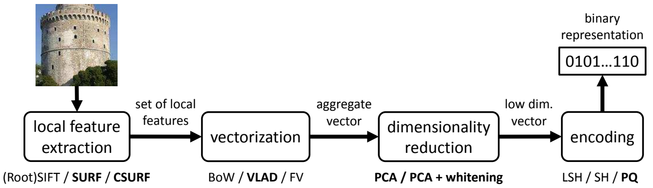


Fig. 1: Steps of the feature extraction and indexing pipeline. Different options for each step are presented, with the selected appearing in bold.

Finally, we focus on improving the performance of the PQ indexing scheme by studying the effects of previously unexplored parameters in Subsection V-H. Additionally, in Subsection V-I we show that besides large-scale image retrieval, the studied VLAD extensions can offer significant improvements in large-scale image classification.

Our main contributions are as follows:

- By combining the above improvements, we manage to *significantly outperform the previous best reported accuracy results on standard benchmarks and at the same time improve the efficiency of the VLAD+PQ framework*.
- The thorough experimental study we conducted increases our understanding of the effects of previously unexplored parameters of the studied framework.
- We evaluate numerous existing and new extensions and provide insights into working and non-working ones.
- We make publicly available an efficient open-source implementation[1] of the image processing, aggregation and indexing methods as well as an implementation of our experimental testbed that facilitates easy reproduction of our experimental results[2].

To the best of our knowledge, this is the first work that contains such an extensive empirical study, covering all the individual steps of the studied large-scale image retrieval framework.

## II. Background

This Section introduces the VLAD+PQ framework [1] along with some alternative choices for its different processing steps. As shown in Fig. 1 there are four main steps involved in the process of transforming an input image into a small fixed-length code: (a) local feature extraction, (b) vectorization, i.e. the aggregation of local features in a single vector, (c) dimensionality reduction, and (d) quantization and indexing. Steps (a), (b) are described in Subsection II-A and steps (c), (d) in Subsection II-B.

### A. Feature Extraction and Vectorization

The presented study explores the performance of systems that rely on the extraction of local features which are aggregated using some pooling method to create a global image representation. Compared to systems that use global features

[1] https://github.com/socialsensor/multimedia-indexing

[2] http://www.socialsensor.eu/results/software/79-image-search-testbed

such as GIST [28], systems based on local features are more robust to geometric transformations and typically exhibit better performance. Among local features, SIFT have shown excellent performance and are established as the features of choice for most systems. However, their increased computation time has motivated the design of alternative, more efficient features, with SURF being among the most successful, as they can be computed several times faster, while being comparable with respect to repeatability, distinctiveness, and robustness.

Regardless of the type of local features, the feature extraction step generates a set $L$ of $D$-dimensional feature vectors $x = [x_1, .., x_D]$ for each image. Usually, hundreds to thousands of vectors are extracted. To make large-scale search tractable, an aggregation method is usually employed that summarizes $L$ into a single, fixed-length vector representation. Until recently, BoW has been the most popular method of this type. In BoW, a codebook $C = \{c_1, ..., c_k\}$ of $k$ visual words is computed offline (typically by applying k-means clustering on a large set of features). Then, given the set of local features extracted from an image, each feature is quantized to its closest cluster centroid. The BoW vector is the $k$-dimensional histogram of the distribution of visual words in an image.

Recently, a number of new representations emerged [19], [20], [29] that encode higher order statistics, compared to BoW, of the distribution of features to visual words. The studied framework is based on VLAD, a state-of-the-art method presented in [19]. As in BoW, a codebook is first computed and each feature is quantized to its closest centroid. However, instead of simply counting the features assigned to each centroid, VLAD records their position relatively to it by accumulating the residual vectors $x - c_i$ of the features $x$ assigned to each visual word $c_i$ into a vector:

$$v_i = \sum_{x \in L_{c_i}} x - c_i \tag{1}$$

where $L_{c_i} \subseteq L$ is the set of features assigned to $c_i$. The VLAD vector $v$ is the concatenation of all $v_i$ and is therefore $d = k \times D$-dimensional. As a final step, $v$ is first power- and then L2-normalized. Power-normalization discounts the influence of large components (usually coming from visual bursts) [1] by raising each component of $v$ to a power of $a \in [0, 1]$, whereas L2 normalization makes the representation invariant to the number of features extracted from each image. Very recently, *intra-normalization* [30] and *residual-normalization* [31] were proposed as alternative schemes to address the problems of burstiness and unequal contribution of individual features, respectively. Intra-normalization consists of applying L2 normalization separately on each $v_i$ and then L2 normalizing the entire vector (power normalization is omitted), while in residual-normalization the residual vector $x - c_i$ of each feature $x$ from its nearest visual word $c_i$ is separately normalized to unit length and power+L2 normalization is still applied on the entire VLAD. Both schemes were found to outperform the power+L2 normalization scheme. However, we opt for using the normalization scheme of [1] (power+L2 normalization with $a$=0.5) throughout this paper for the following reasons: a) according to [30] intra-normalization is consistently better than power+L2 normalization only for well adapted visual vocabularies which are difficult to obtain in very-large scale and dynamic setups (as discussed in Subsection III-B2), b) residual-normalization was compared in [31] only against power+L2 normalization with $a$=0.2 instead of $a$=0.5 suggested as near-optimal in [1].

VLAD is significantly more accurate than BoW when a representation of equal dimensionality is used and at the same time it is cheaper to compute as it requires a much smaller visual vocabulary. Lately, [1] showed that VLAD can be considered as a simplified version of Fisher Vector (FV) [20]. Empirical results, however, suggest that although FV yields better performance than VLAD when full-vectors are used, VLAD performs equally well and in some cases better when vector dimensionality is reduced by PCA [1]. This ability of VLAD to retain an excellent accuracy after heavy dimensionality reduction together with its slightly more efficient computation compared to FV, makes it an ideal representation for large-scale image retrieval systems.

### *B. Dimensionality Reduction, Indexing and Search*

Using the 128-dimensional SIFT descriptor and a small vocabulary of $k = 64$ centroids results in 8192-dimensional VLAD vectors or 32KB of memory per image. This size is prohibitive for large-scale search applications due to memory and search efficiency constraints. To address these issues, compression and binarization techniques are usually employed (e.g. Locality-Sensitive Hashing (LSH) [32], Spectral Hashing (SH) [22]) to transform the vectors into binary codes that have a small memory footprint (all images can fit in main memory) and can be searched efficiently. The adopted framework follows a similar approach. First, the dimensionality of VLAD vectors is significantly reduced with PCA and then PQ [21] is applied to compress the projected vectors. PQ significantly outperforms other state-of-the-art binarization schemes in terms of accuracy for the same efficiency setting.

Quantization is used to reduce the cardinality of a representation space by mapping a $d$-dimensional vector $x \in \mathbb{R}^d$ to a vector $q(x) \in C = \{c_0, c_1, .., c_{k-1}\}$. $C$ is a finite set of reproduction values $c_i \in R^d$ that correspond to the centroids of a k-means clustering and $q$ maps each vector to its closest centroid. A quantizer with $k$ centroids encodes each vector with $B = \log_2(k)$ bits. Given a query vector $y$, a set of database vectors $X = \{x_1, ..., x_n\}$ and a quantizer $q(.)$, the nearest neighbors of $y$ in $X$ can be efficiently found using the Asymmetric Distance Computation (ADC) approach [21]. In ADC, each vector $x_i \in X$ is replaced by its reproduction value $c_i$, while the query vector $y$ is not encoded. The nearest neighbors of $y$ are found by computing the distance of $y$ to every centroid and returning the database vectors that are quantized to the closest centroid. To achieve good vector approximation, however, a large number of centroids is required e.g. $2^{64}$, producing a 64bit code. With such a large number of centroids, learning a k-means quantizer, assigning vectors to centroids, storing the centroids in memory and searching are intractable. *PQ* is a technique that makes this problem tractable by defining a large quantizer as the Cartesian product of smaller quantizers. A $d$-dimensional vector $x$ (in this case a PCA-projected VLAD vector) is first split into $m$ subvectors $x^1, ..., x^m$ of equal lengths $d^* = d/m$ and each subvector is quantized using a separate quantizer. Thus, a product quantizer $q$ is defined as a function $q(x) = (q_1(x^1), ..., q_m(x^m))$ that maps a vector $x$ to a tuple of $m$ indices, one for each subvector. While each individual quantizer $q_j$ has only $k_s$ reproduction values, the set of centroids induced by $q$ is $k = (k_s)^m$. To distinguish between different PQ schemes the notation $m \times b_s$ is used, specifying a product quantizer with $m$ subquantizers that encode each vector with $b_s = \log_2 k_s$ bits. The total number of bits used to encode a vector in this case is $B = mb_s$. Using PQ and the ADC approach, the nearest neighbor is found by computing $NN(y) = \arg\min_i \sum_{j \in [1,m]} ||y^j - q_j(x_i^j)||^2$. In order to search efficiently in $X$, the distances between each subvector $y^j$ of a query image $y$ and the $k_s$ centroids of the respective subquantizer $q_j$ are computed and stored in look-up tables before scanning the database. While PQ+ADC enables fast, approximate nearest neighbor search and a remarkable reduction in memory requirements, the search is still exhaustive. In order to scale to billions of vectors, [21] proposed a non-exhaustive variant that combines ADC with an inverted file structure (IVFADC). Compared to PQ+ADC, PQ+IVFADC requires an additional memory of approximately 4 bytes per image due to the overhead of the identifiers that need to be explicitly stored. However, PQ+IVFADC is significantly faster than PQ+ADC in very large databases and also more accurate because it encodes the residual of each vector from the centroid of a coarse quantizer rather than the vector itself.

## III. Studied Aspects & Extensions

This section describes and motivates the issues studied in this empirical study. These issues pertain either to extensions or to parameter exploration in different steps of the framework. The discussion is structured along the following: a) local features, b) feature filtering techniques, c) aggregation strategies, d) vocabulary size and target dimensionality of PCA, e) whitening, f) multiple vocabulary aggregation, and g) PQ parametrization.

### *A. Local Features*

The type of local features being employed is probably the most critical design choice in the VLAD+PQ pipeline as it heavily affects the system's response time but also the quality of search results (as we show in Section V-A). VLAD was originally combined with SIFT features in [19] whereas a comparison between SIFT and PCA-SIFT presented in [1] showed that SIFT features lead to slightly better results for VLAD. Revisiting the issue of applying PCA on SIFT features, [31] found that better results can be obtained when only centering and rotation to a new uncorrelated basis are applied. Recently in [30], [31], SIFT were replaced by RootSIFT features [25], leading to significant accuracy gains. Despite their widespread use in VLAD-based systems, both SIFT and features derived from SIFT such as PCA-SIFT and RootSIFT suffer from increased computation time which can severely impact the system's overall response time. For example, extracting SIFT from a medium-sized (512x384) image on a single-core takes 350 ms on average, while vectorization and search against a database of 10 million images with the VLAD+PQ framework requires less than 100 ms.

Motivated by this limitation, in [33] we studied the replacement of SIFT with SURF features and found that SURF in addition to being about three times more efficient, compare favorably with respect to accuracy to reference results of the VLAD+SIFT combination from [1]. In Section V-A we perform an *extended evaluation* that includes: a) SURF, b) SIFT, c) RootSIFT and d) a new SURF-based color feature color feature that is described in the next paragraph.

**CSURF:** Lately, color extensions of SIFT features [34] have shown increased discriminative power compared to standard SIFT and have beem widely adopted for image and video classification. However, there are very few works that consider color features in the domain of large-scale image retrieval (e.g. [35]) and, to the best of our knowledge, no one has evaluated the use of color features in the context of retrieval with VLAD. This is not surprising given the previous discussion on SIFT's efficiency and the fact that color extensions require even more extraction time and memory. To make the use of color features in the domain of large-scale image retrieval more practical, we *propose a SURF-based, efficient color feature*, *CSURF*. The idea of combining SURF with color information has previously appeared in [36] where a "Color-SURF" descriptor was evaluated in the context of image matching through matching of the corresponding descriptor sets and was found better than SURF in a popular benchmark. "Color-SURF" is the concatenation of the original SURF descriptor with a color kernel histogram that is calculated around each interest point. In order to compute the distance between two such descriptors, a different distance measure (Euclidean/Bhattacharyya) is used to compare each component (SURF/color histogram) of each vector. Therefore, applying k-means clustering on "Color-SURF" as required by VLAD is not straightforward. Here, we design a different SURF-based color descriptor, *CSURF*, that follows the successful and more principled paradigm of RGB-SIFT [34] on how to incorporate color information while maintaining the invariance properties of the descriptor.

In order to extract CSURF, the image is first transformed to grayscale and interest points are computed using the standard SURF algorithm. Then, instead of computing the SURF descriptor of each interest point on the intensity channel, CSURF computes three SURF descriptors, one on each color band. The final CSURF descriptor is their concatenation. Calculated in this way, CSURF is equivalent to RGB-SIFT [34] but using the SURF algorithm for keypoint detection and description. However, differently from RGB-SIFT where each band's descriptor is normalized independently to unit length, we apply L2-normalization on the entire descriptor only. The intuition is that by normalizing the entire descriptor, we retain relative color intensity information that is lost otherwise. In contrast to "Color-SURF", CSURF can be compared using the Euclidean distance and are therefore directly pluggable to the VLAD pipeline.

### *B. Feature Filtering*

*1) Filtering based on the richness of feature structure:* In [2], a near-duplicate image detection framework was proposed that, departing from quantization-based approaches, based its search on raw SIFT features. Among the main contributions of that work was the introduction of a filtering technique that discards SIFT features with poor internal structure such as those extracted from homogeneous or near-empty image regions. To measure the richness of internal structure of a SIFT feature, [2] uses the Shannon *entropy* that is calculated by treating each SIFT feature as 128 samples of a discrete random variable. It was shown that SIFT features that generate false matches exhibit relatively smaller entropy values (on average) than those generating true matches. Thus, by discarding such features the false positive rate is drastically decreased and a single match between the remaining features of two images is sufficient for establishing near-duplicity.

Motivated by these results, *we explore whether such filtering techniques can also improve the accuracy of image retrieval frameworks that employ feature pooling methods.* To the best of our knowledge, this has not been done before. Specifically, we apply entropy-based filtering to discard poorly structured SURF features (which are found to outperform SIFT in Section V-A) and then perform VLAD aggregation using only the retained ones. We also evaluate *variance* as an alternative filtering criterion, again treating the components of each feature vector as samples of a discrete random variable. Variance measures the spread of a distribution and its use is motivated by the fact that entropy cannot distinguish between two values of a (discretized) SIFT or SURF feature that lie very close (e.g. 1 and 2) and two values which are maximally apart (e.g. 1 and 128). For instance, the features $F_1 = [1, 2, ..., 1, 2]$ and $F_2 = [1, 256, ..., 1, 256]$ have the same entropy. However, the structure of the second is obviously richer. Both filtering methods, along with random filtering are evaluated against no filtering in Section V-B1.

*2) Filtering based on a feature-vocabulary relation:* Attempting to improve the quality of the VLAD representation, [26] and [37] proposed methods that deal with outlier features, i.e features that lie close to the boundaries of the Voronoi cells formed by a specific visual vocabulary. [37] proposes a soft assignment technique that assigns each feature to $k \geq 1$ nearest centroids, with $k$ being dynamically selected according to a nearest neighbor distance ratio. After assignment, weighted vector differences are calculated with nearest centroids receiving larger weights. A computationally more efficient approach is proposed in [26], where outlier features are omitted from the computation of VLAD. This method, denoted here as *dist*, discards all features whose distance from their closest visual word is above the $C^{th}$ percentile of the distribution of distances, of features assigned to this visual word. Percentiles for each visual word, are computed offline and outlier features are filtered during VLAD computation. The intuition behind both methods is that outlier features reduce the repeatability of VLAD since a small amount of distortion may cause them to be quantized to a different visual word, and as a result, a considerably different VLAD vector may be generated. On the other hand, we notice that in contrast to features lying close to cluster centroids (such as those coming from visual bursts), outlier features are less frequent and perhaps more discriminative. In Section V-B2 we evaluate the *dist* method as well as *two new filtering methods*, *std* and *ratio*, that are based on the same intuition. *std* retains only features whose distance from the closest visual word is at least $a$ standard deviations smaller than their average distance from all visual words, while *ratio* retains only features whose distance from the closest visual word is at least $b$ times smaller than their distance from the second closest visual word.

## C. Aggregation Strategy

As described in Section II-A, after the assignment of local features to visual words, VLAD uses a summation formula (1) to aggregate the residual vectors $v_i$ of the features assigned to each visual word $c_i$ . [27] proposed an alternative formula, denoted as *mean aggregation*, for aggregating the differences: $v_i = \frac{1}{|L_{c_i}|} \sum_{x \in L_{c_i}} x - c_i$. The difference with the original VLAD formula, denoted here as *sum aggregation*, is that the sum is normalized by the number of feature vectors quantized to each visual word $c_i$. We notice that by not normalizing the sum, the sum aggregation formula incorporates information about the number of features quantized in a particular visual word. A $v_i$ with a large norm indicates that many features are assigned to a similar position in the Voronoi cell defined by $c_i$. On the other hand, *by using mean aggregation this information is lost*, as graphically illustrated in the Appendix. Despite being counter-intuitive, [27] showed that mean aggregation significantly outperforms sum aggregation. In that study, however, the methods were tested only on full-dimensional VLAD vectors and performance was measured using an image-level ROC curve analysis, a non-standard method. In Section V-C, we reevaluate the effectiveness of this extension on both full-dimensional and PCA-projected VLAD vectors, using a standard evaluation protocol and *draw different conclusions*.

## D. Vocabulary Size and PCA

The accuracy of full VLAD vectors increases with increasing vocabulary sizes, as shown in [1]. Specifically, experiments with vocabularies up to $k = 4096$ centroids were performed indicating a sub-linear relationship between the number of centroids and mean Average Precision (mAP) on the Holidays dataset. However, there are two reasons that make large vocabularies unsuitable for large-scale retrieval with VLAD+PQ. The first reason is increased complexity: with large vocabularies it takes more time to assign features to centroids. The second reason has to do with accuracy: larger vocabularies produce higher-dimensional vectors that have a higher *projection error* when dimensionality reduction is applied. Note that, although optional, dimensionality reduction is a crucial step of the VLAD+PQ pipeline since the *quantization error* incurred by PQ (at the final step of the VLAD+PQ pipeline) is an increasing function of a vector's length. In Section V-D we extensively study the effect of dimensionality reduction with PCA on VLAD+SURF vectors generated from vocabularies of various sizes and projected to various lengths. A similar study was previously presented in [1] but was less extensive (fewer vocabulary sizes and projection lengths were tested) and focused on VLAD vectors that aggregate PCA-SIFT features.

### *E. Whitening*

It has been noted in [38] that visual words do not occur independently as implicitly assumed by common similarity measures used to compare BoW vectors. In fact, it was shown that visual word dependencies are common in large datasets and that by ignoring them the similarity between two image vectors can be over-counted, resulting in poor retrieval performance. Recently, [15] proposed a whitening operation that is performed jointly with dimensionality reduction to limit the impact of the co-occurrences problem on BoW and VLAD vectors. Given an image vector $x$, the vector is first PCA-projected and truncated to $d'$ components, and subsequently whitened and L2 normalized to a new vector $x' = \frac{diag(\lambda_1^{-0.5},...,\lambda_{d'}^{-0.5})Mx}{||diag(\lambda_1^{-0.5},...,\lambda_{d'}^{-0.5})Mx||}$, where M is the $d' \times d$ PCA matrix and $\lambda_i$ is the eigenvalue associated with the $i^{th}$ largest eigenvector. In [15] it was shown that BoW vectors projected to 128 dimensions using this joint dimensionality reduction and whitening step give significantly better results compared to BoW vectors projected to 128 dimensions with plain PCA. In Section V-E, we study (for the first time) *the impact of whitening* on VLAD vectors projected to various lengths and generated using various vocabulary sizes.

### *F. Vocabulary Sensitivity - Multiple Vocabularies*

To deal with the problem of vocabulary sensitivity, i.e. the fact that the similarity between two VLAD vectors is highly dependent on the visual vocabulary used to generate these vectors, [31] introduced a *cluster center adaptation* method. This method tries to maintain a vocabulary whose cluster centers are consistent with the current collection in the sense that the mean of all vectors assigned to a cluster over the entire collection is the cluster center. This is done by first moving the cluster centers to maintain consistency and then re-computing all the VLAD vectors according to the new cluster centers. In [31], this method is shown to improve the performance of full VLAD vectors compared to using a static visual vocabulary learned on a different collection. However, there are practical reasons that make this method incompatible with a large-scale framework. First, the VLAD re-computation step has a significant computational cost, especially as the collection's size increases. Second, when dimensionality reduction is applied the improvements incurred by vocabulary adaptation are diminished [31] since the adapted vectors can be considerably different than those used to learn the PCA matrix. The same holds for the subsequent application of PQ.

A better technique to deal with vocabulary sensitivity is presented in [15] where multiple vocabularies are used to alleviate quantization artifacts in the context of BoW and VLAD. The use of multiple visual vocabularies is a known technique for improving the quality of BoW vectors. A simple strategy consists of generating a set of different BoW vectors, one from each vocabulary, and then concatenating them into a single vector (as done e.g. in [3]). As shown in [15], in addition to reducing efficiency and increasing the memory requirements, the improvement in search quality offered by such methods is limited, mainly due to the redundancy introduced by multiple vocabularies. To address these problems, [15] proposed a joint dimensionality reduction of the multiple vectors. First, multiple (BoW or VLAD) vectors are produced independently and concatenated to a single vector that is L2 normalized. Then, the joint dimensionality reduction and whitening method described in Section III-E is applied. By exploiting the additional information provided by multiple vocabularies and at the same time removing the redundancy between them, this approach was shown to improve the performance of BoW and VLAD vectors reduced to 128 dimensions.

In Section V-F we perform a detailed analysis on the effectiveness of this method, specifically for VLAD. Concretely, we extend the experiments performed in [15] by comparing the performance of PCA-projected and whitened VLAD vectors coming from different combinations of multiple vocabularies with the performance of VLAD vectors coming from single vocabularies of the same total complexities. Furthermore, we perform experiments on more datasets and study additional projection lengths. Our extended analysis provides answers to the following questions regarding the applicability of this method on VLAD that are not conclusively answered in [15]:

- Is the use of multiple vocabularies beneficial compared to using a single vocabulary of the same complexity?
- How far can we go when considering multiple vocabularies? E.g. for a fixed total complexity of 256 visual words, which of vocabulary setups is better: 2×128 or 128×2?
- Do the observations hold for larger projection lengths?

### *G. Product Quantization Optimization*

All previous sections focused on issues related to the generation of a high quality, yet compact vectorized image representation. As discussed in Section III-D, the compactness requirement is imposed by the fact that the subsequent quantization scheme incurs smaller quantization error on vectors of smaller dimensionality. To deal with this trade-off, [1] proposed the minimization of the mean squared approximation error as an objective criterion for optimizing the dimension $d'$, having a fixed constraint on the number of bits $B$ used to represent each vector. The optimal projection length $d'$ is found by trying different values and selecting the one that minimizes this criterion on a learning set. However, the selected value of $d'$ using this criterion is not necessarily optimal with respect to a retrieval quality measure such as mAP. Furthermore, in [1] there is no discussion on what values to use for $m$ and $k_s$ (remember that $B$=$m \log_2 k_s$) and only two arbitrary quantization schemes (16x8 and 256x10) of different code sizes are evaluated. The effect of these parameters for a fixed code size is studied in [21] but only in the context of searching fixed-length local (SIFT) and global (GIST) vectors. [21] concludes that quantization schemes with small values for $m$ (number of subquantizers) and large values for $k_s$ (number of centroids) are better than having many subquantizers with few bits. Note, however, that $k_s$ cannot be

TABLE I: Summary of aspects and extensions studied in the experimental evaluation.

| Aspect | Summary | Related Work(s) | Sec. Discussed / Tested |
|---|---|---|---|
| local features | SURF vs SIFT vs RootSIFT vs CSURF | [25], [30], [31], [33] | III-A / V-A |
| feature filtering | richness of structure, methods: *entropy* / *variance* | [2] | III-B1 / V-B1 |
| | feature-vocabulary relation, methods: *dist* / *std* / *ratio* | [26], [37] | III-B2 / V-B2 |
| aggregation method | *sum* vs *mean* aggregation | [27] | III-C / V-C |
| vocabulary size & PCA | the effect of $k$ with respect to $d'$ | [1] | III-D / V-D |
| whitening | the effect of whitening with respect to $k$ and $d'$ | [15] | III-E / V-E |
| multiple vocabularies | the effect of multiple vocabularies with respect to $k$ and $d'$ | [15], [31] | III-F / V-F |
| product quantization | the joint effects of $d', m, k_s$ and large scale experiments | [21] | III-G / V-H |
| classification | improvements in large scale image classification | | -/V-I |

very large (e.g. larger than $2^{13}$) since this would prohibitively increase both the quantization cost and the memory required for storing the resulting product quantizer.

In Section V-H, we attempt to *shed more light into the joint effects* of $d'$, $m$ and $k_s$ for a fixed code size. Besides these parameters, we evaluate the *merits of applying a random orthogonal transformation on PCA-projected and whitened VLAD* vectors before proceeding with PQ. This transformation was shown to improve the search results when applied to PCA-projected (but not whitened) VLAD vectors in [19] as it manages to balance the energy of the subvectors. Finally, we conduct large-scale experiments on a dataset of 10M images and *compare PQ+ADC with the non-exhaustive PQ+IVFADC* search variant in terms of accuracy and efficiency.

## IV. Experimental Setup

This section describes our experimental setup. Subsection IV-A explains the evaluation protocol and presents the evaluation measures and the datasets used for learning and benchmarking while Subsection IV-B discusses details related to image processing. Table I serves are as a reference for the aspects and extensions studied in Section V.

### A. Evaluation Protocol and Datasets

Since the studied extensions and paremeter settings concern steps of a sequential processing pipeline (see Figure 1), it is expected that choices made on earlier steps of the pipeline (e.g. type of local features/feature filtering) may affect the settings and/or methods that lead to optimal performance in subsequent steps. Therefore, the full optimization of the pipeline requires exhaustive exploration of all different combinations of methods and parameters. To reduce the complexity of the analysis and keep the load of the experiments reasonable we make the following main relaxed assumptions:

(a) A more discriminative vectorized representation will lead to a more discriminative binary signature after the application of PQ, compared to a less discriminative representation of the same length. This reasonable assumption was also adopted (implicitly) in [1], [15] and allows us to exclude PQ from experiments on extensions and parameter settings that concern previous steps of the pipeline.

(b) When an extension or selection of parameters improves significantly the performance at an earlier step of the pipeline, we adopt this choice for subsequent steps. For instance, given the dominance (Subsection V-A) of SURF in terms of accuracy compared to other local features (CSURF is excluded due to increased computational complexity), we use VLAD+SURF for the subsequent sets of experiments. However, in cases where the performance differences are marginal we double check whether an increase in accuracy is propagated to subsequent steps (e.g. in Subsection V-B).

TABLE II: Datasets used in evaluation. #n denotes the number of images, #q denotes the number of images treated as queries.

| Name | Use | #n | #q | Source |
|---|---|---|---|---|
| Holidays | retrieval | 1491 | 500 | [12] |
| Oxford | retrieval | 5063 | 55 | [5] |
| Paris | retrieval | 6412 | 55 | [11] |
| UKB | retrieval | 10200 | 10200 | [3] |
| Flickr50K | distractors | 50K | - | [39] |
| ImageNET10M | distractors | 10M | - | [40] |
| Flickr100K | learning | 100K | - | [39] |

Experiments are conducted on the following four widely used benchmark collections for image retrieval:

**Holidays** [12] contains 500 groups of personal holiday photos as well as groups of photos taken to test the robustness of a representation to various transformations (rotations, viewpoint and illumination changes, blurring, etc.). One image in each group is treated as the query and the correct retrieval results are the other images of the group. The collection includes a large variety of scene types (natural, man-made, water and fire effects, etc.). Retrieval accuracy is measured in terms of mAP.

**Oxford** [5] and **Paris** [11] consist of images collected from Flickr by searching for particular Oxford and Paris landmark buildings, respectively. Both collections have been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. This gives a set of 55 queries over which an image retrieval system can be evaluated. Differently from Holidays, all images of these collections are in “upright” orientation because they are displayed on the web. Retrieval accuracy is measured by mAP in both datasets, treating each query image as not present in the database in the query that involves it[3].

**UKB** [3] is an image recognition benchmark that contains 10200 images of 2550 distinct objects (4 images per object). Performance is measured by querying the database once for each image and counting the average number of relevant images (including the query itself) ranked in the top 4 positions (4 × Recall@4).

[3] Note that the evaluation software provided with the Oxford dataset treats query images as positive examples, giving significantly higher mAP scores since the query image is always returned in the first position.

TABLE III: Retrieval performance of full as well as PCA-projected ($d'$=128) VLAD vectors using different types of features. The best result in each dataset is marked with an asterisk for full vectors and typeset in bold for PCA-projected vectors.

| **Dataset** | **VLAD+SIFT** | | **VLAD+RootSIFT** | | **VLAD+SURF** | | **VLAD+CSURF** | |
|---|---|---|---|---|---|---|---|---|
| | $d$ =8192 | $d'$ =128 | $d$ =8192 | $d'$ =128 | $d$ =4096 | $d'$ =128 | $d$ =12288 | $d'$ =128 |
| Holidays (mAP) | 0.561 | 0.534 | 0.589 | 0.547 | 0.649 | 0.638 | 0.717* | **0.697** |
| Oxford (mAP) | 0.243 | 0.131 | 0.242 | 0.124 | 0.328* | **0.238** | 0.256 | 0.167 |
| Paris (mAP) | 0.207 | 0.083 | 0.203 | 0.068 | 0.321* | **0.180** | 0.296 | 0.254 |
| UKB (4×R@4) | 2.794 | 2.896 | 2.896 | 2.849 | 3.196 | 3.237 | 3.520* | **3.482** |

To evaluate the accuracy at a more challenging retrieval scenario where more images can be confused with those in the set of relevant images for each query, we merge each collection with additional images that act as distractors. For this purpose, we use a subset of 50K images (*Flickr50K*) from the MIR Flickr 1M collection [39] for most experiments. To evaluate the accuracy and the efficiency of the framework on a very large scale, in Section III-G we use a larger set of 10M distractor images (*ImageNET10M*) that contains images downloaded from ImageNET [40] (fall 2011 release URLs). Finally, *Flickr100K*, another subset from the MIR Flickr 1M collection (disjoint from all previous datasets), is used for performing the various learning tasks (visual vocabularies, PCA matrices, product quantizers). We believe that the choice of an independent learning dataset, better reflects the accuracy of a real system where relevant images are only a small fraction of the overall image database. All datasets are listed in Table II.

### *B. Image Pre-processing & Feature Extraction*

All images used in the evaluation were first scaled to a maximum size of $512 \times 384$ pixels prior to feature extraction. This down-scaling amounts to an almost 4-fold size reduction for the images of the four benchmark collections, while most distractor images are already around this size. As shown in Subsection V-G, the use of larger images leads to significantly improved accuracy. However, larger images also increase feature extraction time. Furthermore, compared to other studies where collection images are usually larger than distractor images, we believe that using approximately equal sizes for collection and distractor images represents a more challenging evaluation scenario.

For feature extraction, we used the high-quality open-source implementations of SURF and SIFT provided in BoofCV[4]. For SIFT, BoofCV implements the feature detection and the description algorithms as described in [10] (with minor algorithmic changes) while for SURF it slightly deviates from the original algorithms implementing the SURF-S [41] version.

## V. Experimental Results

### *A. Comparison of Local Features*

Table III shows the results obtained on Holidays, Oxford, Paris and UKB with full and PCA-projected ($d'$=128) VLAD vectors generated using SIFT, RootSIFT, SURF and CSURF features. In all cases, a visual vocabulary of $k = 64$ visual words is used.

TABLE IV: Extraction times for different types of features on 512x384 images (results averaged over 100 images). Extraction was performed using one core of an i5 2.4 GHz processor.

| Feature: | SIFT/RootSIFT | SURF | CSURF |
|---|---|---|---|
| Time (ms): | 350.6 | 135.7 | 205.6 |

We observe that the VLAD+SURF combination significantly outperforms both VLAD+SIFT and VLAD+RootSIFT in all datasets, when full vectors are used. This is an interesting result since the main motivation for using SURF in place of SIFT was SURF's better extraction efficiency (see Table IV). Investigating this issue further, we noticed that the version of SURF implemented in BoofCV (SURF-S [41]) applies several algorithmic improvements on the original algorithm, resulting in improved stability and runtime performance. Additionally, the results show that, compared to VLAD+SIFT and VLAD+RootSIFT, VLAD+SURF retains a higher percentage of its initial accuracy when dimensionality reduction is applied. Being half-dimensional than SIFT ($D$=64 versus $D$=128), SURF result in more compact VLAD vectors that are more amenable to dimensionality reduction.

With respect to VLAD+CSURF, while it achieves the best overall performance on Holidays and UKB, it is outperformed by VLAD+SURF on Oxford and Paris. A closer examination of the images of the four datasets shows that the query images of Holidays and UKB exhibit a larger chromatic variability (i.e. each query is chromatically distinct) compared to the query images of Oxford and Paris. Therefore, exploiting color information is more useful in these datasets, while in Oxford and Paris shape is the most distinctive factor.

Overall, the results of this subsection suggest that *SURF constitutes an excellent replacement for SIFT and RootSIFT* in the context of VLAD since it leads to consistently increased search accuracy and at the same time it can be extracted much faster. In cases where color is expected to be a discriminative factor *CSURF constitutes an even better alternative*, at the cost of increased extraction time and a larger representation size. Since we focus on very efficient settings, we employ SURF for the rest of the experiments.

### *B. The Effect of Feature Filtering Methods*

*1) Filtering based on the richness of feature structure:* In this experiment we study the effect of the entropy-based and variance-based filtering methods presented in Section III-B1. Since SURF take values in a continuous range, in order to calculate entropy we first discretize them by applying equal width binning separately on each component of the descriptor.

[4] http://boofcv.org

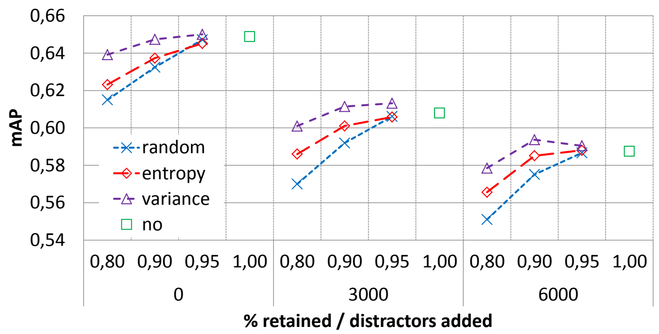


Fig. 2: Filtering results using full VLAD vectors on Holidays.

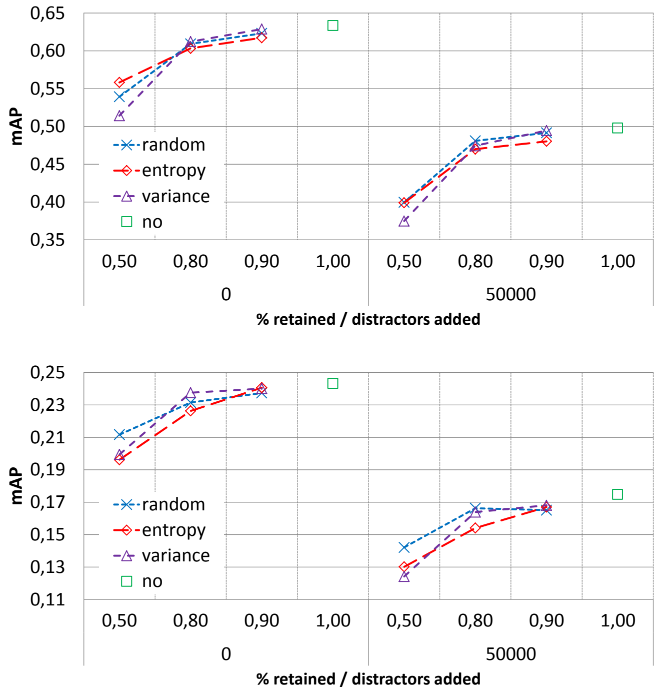


Fig. 3: Filtering results using PCA-projected ($d'$=128) VLAD vectors on Holidays (top) and Oxford (bottom).

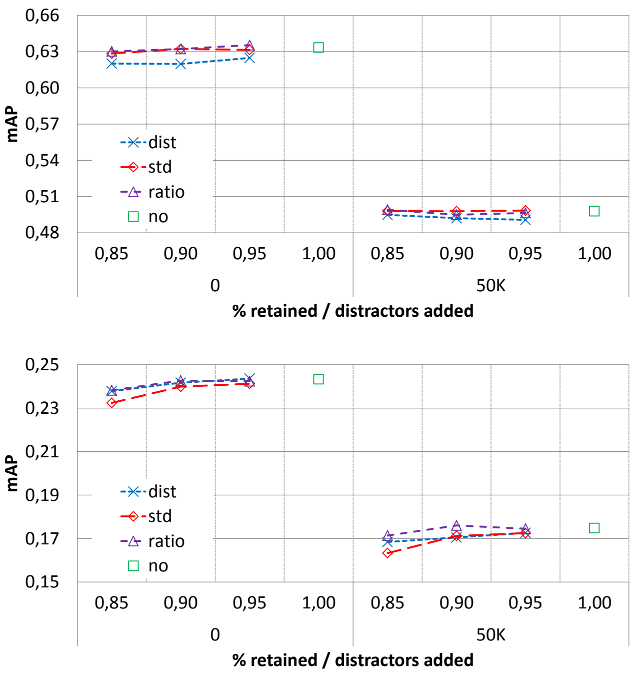


Fig. 4: Outlier rejection results using PCA-projected ($d'$=128) VLAD vectors on Holidays (top) and Oxford (bottom).

We use 128 bins whose ranges are computed on a set of 200K SURF features (different numbers of bins have also been evaluated with similar results). Using images from Flickr100K, we compute appropriate thresholds values for both entropy and variance so that approximately 20%, 10% and 5% of the features are discarded. We also use random filtering to discard the same percentages of features. Figure 2 shows mAP results on Holidays when 4096-dimensional ($k = 64$) VLAD vectors are used, for different percentages of features retained and with different levels of distractors. We observe that all filtering methods operate worse as the percentage of filtered features increases, with the exception of variance and 6000 distractors where there is a slight increase in performance when retaining 90% of the features (compared to 95%). Both entropy-based and variance-based filtering give better results than random filtering on all operating points, indicating that *both criteria are good at detecting less-discriminative features*. Furthermore, the proposed variance-based filtering always outperforms entropy-based filtering suggesting that *variance is indeed a better filtering criterion than entropy*. Comparing the results with those obtained without filtering, we observe that all filtering methods perform worse than no filtering with zero additional distractors. However, as the number of distractors increases, variance-based filtering (discarding 5% or 10% of the features) gives the best results.

Figure 3 shows mAP results on Holidays and Oxford when PCA-projected ($d'$=128) VLAD vectors are used, for different percentages of features retained. In this case, all filtering methods give similar results that are worse than no filtering. This can be attributed to the fact that features with poor internal structure are always assigned to a subset of visual words that fail to explain the variability in the data and as a result their influence is discounted after the application of PCA. Since the use of low-dimensional, PCA-projected VLAD vectors is essential for good accuracy of very efficient PQ schemes (as we show in Section V-H), we conclude that *entropy and variance-based filtering are not appropriate for large-scale retrieval* and are therefore not considered for the rest of the experiments.

2) *Filtering based on a feature-vocabulary relation:* In this experiment we evaluate the outlier feature rejection methods described in Section III-B2. For the *dist* method, [26] reports that $C$=90 was experimentally found to give the best results. Here, we additionally report results using $C$=85 and $C$=95. The percentile values for each visual word are computed on Flickr100K. Similarly, the $a$ and $b$ parameters of the *std* and *ratio* methods are tuned to reject approximately 5%, 10% and 15% of the features (this amounts to to $a$=1.39, 1.36 and 1.34 and $b$=0.98, 0.99 and 0.995 respectively).

Figure 4 shows mAP results on Holidays and Oxford when

TABLE V: Sum vs. mean aggregation (mAP).

| **Dataset** | **Dimensionality**<br>128 | | 1024 | | full | |
|---|---|---|---|---|---|---|
| | sum | mean | sum | mean | sum | mean |
| Holidays | **0.633** | 0.554 | **0.679** | 0.598 | **0.649** | 0.601 |
| Holidays+50K | **0.498** | 0.396 | **0.567** | 0.474 | **0.522** | 0.475 |
| Oxford | **0.243** | 0.191 | **0.310** | 0.228 | **0.327** | 0.246 |
| Oxford+50K | **0.175** | 0.096 | **0.230** | 0.115 | **0.226** | 0.128 |
| Paris | **0.290** | 0.201 | **0.327** | 0.234 | **0.350** | 0.272 |
| Paris+50K | **0.164** | 0.085 | **0.196** | 0.104 | **0.209** | 0.116 |

PCA-projected ($d'$=128) VLAD vectors are used, for different percentages of features retained, with zero and 50K additional distractors. In contrast to the results of [26], we see that the *dist* method performs worse than no filtering for all $C$ values on both datasets. The same holds for *std* and *ratio*. A possible explanation for the different results compared to [26] is that in our experiment, we use an independent dataset for learning the percentiles (this represents a more realistic scenario). Also the evaluation in [26] was performed using an image-level ROC curve analysis, a non-standard method. Overall, our evaluation suggests that *filtering outlier features does not have an impact on retrieval accuracy*.

### C. Sum versus Mean Aggregation

In this experiment we compare the performance of mean-aggregated VLAD vectors with the performance of the original sum-aggregated VLAD. Table V shows the results obtained on Holidays, Oxford and Paris (with zero and 50K additional distractors) using both PCA-projected and full-dimensional vectors. Our results contradict those presented in [27]. We see that in all cases, *the originally proposed VLAD method outperforms the extension that uses mean aggregation*. These results are in agreement with the discussion of Section III-C, where we argued that by normalizing the sum, the mean aggregation formula discards information about the number of features assigned to a similar position in the Voronoi cell.

### D. Vocabulary Size and PCA

In this experiment we study the impact of vocabulary size on the quality of the VLAD+SURF representation. In order to keep the assignment cost low and to reduce the impact of dimensionality reduction in accuracy (as explained in Section III-D), we limit our analysis on vocabulary sizes up to $k$=512 centroids.

Figure 5 shows the retrieval performance on Holidays using VLAD+SURF vectors of varying vocabulary size. As expected, accuracy increases with vocabulary size. Interestingly, using VLAD+SURF vectors and $k$=512 we obtain a mAP score of 68.8% that is similar to the one obtained in [1] where a significantly more expensive setting was used: FV+PCA-SIFT with $k$=4096.

Figure 6 shows mAP results obtained on Holidays and Oxford using PCA-projected VLAD+SURF vectors produced from vocabularies of different sizes. We report results for vectors with up to 1024 dimensions that are more suitable for large-scale retrieval. The results confirm that larger vocabulary sizes suffer more from dimensionality reduction. The largest vocabulary ($k$=512), gives worse results than most of the smaller vocabularies for all projection lengths on both datasets. The best results are obtained using $k$=128 (followed closely by $k$=256) for all projection lengths. Interestingly, with $d'$=1024 a 68.7% mAP is obtained on Holidays that is similar to the best score obtained using full dimensional VLAD+SURF vectors ($k$=512 and $d$=32768). Also, using only $d' = 128$ dimensions, a 63.7% mAP is obtained that is 12.7% higher than the best mAP (56.5%) reported in [1] for the same dimensionality. Note that with this size, 1M vectors can fit in 1GB of main memory.

By comparing Figures 5 and 6 (top) we see that for vocabulary sizes smaller than $k = 256$ and a small dimensionality reduction we achieve better accuracy than using full vectors produced from the same vocabulary. This phenomenon was also observed in [1], but it was not explained. Here, we *provide a justification* of this increase in performance, extending the analysis presented in [15] for BoW vectors. Similarly to BoW,

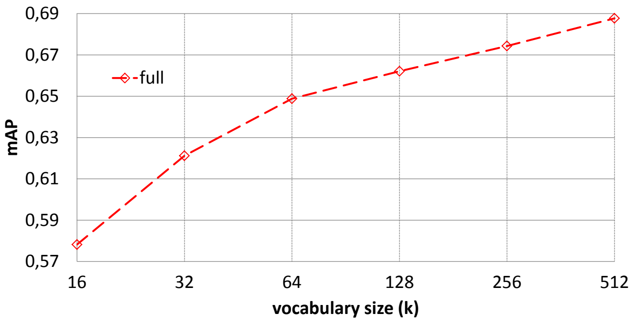


Fig. 5: VLAD+SURF vectors of varying vocabulary size, mAP on Holidays.

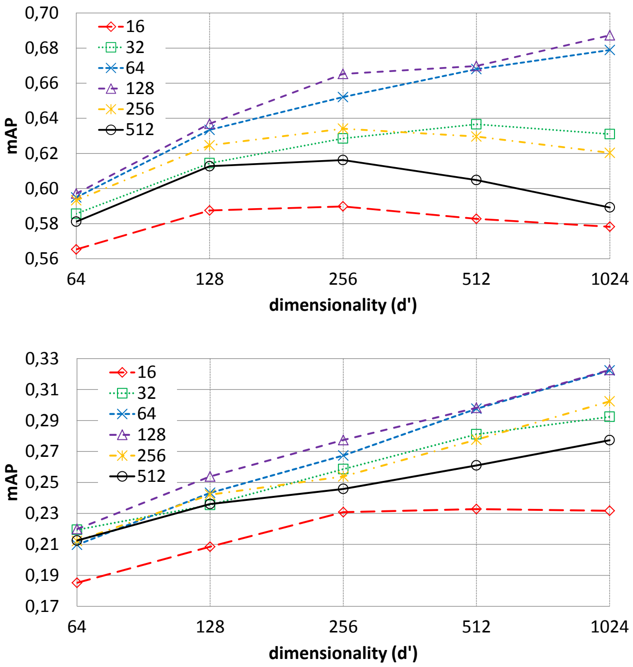


Fig. 6: PCA-projected VLAD+SURF vectors produced from vocabularies of different sizes, mAP on Holidays (top), mAP on Oxford (bottom).

VLAD are usually compared using cosine similarity that is equivalent to the inner product when the vectors are L2 normalized. In case that no features are assigned to a particular visual word $c_i$, the corresponding vector of aggregated differences $v_i$ will be the vector of all zeros $\vec{0}$. Given two images $a$ and $b$ and the corresponding VLAD vectors $v^a$ and $v^b$, and assuming that $v_i^a = \vec{0}$, the contribution of visual word $c_i$ to the cosine similarity of $v^a$ and $v^b$ will be the same when either $v_i^b = \vec{0}$ or $v_i^b \neq \vec{0}$. This way, the information that two visual words are jointly missing from two images is not taken into account although it can be a strong indication of similarity, especially for small vocabularies. [15] showed that by subtracting the mean BoW vector (calculated on a learning set) from the original BoW vector, the similarity measure is improved. By applying PCA this centering is performed implicitly. This explains the observed increase in accuracy (compared to using full vectors) for smaller vocabularies and a limited reduction. For larger vocabularies and/or heavier reductions, *the positive effect of centering is cancelled by the large projection error.*

### E. The Effect of Whitening

In this experiment we study the effect of whitening on VLAD vectors. Figure 7 shows results for Holidays and Oxford [5]. On Holidays, we see that whitening improves the accuracy of 128-dimensional vectors for most vocabulary sizes. The best mAP result for 128-dimensional vectors without whitening is 63.7% while with whitening we obtain 65.7%, a 3% relative improvement. The situation is similar on Oxford where for 128-dimensional vectors, whitening improves the results for all vocabulary sizes. Here, the relative improvement is even larger (18%), as the previous best mAP result of 25.4% increases to 30.0%. The picture is different for 1024-dimensional vectors, where, while a significant improvement is observed on Oxford for all vocabulary sizes, on Holidays, the best accuracy is obtained without whitening. This difference is probably due to the abundance of visual word co-occurrences (the problem that whitening tries to address) on Oxford [38].

As a general trend, we observe that in both datasets whitening has a more positive impact for 128-dimensional vectors and for larger vocabularies ($k$=256 and $k$=512). A related observation was made in [15], where it was suggested that for large projection lengths, whitening may have a negative impact because it magnifies the noise of the low-energy components. For VLAD+SURF vectors generated from a $k$=512 vocabulary ($d$=32768), $d'$=1024 is still a small projection length. In conclusion, we suggest that *whitening should always be performed jointly with PCA on VLAD* vectors when we are interested in low-dimensional representations.

### F. Multiple Vocabularies

In this experiment we evaluate the multiple vocabulary aggregation method described in Section III-F. As in Section V-D we consider vocabularies with a maximum total number of $k$=512 visual words to keep the assignment complexity reasonable. Table VI lists the evaluated multiple vocabulary setups. In order to obtain a different learning set for each vocabulary in a multiple vocabulary setup, we take different random samples of 100K features from a set of approximately 70M SURF features extracted from Flickr100K.

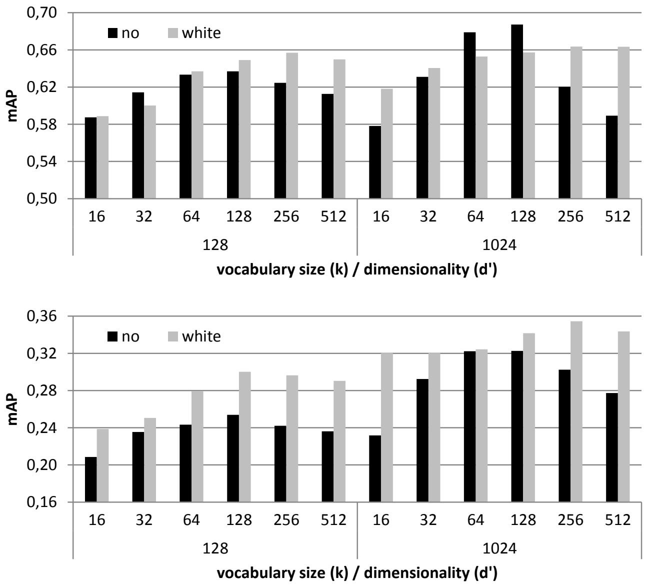


Fig. 7: Retrieval performance (mAP) obtained with 128-dimensional (left) vs. 1024-dimensional (right) VLAD vectors produced with vocabularies of different sizes, with and without whitening on Holidays (top) and Oxford (bottom).

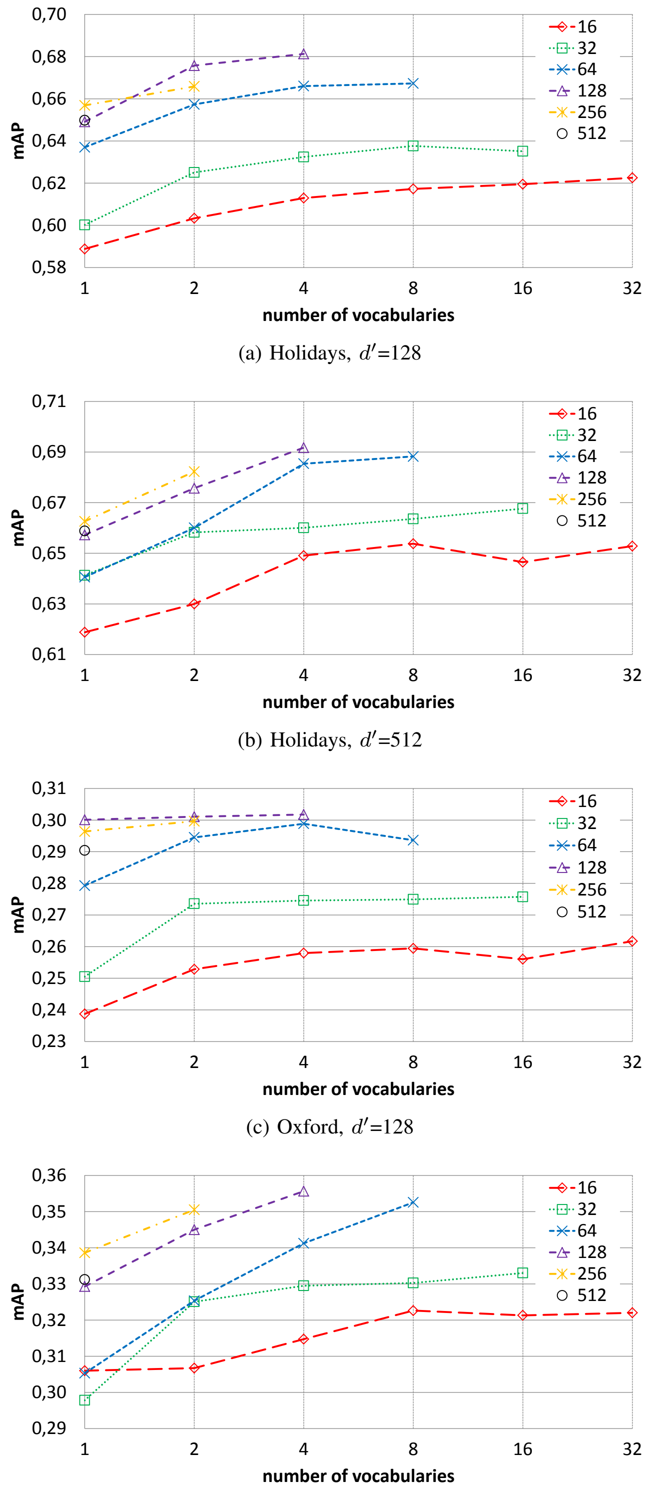
TABLE VI: Multiple vocabulary setups.

| Total complexity | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|
| Setups | 2x16 | 4x16<br>2x32 | 8x16<br>4x32<br>2x64 | 16x16<br>8x32<br>4x64<br>2x128 | 32x16<br>16x32<br>8x64<br>4x128<br>2x256 |

Figures 8a and 8b show results on Holidays using a single and multiple vocabularies of different sizes to generate 128- and 512-dimensional VLAD vectors respectively. Figures 8c and 8d show the corresponding results on Oxford[6]. The best results for both datasets and projection lengths are obtained using multiple vocabulary aggregation. Specifically, the 4x128 setup is the top performer in all cases, increasing the performance of the best single vocabulary setup by 3.7% (0.5%) for 128-dimensional vectors and by 4.4% (5.0%) for 512-dimensional vectors on Holidays (Oxford). However, we can see that extreme setups (i.e. many small vocabularies) perform worse than more conservative setups, even worse than single vocabulary setups of the same total complexity. This performance trend is different to the one observed in [15] with BoW vectors where performance improved with increasing number of vocabularies (of a fixed total complexity). Finally, we observe that the relative improvements are larger for 512-

[5] Results for additional projection lengths and under the presence of additional distractor images are provided in the supplementary material.

[6] Results for additional projection lengths are provided in the supplementary material.



(a) Holidays, $d'$=128

(b) Holidays, $d'$=512

(c) Oxford, $d'$=128

(d) Oxford, $d'$=512

Fig. 8: Multiple vocabularies experiment on Holidays and Oxford with vectors projected to 128 and 512 dimensions.

dimensional vectors compared to 128-dimensional in both datasets. Comparing these results with the results of Figure 5 where full vectors are used, we observe that using only 128-dimensional vectors, we achieve a similar mAP (69%) as with 32768 dimensional full VLAD vectors. Also this result is significantly better than the 61.4% mAP reported in [15] for 128-dimensional VLAD vectors coming from a $k$ =4x256 multiple vocabulary setup.

TABLE VII: Comparison against the state-of-the-art on short vectors. Accuracy is measured by mAP on Holidays and Oxford and by 4×R@4 on UKB. Results with an asterisk on Oxford indicate that Paris was used for learning instead of an independent dataset.

| | Representation | $k$ | Holidays | Oxford | UKB |
|---|---|---|---|---|---|
| 128d vectors | BoW+PCASIFT [1] | 20K | 45.2 | 15.9 | 2.95 |
| | VLAD+PCASIFT [1] | 64 | 55.7 | 25.7 | 3.35 |
| | FV+PCASIFT [1] | 64 | 56.5 | 24.3 | 3.33 |
| | mBOW+SIFT [15] | 4x8K | 56.7 | 41.3* | 3.19 |
| | mVLAD+SIFT [15] | 4x256 | 61.4 | | 3.36 |
| | VLAD$_{ai}$+ROOTSIFT [30] | 256 | 62.5 | **44.8*** | |
| | VLAD*+ROOTSIFT [31] | 64 | | 32.5* | |
| | VLAD+CSURF | 64 | **73.8** | 29.3 | **3.50** |
| | mVLAD+SURF | 4x128 | 71.8 | **38.7** | 3.32 |

## G. Comparison with the state-of-the-art on short vectors

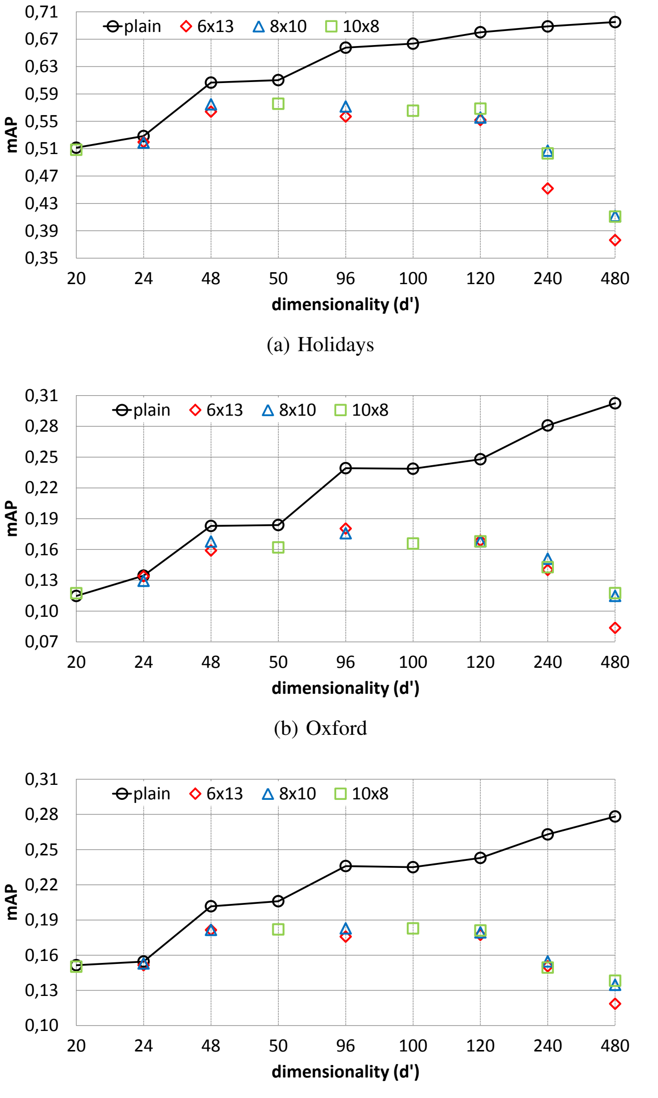
In this section we compare our optimized VLAD representation with reference state-of-the-art results, after modifying the experimental setup that we used in previous subsections to match the setup most commonly used in similar studies. Specifically, we: a) use larger, 1024x768 dimensional images for all datasets and b) calculate mAP on Oxford using the provided evaluation software which treats query images as positive examples. Table VII shows results on Holidays, Oxford and UKB, obtained using 128-dimensional vectors of various types. The last 2 rows correspond to our PCA-projected and whitened VLAD+CSURF and multiple vocabulary aggregated VLAD+SURF vectors (mVLAD+SURF).

On Holidays we improve the previous best reported result [30] by 18% with VLAD+CSURF and by 15% with mVLAD+SURF. On Oxford, our mVLAD+SURF and VLAD+SURF representations obtain mAP scores that are 59% and 21% better than the best result reported in [1] using an independent learning dataset (better results have been reported in [15] and [30] but using Paris for learning). Finally, VLAD+CSURF improves the state-of-the-art [15] by 4% on UKB while mVLAD+SURF obtains slightly lower results.

## H. Product Quantization and Large-Scale Experiments

In this experiment we study the parameters of PQ discussed in Section III-G. PQ is applied on PCA-projected and whitened VLAD vectors generated using four vocabularies of size $k$=128 that were found to give the best results in Section V-F. To study the joint effects of $d'$, $m$ and $k_s$ we use three $m \times k_s$ schemes (6x13, 8x10, 10x8) that allocate approximately the same number of bits (78 and 80), and for each scheme we evaluate the performance using six different projection lengths $d' = \{20/24, 48/50, 96/100, 120, 240, 480\}$[7]. Note that with such a small memory footprint, up to 100M vectors can be stored in 1GB of memory. Figures 9a, 9b and 9c

[7] The alternative $d'$ values (e.g. 20/24) are used because $d'$ should be an exact multiple of $m$.



(a) Holidays

(b) Oxford

(c) Paris

Fig. 9: mAP results using 6x13, 8x10 and 10x8 PQ schemes and uncompressed PCA-projected VLAD vectors, for dimensionality reduction to varying dimensions $d'$.

show mAP results on Holidays, Oxford and Paris respectively. In all datasets, we observe a great variation with respect to $d'$, with the best results obtained with $d'$ between 48 and 100. With such a small quantization code, larger dimensional vectors incur a significant quantization loss. With respect to the parameters $m$ and $k_s$, we observe that *all quantization schemes exhibit similar accuracy near the optimal* $d'$. These results are *different from [21]* (where for a fixed code size quantization schemes with smaller values for $m$ were found better) and *in favor of schemes with smaller* $k_s$ values due to the fact that they are more efficient. Furthermore, experimental results provided in the supplementary material, suggest that the random transformation step is helpful only for projection lengths $d' > 100$ while for smaller $d'$ the *results are similar to not applying random transformation.*

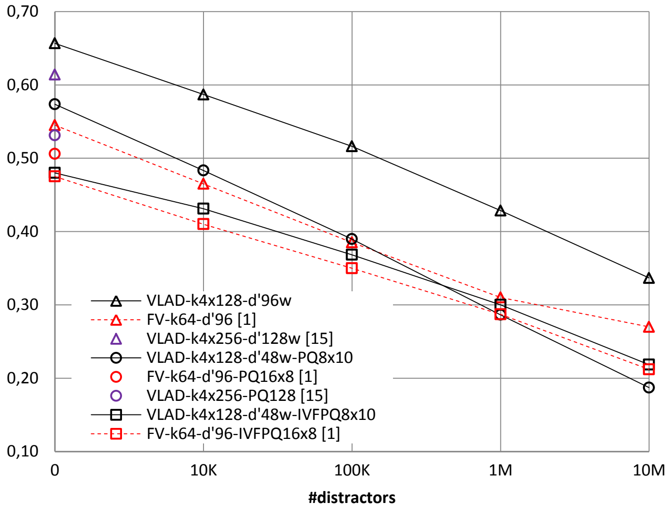


Fig. 10: mAP results on Holidays + ImageNET10M

*Large-scale Experiments on Holidays+ImageNET10M*: Figure 10 shows the performance of various large-scale systems on datasets of increasing size (10K, 100K, 1M, 10M) constructed by combining the images of Holidays and ImageNET10M. We report results for the following systems:

1) VLAD-$k$4x128-$d'$96w: our VLAD+SURF with $k$=4x128, reduced to $d'$=96 with PCA+whitening.
2) FV-$k$64-$d'$96: FV+PCA-SIFT with $k$=64, reduced to $d'$=96 with PCA, results from [1].
3) VLAD-$k$4x256-$d'$128w: VLAD+SIFT with $k$=4x256, reduced to $d'$=128 with PCA+whitening, results from [15].
4) VLAD-$k$4x128-$d'$48w-PQ8x10: our VLAD+SURF with $k$=4x128, reduced to $d'$=48 with PCA+whitening and encoded to 80 bits using the 8x10 PQ scheme.
5) FV-$k$64-$d'$96-PQ16x8: (2) encoded to 128 bits using the 16x8 PQ scheme, results from [1].
6) VLAD-$k$4x256-PQ128: (3) encoded to 128 bits using PQ ($d'$ and PQ scheme not given), results from [15].
7) VLAD-$k$4x128-$d'$48w-IVFPQ8x10: (4) combined with inverted lists (IVFADC $w$=64/8192).
8) FV-$k$64-$d'$96-IVFPQ16x8: (5) combined with inverted lists (IVFADC $w$=64/8192), results from [1].

Looking at the results obtained using uncompressed VLAD vectors (triangles), we observe that our 96-dimensional representation is significantly better than both the 128-dimensional representation of [15] and the 96-dimensional representation of [1] for all distractor levels. Looking at the results using PQ encoded vectors and ADC search (circles), we observe that we outperform other state-of-the-art systems that use 128 bits per image by using only 80 bits. When no additional distractors are present, our VLAD-$k$4x128-$d'$48w-PQ8x10 system obtains a 57.6% mAP while FV-$k$64-$d'$96-PQ16x8 and VLAD-$k$4x256-PQ128 obtain 50.6% and 53.1%, respectively. Looking at the results using PQ encoded vectors and IVFADC search (squares), we observe that our VLAD-$k$4x128-$d'$48w-IVFPQ8x10 system, which uses only 80 bits per image, obtains slightly better results than FV-$k$64-$d'$96-IVFPQ16x8 which uses 128 bits per image. Interestingly, we notice that *PQ+ADC schemes outperform PQ+IVFADC schemes for up to 100K distractors*. This suggests that *PQ+ADC should be*

TABLE VIII: Classification performance (mAP) of different image representations.

| **Method** | $d$ | $d'$ | mAP |
|---|---|---|---|
| **BOW** | 500 | 500 | 0.111 |
| **VLAD** | 32768 | 128 | 0.192 |
| **VLAD+** | 32768 | 128 | 0.233 |
| **cVLAD+** | 24576 | 128 | **0.259** |

*preferred over PQ+IVFADC for small-medium databases since it also has comparable or better efficiency* as discussed below.

**Timings**: With our single core implementation, searching the 10M dataset using 96-dimensional vectors takes (on average): 3.7 s when no encoding is used, 744 ms with the PQ+ADC 8x10 scheme and only 24 ms with the non-exhaustive PQ+IVFADC 8x10, 64/8192 scheme. For smaller datasets (up to 100K with our setup) ADC is slightly faster than IVFADC. We observed that the main overhead of IVFADC for small databases is the calculation of multiple ($w$) lookup tables (that negates the benefit of scanning a small subset of the database) and not the assignment to the coarse quantizer as mentioned in [21].

### I. Classification Experiments

Although our primary focus is on image representations for example-based large-scale retrieval, we expect that the improvements reported so far for VLAD can be of interest for large-scale image classification as well. To evaluate our hypothesis, we conducted classification experiments on the NUS-WIDE [42] dataset. NUS-wide is among the largest benchmark datasets for image classification containing 260K images from Flickr with manual ground truth annotations for 81 concepts. An interesting property of the dataset is that each image can be annotated with multiple concepts rendering the problem into a multi-label classification one. For all images we extract the following three types of vectors:

1) **VLAD**: original VLAD vectors (as described in [1]) using SIFT features and a single $k$=256 visual vocabulary, reduced to $d'$=128 with PCA.
2) **VLAD+**: improved VLAD vectors (as described in this paper) using SURF features and multiple $k$=4x128 visual vocabularies, reduced to $d'$=128 with PCA+whitening.
3) **cVLAD+**: improved VLAD vectors (as described in this paper) using CSURF features and a $k$=128 visual vocabulary, reduced to $d'$=128 with PCA+whitening.

Additionally, we use baseline Bag-of-Words (**BoW**) features provided along with NUS-WIDE (see [42] for details). Prior to learning, all types of vectors (including BoW) are normalized to unit length. For multi-label classification we apply the One-vs-All approach (implementation from Mulan [43]) coupled with Logistic Regression (implementation from LibLINEAR [44]). Performance for each dataset is measured in terms of mAP, using the original train-test splits. The results are reported in Table VIII. All VLAD based representations *perform impressively better* that the **BoW** baseline. We also see that **VLAD+** achieves a large 21.4% increase over standard **VLAD**. Finally, using the new CSURF descriptor, **cVLAD+** achieves the best overall performance that is 233% better than the performance of **BoW**.

## VI. Conclusions

Through a comprehensive study of the VLAD+PQ framework, we constructed 128-dimensional vectors that obtain significantly better accuracy than the state-of-the-art on three popular image retrieval benchmarks and 80-bit compressed image signatures that outperform less efficient setups on the Holidays benchmark. Furthermore, we showed that the proposed modifications lead to significant improvements on the image classification domain. Aside the above performance improvements, we believe that the extensive experimental study presented here and the accompanying experimental testbed offer valuable insights to image retrieval researchers and practitioners on the role of different feature extraction, aggregation and indexing steps involved in the VLAD+PQ framework. More specifically, the following practical conclusions are drawn from our empirical study:

- SURF constitutes an excellent replacement for SIFT (and RootSIFT) in the context of VLAD since it leads to consistently increased search accuracy and at the same time it can be extracted much faster. In cases where color is expected to be a discriminative factor CSURF constitutes an even better alternative.
- Feature filtering techniques based on either the richness of feature structure or the feature-vocabulary relation do not improve the VLAD representation.
- The originally proposed sum-aggregated VLAD outperforms the mean-aggregated extension.
- Whitening should always be performed jointly with PCA on VLAD vectors when we are interested in low-dimensional representations.
- When appropriate multiple vocabulary setups are used, the multiple vocabulary aggregation technique can offer significant improvements over using a single vocabulary.
- When aggressive compression is applied, the selection of PQ parameters $m$ and $k_s$ (for a constant number of bits) has negligible impact on accuracy. Thus, schemes with smaller $k_s$ values should be preferred due to being more efficient. On the other hand, there is great variation in accuracy with respect to $d'$ and thus its value should be carefully selected.
- PQ+IVFADC should be preferred over PQ+ADC for datasets larger than 100K images as it is more accurate and has better efficiency.

## Appendix
## A Graphical Illustration of Sum Versus Mean VLAD Aggregation

Figure 11 shows (on the left) a visual vocabulary with three visual words and the quantized (2-dimensional) features of two hypothetical images. On the right we see a graphical illustration of the VLAD signatures of the images using sum (right-top) and mean (right-bottom) aggregation. The aggregated residual for each visual word $c_i$ is depicted with an arrow $v_i$ starting from the origin. The distance between

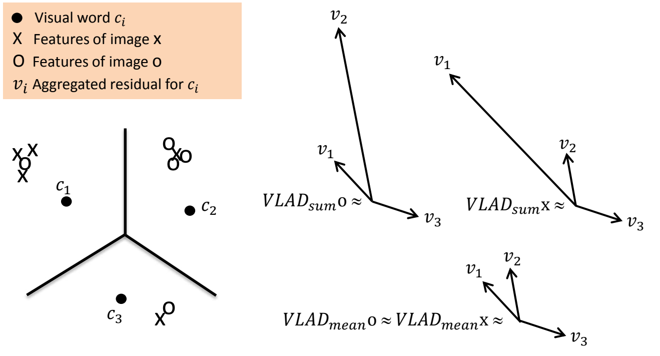


Fig. 11: Graphical illustration of sum versus mean aggregation.

two VLAD signatures depicted in this way is the sum of the Euclidean distances between the corresponding aggregated residuals. Looking at the quantized features of each image we observe that the two images differ significantly since the majority of the features of image x are quantized in $c_1$ while the majority of the features of image o are quantized in $c_2$. While this difference is captured by the sum-aggregated VLAD signatures, the mean-aggregated VLAD signatures of the images are identical due to the fact that mean-aggregation discards information about the number of features quantized in each visual word.

## References

[1] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[2] W. Dong, Z. Wang, M. Charikar, and K. Li, "High-confidence near-duplicate image detection," in *ICMR*, 2012.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR*, vol. 2, 2006, pp. 2161–2168.

[4] 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*, 2007, pp. 1–8.

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

[6] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *ACM International Conference on Image and Video Retrieval*, 2007.

[7] L. S. Kennedy and M. Naaman, "Generating diverse and representative image search results for landmarks," in *International Conference on World Wide Web*, 2008, pp. 297–306.

[8] L. Xie, A. Natsev, J. R. Kender, M. Hill, and J. R. Smith, "Visual memes in social media: tracking real-world news in youtube videos," in *in ACM International Conference on Multimedia*, 2011.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.

[11] 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.

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

[13] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proceedings of the International Conference on Multimedia*. ACM, 2010.

[14] L. Chu, S. Jiang, S. Wang, Y. Zhang, and Q. Huang, "Robust spatial consistency graph model for partial duplicate image retrieval," *Multimedia, IEEE Transactions on*, vol. 15, no. 8, pp. 1982–1996, Dec 2013.

[15] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening," in *ECCV*, 2012.

[16] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," in *Proceedings of the British Machine Vision Conference*, vol. 3, 2008, p. 4.

[17] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *CVPR*, 2009.

[18] H. Jégou, M. Douze, and C. Schmid, "Packing bag-of-features," in *ICCV*, 2009, pp. 2357–2364.

[19] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *CVPR*, 2010.

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

[21] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[22] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS*, 2008.

[23] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications (VISSAPP)*, 2009.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, 2008.

[25] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.

[26] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, H. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vectors for on-device image matching," in *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2011.

[27] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual enhanced visual vector as a compact signature for mobile visual search," *Signal Processing*, 2012.

[28] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[29] D. Picard and P.-H. Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 669–672.

[30] R. Arandjelović and A. Zisserman, "All about VLAD," in *CVPR*, 2013.

[31] J. Delhumeau, P.-H. Gosselin, H. Jégou, and P. Pérez, "Revisiting the VLAD image representation," in *ACM Multimedia*, 2013.

[32] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, 2004.

[33] E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "An empirical study on the combination of surf features with vlad vectors for image search," in *WIAMIS*, 2012.

[34] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[35] L. Li, S. Jiang, and Q. Huang, "Multi-description of local interest point for partial-duplicate image retrieval," in *IEEE International Conference on Image Processing*, 2010.

[36] P. Fan, A. Men, M. Chen, and B. Yang, "Color-surf: A surf descriptor with local kernel color histograms," in *IEEE International Conference on Network Infrastructure and Digital Content*, 2009.

[37] H. Cai, X. Wang, and Y. Wang, "Compact and robust fisher descriptors for large-scale image retrieval," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2011, pp. 1–6.

[38] O. Chum and J. Matas, "Unsupervised discovery of co-occurrence in sparse high dimensional data," in *CVPR*, 2010.

[39] B. T. Mark J. Huiskes and M. S. Lew, "New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative," in *ACM International Conference on Multimedia Information Retrieval*, 2010.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[41] P. Abeles, "Speeding up surf," in *9th International Symposium on Visual Computing*. Springer, 2013, pp. 454–464.

[42] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., 2009.

[43] G. Tsoumakas, E. S. Xioufis, J. Vilcek, and I. P. Vlahavas, "Mulan: A java library for multi-label learning." *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 2411–2414, 2011.

[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

**Eleftherios Spyromitros-Xioufis** was both in Thessaloniki, Greece, in 1985. He received the B.S. degree in computer science in 2009 and the M.S. in information systems in 2011 from the Aristotle University of Thessaloniki (AUTH), Greece. He is currently pursuing the Ph.D. degree in computer science at the same University as a member of the Machine Learning and Knowledge Discovery group.

From 2012 to 2013, he was a Research Assistant with the Information Technologies Institute / Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include multi-target prediction, content-based multimedia annotation and retrieval, classification of evolving data streams and recommender systems.

Mr. Spyromitros-Xioufis is member of the Hellenic Artificial Intelligence Society since 2011. In 2012 he received a Ph.D. Excellence scholarship from the Research Committee of AUTH and a Research Excellence award from the Greek Ministry of Education for his achievements in international machine learning and data mining contests including a 1st place in the music instruments track of the ISMIS 2011 contest on music information retrieval and a 2nd place in the cold-start recommendations task of the ECMLPKDD 2011 discovery challenge.

**Symeon Papadopoulos** received the Diploma degree in Electrical and Computer Engineering in the Aristotle University of Thessaloniki (AUTH), Greece in 2004. In 2006, he received the Professional Doctorate in Engineering (P.D.Eng.) from the Technical University of Eindhoven, the Netherlands. Since September 2006, he has been working as a research associate with the Information Technologies Institute (ITI), part of the Centre for Research and Technology Hellas (CERTH), on a wide range of research areas such as information search and retrieval, social network analysis, data mining and web multimedia knowledge discovery. In 2009, he completed a distance-learning MBA degree in the Blekinge Institute of Technology, Sweden. In 2012, he defended his Ph.D. thesis in the Informatics department of AUTH on the topic of large-scale knowledge discovery from social multimedia. He is currently Chair of the IEEE Special Technical Community on Social Networking (STCSN).

**Ioannis (Yiannis) Kompatsiaris** is a Senior Researcher (Researcher A') with the Information Technologies Institute / Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include semantic multimedia analysis, indexing and retrieval, social media and big data analysis, knowledge structures, reasoning and personalization for multimedia applications, eHealth and environmental applications. He received his Ph.D. degree in 3-D model based image sequence coding from the Aristotle University of Thessaloniki in 2001. He is the co-author of 69 papers in refereed journals, 35 book chapters, 8 patents and more than 240 papers in international conferences. He has been the co-organizer of various international conferences and workshops and has served as a regular reviewer for a number of journals and conferences. He is a member of ACM.

**Grigorios Tsoumakas** was born in Thessaloniki, Greece, in 1977. He received a degree in computer science from the Aristotle University of Thessaloniki (AUTH), Greece, in 1999, an MSc in artificial intelligence from the University of Edinburgh, United Kingdom, in 2000 and a PhD in computer science from AUTH in 2005. Since 2013 he is an Assistant Professor with the Department of Informatics of AUTH, where he also served as a Lecturer from 2007 to 2013. He has authored more than 70 articles, which have received more than 3000 citations (Google Scholar). His h-index is 18. His research interests include various aspects of machine learning, knowledge discovery and data mining, including ensemble methods, distributed data mining, text classification and multi-target prediction. Dr. Tsoumakas has been a member of the ACM since 2000. He has achieved top positions in several international machine learning and data mining competitions, including 1st place in the BioASQ 2013 semantic indexing challenge, 1st place in the concept-based retrieval sub-task of the photo-annotation task of ImageCLEF 2011 and 2nd place in the music instruments track of the ISMIS 2011 music information retrieval contest.

**Ioannis Vlahavas** is a Professor at the Department of Informatics at the Aristotle University of Thessaloniki. He received his Ph.D. degree in Logic Programming Systems from the same University in 1988. He specializes in knowledge based and AI systems, Machine Learning, and he has published over 260 papers and book chapters, and co-authored 9 books in these areas. Google scholar gives a number of 3880 citations and an h-index of 30. He teaches AI, Machine Learning, and DSS. He has been involved in more than 30 research and development projects, leading most of them. He has served as general chair or co-chair or workshop chair in many conferences. He is leading the Logic Programming and Intelligent Systems Group (LPIS Group, lpis.csd.auth.gr). http://plase.csd.auth.gr/vlahavas