# Voronoi-based Compact Image Descriptors: Efficient Region-of-Interest Retrieval With VLAD And Deep-Learning–based Descriptors

Aaron Chadha and Yiannis Andreopoulos

***Abstract*—We investigate the problem of image retrieval based on visual queries when the latter comprise arbitrary regions-of-interest (ROI) rather than entire images. Our proposal is a compact image descriptor that combines the state-of-the-art in content-based descriptor extraction with a multi-level, Voronoi-based spatial partitioning of each dataset image. The proposed multi-level Voronoi-based encoding uses a spatial hierarchical K-means over interest-point locations, and computes a content-based descriptor over each cell. In order to reduce the matching complexity with minimal or no sacrifice in retrieval performance: *(i)* we utilize the tree structure of the spatial hierarchical K-means to perform a top-to-bottom pruning for local similarity maxima; *(ii)* we propose a new image similarity score that combines relevant information from all partition levels into a single measure for similarity; *(iii)* we combine our proposal with a novel and efficient approach for optimal bit allocation within quantized descriptor representations. By deriving both a Voronoi-based VLAD descriptor (termed as Fast-VVLAD) and a Voronoi-based deep convolutional neural network (CNN) descriptor (termed as Fast-VDCNN), we demonstrate that our Voronoi-based framework is agnostic to the descriptor basis, and can easily be slotted into existing frameworks. Via a range of ROI queries in two standard datasets, it is shown that the Voronoi-based descriptors achieve comparable or higher mean Average Precision against conventional grid-based spatial search, while offering more than two-fold reduction in complexity. Finally, beyond ROI queries, we show that Voronoi partitioning improves the geometric invariance of compact CNN descriptors, thereby resulting in competitive performance to the current state-of-the-art on whole image retrieval.**

***Index Terms*—visual queries, Voronoi partitioning, vector of locally aggregated descriptors, convolutional neural networks, deep learning.**

## I. Introduction

IMAGE retrieval based on visual queries is a topic of intensive research interest since it finds many applications in visual search [1]–[3], detection of copyright violations [4], recommendation services [5] and object or person identification [6]. For much of the past decade, the state-of-the-art for content-based image retrieval was to encode the image by first describing salient points using a locally-invariant feature descriptor, such as SIFT [7] or an image decomposition (e.g., wavelets [8]–[11]). As such, a visual vocabulary is learned offline using K-means or mixture-of-Gaussians (MoG) clustering [12], which quantizes the feature space into cells (visual words). The SIFT cell assignments of each database (or query) image are then produced and aggregated in order to obtain a compact representation that can be used for visual-query based retrieval. Notable contributions in this domain have relied on the bag-of-words (BoW) image representation [13], where the SIFTs assigned to each visual word are aggregated into a histogram used for retrieval purposes. Amongst the successful extensions to BoW are feature soft-assignment [14], spatial matching methods [15]–[17] and indexing methods [13], [18].

Despite the success of BoW approaches, their large storage and memory access requirements make them unsuitable for image retrieval within large image datasets (e.g., tens of millions of images). For such problems, the vector of locally aggregated descriptors (VLAD) [19] was introduced, as a non-probabilistic variant of the Fisher vector image descriptor [20] that encodes the distribution of SIFT assignments according to cluster centers. VLAD has been shown to achieve very competitive retrieval performance to BoW methods with orders-of-magnitude reduction in complexity and memory footprint, i.e., requiring 16–256 bytes per image instead of the tens-of-kilobytes required by BoW methods [19]. With such a reduced memory footprint, it has been shown that a standard multicore server can load and retain the VLADs of a billion-image dataset in its random access memory [21]. This facilitates the scale-up of visual search to big data by using standard cloud computing clusters comprising groups of tens or even hundreds of such servers [22].

However, with increasing dataset sizes and complexity in retrieval, there is a need for deeper models to learn more complex representations and with larger learning capacity. To this end, deep convolutional neural networks (CNNs) have recently come to the forefront in visual recognition [23]–[26]. Deep CNNs, as well as hybrid neural network variants like the FV-NN approach of Peronnin *et al.* [27], have the potential to go beyond "shallow" learned encodings like VLAD because they are scalable with training. For example, deep CNNs trained discriminatively on a large and diverse labelled dataset like ImageNet [28] have been shown to outperform Fisher vectors for image classification [26]. In addition, recent work [2], [29]–[31] demonstrates that features extracted from intermediate layers of a deep CNN are actually transferable to image retrieval. The aggregated features tend to provide

The authors are with the Electronic and Electrical Engineering Department, University College London, Roberts Building, Torrington Place, London, WC1E 7JE, UK (e-mail: {aaron.chadha.14, i.andreopoulos}@ucl.ac.uk). This work has been presented in part at the 2015 Int. Conf. on Computer Vision Syst. (ICVS 2015). The authors acknowledge the support from Innovate UK (project VideoClarity, 101932) and EPSRC (project EP/M00113X/1 and EPSRC CASE award co-sponsored by BAFTA). A. Chadha was supported in part by a fellowship from the Royal Commission from the Exhibition of 1851.

a rich semantic representation of the image, which in the case of retrieval, has been shown to offer comparable, if not substantially better performance to VLAD and Fisher vector descriptors [29], [32], [33]. However, deep CNNs are not without some disadvantages: beyond the high computational cost that is inherent with a large training set and numerous layers [26], [34], the CNN activations lack geometric invariance [33], [35], which pertains to the VLAD descriptor and its variants remaining a viable option, especially for fine-grained search.

In this paper, we are interested in the problem of designing a visual-query based retrieval system that is capable of handling both small and large-size "object", or, more broadly, region-of-interest (ROI) queries over image datasets. Given a ROI representing a visual query, the proposed system should return all images from the database containing this query, with matching complexity and storage requirements that remain of the order of standard encodings. This is considerably more challenging than whole-image retrieval systems, as the query object may be occluded or distorted, or be seen from different viewpoints and distances in relevant images [15]. This is also the reason why the original VLAD proposal does not perform as well for this problem [21]. We therefore propose a new Voronoi-based encoding (VE), in which we spatially partition the image, using a hierarchical K-means, into Voronoi cells and thus compute multiple descriptors over cells. We couple this with an adaptive search algorithm that minimizes the overall computation for similarity identification by first finding the cells most representative to the query and then deriving a novel single-score metric for the image over these cells. We propose a novel product quantization framework (based on symmetric distance computation) for our proposal. Finally, we show that our proposed framework is agnostic to the descriptor basis by testing on both a Voronoi-based VLAD descriptor and Voronoi-based deep CNN feature descriptor and assessing performance against their respective state-of-the-art variants. Overall, our system design for object retrieval adheres to the following principles:

1) The system should provide for substantial improvement over the base descriptor's (VLAD or CNN) mean Average Precision (mAP) when ROI queries are small relative to the image size.
2) The system should maintain competitive mAP to the base descriptor representations under ROI queries occupying a sizeable proportion (or the entirety) of images.
3) The system should be amenable to big-data processing, i.e., its descriptors' size and matching complexity should be comparable to the base descriptor.

In the following section we discuss the background and related work, with Table I summarizing the nomenclature. In Section III we present the offline and online components of our proposed system and Section IV presents the extension of the proposed approach to quantized representations. Section V presents experimental results for Fast-VDCNN on the Holidays dataset [36] and Fast-VVLAD on the Caltech Cars (Rear) dataset [37], and Section VI draws concluding remarks.

## II. Background and Related Work

### A. Vector of Locally Aggregated Descriptors

VLAD is a fixed-size compact image representation that stores first-order information associated with clusters of image salient points [19], [38]. In essence, VLAD is intrinsically related to the Fisher vector image descriptor [39].

In the offline part of the VLAD encoding, based on a training set of $D_{\text{SIFT}}$-dimensional SIFT descriptors derived from $Y$ training images, a visual word vocabulary is first learned using K-means clustering. This vocabulary comprises $K$ clusters with $D_{\text{SIFT}}$-dimensional centroids $\mathbf{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K\}$.

For each new test image $I$ (out of a test dataset comprising $W$ images), $N$ interest points are detected (using an affine invariant detector) and described using $D_{\text{SIFT}}$-dimensional SIFT descriptors, thus forming a descriptor ensemble $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. The descriptors $\boldsymbol{x}_n$, $1 \leq n \leq N$, are assigned to the nearest cluster in the vocabulary via a cluster assignment function $f(\boldsymbol{x}_n)$. VLAD then stores the residuals of the SIFT assignments from their associated centroids. The VLAD $D_{\text{SIFT}}$-dimensional encoding $\boldsymbol{v}_k$ for the $k$-th cluster, $1 \leq k \leq K$, is given by [19], [38]:

$$\boldsymbol{v}_k = \sum_{\forall x_n : f(x_n) = k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k). \tag{1}$$

The VLAD encodings for each cluster are concatenated into a single descriptor $\boldsymbol{\phi}(I) = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K]^{\text{T}}$ with fixed dimension $KD_{\text{SIFT}}$, which is independent of the number of the SIFT descriptors found in the image. The VLAD vectors are then sign square-rooted and $L_2$-normalized [38] and the vectors across all $W$ images of the test dataset are thus aggregated into a single $KD_{\text{SIFT}} \times W$ matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_W]$.

In a practical system, the SIFT descriptor length $D_{\text{SIFT}}$ is typically 128; if the feature space is coarsely quantized with $K$ set to 64, then the VLAD image descriptor has 8192 dimensions. Further dimensionality reduction is achieved with principal component analysis (PCA) (learned on an independent training set), thus further minimizing the memory footprint per image descriptor [40], [41]. The $KD_{\text{SIFT}} \times D$ projection matrix $\mathbf{R}$ used by VLAD comprises only the $D$ largest eigenvectors of the covariance matrix [40], [41]. The projected VLAD, $\widetilde{\boldsymbol{\phi}}_{\text{test}}$, of each image in the test dataset is then $L_2$-normalized, thereby completing the offline part of the VLAD generation.

During online ROI-query based retrieval, after the VLAD encoding and projection of the ROI query has been carried out, the similarity between that and the (projected) VLAD of a test dataset image, $\widetilde{\boldsymbol{\phi}}_{\text{ROI}}$ and $\widetilde{\boldsymbol{\phi}}_{\text{test}}$, can be simply measured using the squared Euclidean distance [38]. With $L_2$ normalized vectors, this is a monotonic function of the inner product, such that:

$$S_{\text{ROI,test}} = \left\langle \widetilde{\boldsymbol{\phi}}_{\text{ROI}}, \widetilde{\boldsymbol{\phi}}_{\text{test}} \right\rangle, \tag{2}$$

where the similarity score $S_{\text{ROI,test}}$ ranges between -1 (completely dissimilar) to 1 (perfect match).

TABLE I
NOMENCLATURE TABLE.

| Symbol | Definition |
|---|---|
| $U$ | dimensions of unprojected descriptor |
| $D, D'$ | dimensions of PCA-projected and truncated descriptor and descriptor blocks (resp.) |
| $Y$ | number of training-set images |
| $W$ | number of test-dataset images |
| $\mathbf{R}$ | $U \times D$ PCA projection matrix |
| $\mathbf{\Lambda}, \mathbf{\Lambda}_{\text{S}}$ | diag. eigenvalue matrix, diag. eigenvalue submatrix |
| $\widetilde{\phi}_{\text{RQI}}, \widetilde{\phi}_{\text{test}}$, and $\widehat{\phi}_{\text{RQI}}, \widehat{\phi}_{\text{test}}$ | PCA-projected descriptor of a query ROI and a test image (resp.), and whitening-and-normalization based product quantization (WNPQ) descriptor of the same |
| $B, B'$ | number of bits for quantized descriptor & constituent block |
| $Z, Z'$ | number of quantization centroids per descriptor and descriptor block |
| $S_{\text{des1,des2}}$ | similarity score between descriptor "des1" and "des2" |
| $M$ | number of quantization subspaces (blocks) for Product Quantization (PQ) |
| $\mathbf{C}_1, \ldots \mathbf{C}_M$ | PQ codebook per quantization block $m$, $1 \leq m \leq M$ |
| $L$ | number of levels (scales) used for Voronoi-based encoding (VE) |
| $V_1, ..., V_L$ | number of Voronoi cells per level $l$, $1 \leq l \leq L$ |
| $V_{\text{tot}}$ | number of Voronoi cells in VE |
| $l_{\text{ph1}}$ | level that Phase 1 exits in Fast-VE adaptive search, $0 \leq l_{\text{ph1}} < L$ |
| $S_0^*, \ldots, S_{l_{\text{ph1}}}^*$ | similarity score for cell with maximum similarity to the query per level $l$, $0 \leq l \leq l_{\text{ph1}}$ |
| $v_0, \ldots, v_{l_{\text{ph1}}}$ | difference between number of interest points in query and cell corresponding to $S_l^*$, $0 \leq l \leq l_{\text{ph1}}$ |
| $\widehat{w}_0, \ldots, \widehat{w}_{l_{\text{ph1}}}$ | L1-normalized Gaussian weighting per level $l$ for Fast-VE, $0 \leq l \leq l_{\text{ph1}}$ |
| $V_{\text{F}}$ | number of cells accessed in Fast-VE |
| $\mathbf{\Sigma}_1, \ldots \mathbf{\Sigma}_M$ | covariance matrix per descriptor block $m$, $1 \leq m \leq M$ |

### B. Multi-VLAD

For ROI-based retrieval, VLAD and the similarity measure of (2) will produce suboptimal results for small ROI, because information encoded from the remaining parts of the dataset image will distort the similarity scoring [21].

Lazebnik *et al.* [42] introduced the concept of spatially partitioning an image into a rectangular grid over multiple scales and encoding per block, as a method of incorporating spatial information; this has found application in both image classification [42], [43] and retrieval [44]. Similarly, the recently-proposed Multi-VLAD descriptor [21] attempts to improve VLAD performance for small ROI by spatially partitioning the dataset images into a rectangular grid over three scales and computing a VLAD descriptor per block.

At the finest scale (level 2), nine VLADs are encoded over a 3×3 rectangular grid. At medium scale (level 1), four VLADs are encoded over a 2×2 grid, where each block is composed of 2×2 blocks from the finest scale. Finally, a single VLAD is encoded over the whole dataset image (level 0). At each scale, Multi-VLAD excludes featureless regions near image borders by adjusting the grid boundary. Moreover, each VLAD is PCA projected and truncated to a 128-dimensional vector. The similarity is thus computed between the VLAD encoded over the query ROI and each of the 14 VLAD descriptors via (2) and the dataset image is assigned a similarity score to the ROI equal to the maximum similarity over its constituent VLADs.

For ROI queries occupying about 11% of image real estate, the Multi-VLAD descriptor has been shown to outperform the single (128 × 14)-D VLAD (computed over the whole image) in terms of mAP. However, Multi-VLAD achieves 20% lower mAP than the (128 $times$ 14)-D VLAD when queries occupy a sizeable proportion of the image [21]. In addition, it incurs a 14-fold penalty in storage and matching complexity in comparison to the baseline 128-D VLAD.

### C. Deep Convolutional Neural Networks for Retrieval

Deep CNNs are feed-forward neural networks comprising multiple layers, and typically trained for classification on large *a-priori* labelled datasets, such as ImageNet [28]. It has recently been shown that extracted features from intermediate layers are transferable to other visual recognition tasks, including image retrieval [29]–[31]. While descriptors derived from these extracted features have been shown to match or outperform "shallow" learned methods, such as VLAD, they suffer due to lack of geometric invariance [33], [35]. Recent works [30], [32], [33], [45] have proposed patch-based methods for overcoming the lack of geometric invariance of the descriptor in instance retrieval. Our proposal is more inline with grid based spatial search methods of Carlsson *et al.* [30], [32], as we are not explicitly computing a global descriptor over extracted features from multiple patches like CNN+VLAD [33] and CKN-mix [45] (which require additional computational pre-processing, e.g., for learning encoding centers). Therefore, we compare performance against a generic grid-based spatial search, which we refer to as Multi-CNN.

A Multi-CNN descriptor can be devised analogously to Multi-VLAD, i.e., by dividing the image at level $l$ into an $(l+1) \times (l+1)$ grid and computing the similarity score between two images as the global maximum inner product over all partitions. For the general case of $L$ levels, the total number of partitions (incl. the whole image as level 0) is:

$$P_{\text{tot}} = \sum_{l=0}^{L-1} (l+1)^2 = \frac{(L+1)(L+2)(2L+3)}{6} \qquad (3)$$

In this paper we only consider networks pre-trained on ImageNet [28]. While fine-tuning on a tailored dataset may increase performance [29], this requires additional training and diverts away from the generality of the CNN features extracted from a large and diverse dataset.

### D. Product Quantization

In order to further reduce the search complexity and the required memory footprint when handling large datasets, $D$-dimensional vectors are typically quantized to produce compact $B$-bit representations [46].

Consider a $D$-dimensional query vector $\widetilde{\boldsymbol{\phi}} \in \mathbb{R}^D$. A global K-means clustering approach can be used to map $\widetilde{\boldsymbol{\phi}}$ to vector $q(\widetilde{\boldsymbol{\phi}})$ in codebook $\mathbf{C} = \{\boldsymbol{c}_i, 1 \leq i \leq Z\}$. For quantizer $q$ with $Z$ centroids, the total number of bits used to encode $\widetilde{\boldsymbol{\phi}}$ is $B = \log_2(Z)$. However, for a 64-bit encoding of a 128-D query vector (0.5 bits per dimension), $Z = 2^{64}$ centroids must be learned using K-means, which is clearly infeasible. The learning and storage requirements of this quantization problem can be reduced either via traditional approximate nearest neighbor (ANN) algorithms [47]–[50], or more recent advances [51]–[53]. In this paper, we focus on an efficient method for ANN search, named product quantization (PQ) [46], which uses multiple subquantizers rather than a single global quantizer [46]. PQ considers each unquantized vector $\widetilde{\boldsymbol{\phi}}$ as the concatenation of $M$ subvectors, $\widetilde{\boldsymbol{\phi}} = \left[\widetilde{\boldsymbol{\phi}}_1, \widetilde{\boldsymbol{\phi}}_2, \ldots, \widetilde{\boldsymbol{\phi}}_M\right]$, each with equal dimension $D' = D/M$. Each subvector $\tilde{\boldsymbol{\phi}}_m$ is encoded from its own subcodebook $\mathbf{C}_m = \{\boldsymbol{c}_{m,i}, 1 \leq i \leq Z'\}$, learned using K-means and considered to be of size $Z' = Z^{\frac{1}{M}}$ for all $m$, $1 \leq m \leq M$. As such, the new codebook $\mathbf{C}$ is the Cartesian product of the subcodebooks, with total size $Z = (Z')^M$:

$$\mathbf{C} = \mathbf{C}_1 \times \cdots \times \mathbf{C}_M. \tag{4}$$

Crucially, via this vector partitioning approach, the learning complexity and storage requirement is reduced to $\mathcal{O}(MZ'D') = \mathcal{O}\left(Z^{\frac{1}{M}}D\right)$. The total number of bits used to encode each $\widetilde{\boldsymbol{\phi}} \in \mathbb{R}^D$ is now given by $B = M \times B'$, where $B'$ is the number of bits used to encode each subvector $\boldsymbol{x}_m$, i.e., $B' = \log_2(Z') = \frac{B}{M}$.

Previous work proposed PQ with asymmetric distance computation (ADC) [19], [38], which only encodes the vectors of the test dataset, and PQ with symmetric distance computation (SDC) [46], where both query and test vectors are quantized. By not encoding the query vectors, ADC reduces the overall quantization distortion, thus enhancing the discriminatory power of the system. On the other hand, in SDC the distances between any two subcodewords in the $m$-th subspace are precomputed and stored in a $Z' \times Z'$ lookup table, thus enabling efficient ANN search by simple lookup table accesses. Experimental results [19], [38], have shown that ADC and SDC variants of PQ-based VLAD achieve comparable retrieval performance to unquantized VLAD representations with four to ten-fold reduction in storage and search complexity.

## III. Proposed Voronoi-based Encoding and its Fast Online Implementation

The Voronoi-based encoding proposed in Subsection III-A constitutes the offline component of our system. Subsection III-B describes the proposed acceleration for online Voronoi-based ROI query search and possibilities for memory compaction to reduce storage requirements.

### A. Voronoi-based Encoding and Compact Descriptors

Instead of spatially partitioning the images into a rectangular grid, we propose to partition the image into Voronoi cells over $L$ levels (scales), using hierarchical spatial K-means clustering. The key intuition is that objects that may constitute ROI queries tend to appear as clusters of salient points, potentially interspersed with featureless regions in the image. Therefore, a ROI-oriented partitioning must attempt to adaptively isolate these spatial clusters at multiple levels.

Initially, the entire image is encoded; this comprises level 0 of the Voronoi-based encoding. For level 1, a spatial K-means is computed over the interest point locations in the whole image, which effectively partitions the image into $V_1$ Voronoi cells. Next, for level 2, a spatial K-means is computed over the interest point locations within each level-1 Voronoi cell, thus partitioning each cell into $V_2$ constituent cells. In general, for level $l$, $1 \leq l < L$, each of the $V_{l-1}$ cells of the previous level is partitioned into $V_l$ cells, with $V_0 \triangleq 1$. A base descriptor, whether this be VLAD or aggregated deep CNN features, is encoded over each cell following the description of Section 2.1, giving a total of

$$V_{\text{tot}} = 1 + \sum_{l=1}^{L-1} \prod_{m=1}^{l} V_m \tag{5}$$

encodings per image. When PCA-projecting each cell descriptor, we aggregate each level $l$ into a single matrix $\boldsymbol{\Phi}_l$.

A three-level Voronoi partitioning for an image from the Caltech Cars image dataset with $V_1 = V_2 = 3$ is illustrated in Fig. 1. The detected points are shown in color in the left image of Fig. 1, and the level-1 and level-2 Voronoi cells are superimposed with dashed lines on the middle and right image (resp.), with their corresponding descriptors appearing with different colors.

In essence, there are two variables to consider when implementing a Voronoi-based encoding; the number of levels $L$ and the number of Voronoi cells $V_l$, $1 \leq l < L$, to encode. For the purposes of this paper, we will consider $V_l$ to be constant for all levels $l$. In addition, it is worth noting that for the Voronoi-based encoding, we construct a single PCA projection matrix using the entire images of the training set. This is because we found that there is very little gain in retrieval performance when learning separate PCA projection matrices for each Voronoi partition level, mostly due to sufficient variability in scale of ROI in the training images alone. Finally, given we are dealing with PCA on high dimensional data, for the case where the unprojected cell descriptor dimension, $U$, is greater than the training size $Y$, we use the manipulation described by Bishop [54]. In essence, we define the covariance matrix for the $U \times Y$ training descriptor matrix $\boldsymbol{\Phi}_{\text{T}}$ as $\boldsymbol{\Phi}_{\text{T}}^T \boldsymbol{\Phi}_{\text{T}}$, which provides for a lower dimensional $Y \times Y$ matrix to work with. Following singular value decomposition, we then rotate the derived projection matrix, $\mathbf{R}_{\text{ROT}}$, into the original covariance data space to obtain the $U \times D$ projection matrix, $\mathbf{R}$, using the equivalence:

$$\mathbf{R} = \boldsymbol{\Phi}_{\text{T}} \mathbf{R}_{\text{ROT}} \boldsymbol{\Lambda}^{-1}, \tag{6}$$

where $\boldsymbol{\Lambda} = \text{diag}(\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}}, \ldots, \lambda_Y^{-\frac{1}{2}})$ is the diagonal matrix of eigenvalues of $\boldsymbol{\Phi}_{\text{T}}^T \boldsymbol{\Phi}_{\text{T}}$.

We conclude this subsection by summarizing the VLAD and deep CNN descriptors utilized for each Voronoi partition.

*1) Voronoi-based VLAD (VVLAD):* We require a detector that is robust to scale and viewpoint changes, while also detecting enough points in salient regions to allow for reliable partitioning. Therefore, for VVLAD, we use the Hessian Affine detector [55], [56], which is based on the multi-scale determinant of the Hessian matrix (computed locally), and detects affine covariant regions. SIFT descriptors are produced based on the detected points. It is worth noting that: *(i)* salient point detection is an implicit step in each VLAD computation and not additional processing; *(ii)* unlike Multi-VLAD, there is no need to preprocess the image and exclude featureless regions. As shown in the example of Fig. 1, smaller Voronoi cells are adaptively formed around regions of tight clusters of detected points.

*2) Voronoi-based Deep CNN (VDCNN):* In this case, the salient point detection constitutes additional pre-processing. Nevertheless, this can be achieved efficiently by using the FAST corner detector [57], which classifies a pixel as a corner based on its relative intensity to a set of contiguous pixels. As for the case of VVLAD, the image is partitioned into Voronoi cells based on the location of detected points. Since the deep CNN must take a rectangular input image segment, we compute a bounding box over the constituent points of each cell, and then resize and subtract an average image, as per convention, before feeding into the pre-trained deep CNN. Given that the cells are treated independently, the feed-through can be done in parallel, using multiple copies of the network. In terms of the deep CNN descriptor specifics, we use the CNN-S architecture [58] pretrained on ILSVRC-2012 with batch normalization [59]. The network is sufficiently deep to provide a rich semantic representation of the image/image partitions without overfitting to the classification task. The conventional approach to generating a feature descriptor from the network is to simply extract one of the fully-connected layers [26], [29], [32]. Instead, we extract the last max-pooling layer (Layer 13) of the network, which precedes the fully-connected network and should be less tuned to the classification task. From this layer, we generate a 512-D feature descriptor by averaging the CNN activations over the spatial dimensions. We can also (optionally) apply PCA-projection and truncation to achieve further compaction to 128 dimensions.

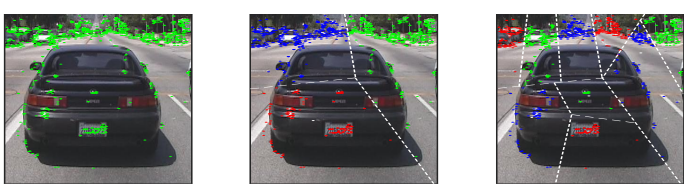


Fig. 1. Three-level Voronoi partitioning for an image from Caltech Cars dataset. For illustration purposes, SIFT descriptors are color-differentiated for each cell.

### *B. Fast Online Implementation: Adaptive Search and Image Similarity Score*

Conventionally, we could assign an image score as the global maximum similarity to a query over cells, using (2) for each cell. However, the proposed Voronoi partitioning essentially gives us a tree of spatial Voronoi cells where, for $L$ levels, $\prod_{l=1}^{L-1} V_l$ "leaf" Voronoi cells exist at the bottom of the tree. Given that there is inherent mutual information between a cell and its constituent cells, rather than accessing data for all levels and measuring similarity over all $V_{\text{tot}}$ cells of the tree indiscriminately, we can design an adaptive search with top-to-bottom tree pruning to find the most relevant Voronoi cells to the query. This reduces the overall execution time and memory accesses when performing a retrieval task, which makes our proposal applicable to very large image databases that would contain millions of images. The top-to-bottom search is carried out in two phases.

**Phase-1:** Considering the cell of level $l-1$ with maximum similarity to the query [measured via (2)], in Phase-1 of the search, we assume that either this cell or a constituent cell within it (at level $l$) will attain high similarity to the query. If the cell of level $l-1$ is found to attain the highest similarity to the query, we terminate the search for that image at level $l-1$ and proceed to Phase-2. On the other hand, if we find that a constituent cell of level $l$ attains the maximum similarity, we repeat Phase-1 for that cell and its constituent cells at the next level $(l+1)$, until we reach the bottom of the tree, in which case we move to Phase-2.

**Phase-2:** Let us denote the maximum similarity found by Phase-1 for each level $l$ as $S_l^*$ and assume that Phase-1 exited at level $l_{\text{ph1}}$, $0 \leq l_{\text{ph1}} < L$. Rather than assigning $S_{l_{\text{ph1}}}^*$ as the similarity score between the ROI query and the test image $I$ in the dataset, we perform a weighted sum over all $S_0^*, \ldots, S_{l_{\text{ph1}}}^*$. To this end, we first compute the difference $v_l$, $0 \leq l \leq l_{\text{ph1}}$, between the number of interest points in the query and the number of interest points in the image dataset cell corresponding to $S_l^*$. This difference is subsequently used within a scaled inverse function. The weight for $S_l^*$ $(0 \leq l \leq l_{\text{ph1}})$ is thus defined as:

$$w_l = \frac{C}{\max\left(|v_l|, 1\right)}. \tag{7}$$

where $C$ controls the order (set as the modal order of magnitude over all $v_l$). The weight vector over all levels is $L_1$-normalized so that the image score can be ranked independently of the level $l_{\text{ph1}}$ at which Phase-1 terminated. Denoting the $L_1$-normalized weight as $\hat{w}_l$, the proposed similarity score between a ROI query and dataset image $I$ after Phase-2 is:

$$S_{\text{ROI},I} = \sum_{l=0}^{l_{\text{ph1}}} \hat{w}_l S_l^*. \tag{8}$$

For example, for a three-level partition, if a query object is small relative to the image size, we expect that the total number of interest points over the query would be comparable to that of a level-2 cell. Hence, the level-2 maximum dot product $S_2^*$ should receive the largest weighting $\hat{w}_2$ when computing the similarity score. This is expected to be a more robust similarity scoring than just taking a global maximum over all $S_0^*, \ldots, S_{l_{\text{ph1}}}^*$ (as in Multi-VLAD) as the similarity score, since we account for relevant information from all levels.

**Summary:** We term this two-phase search coupled with the Voronoi-partitioning as *Fast Voronoi-based encoding* (Fast-VE), because it reduces the expected number of cells that are accessed at runtime. The upper bound for the matching complexity is now:

$$V_{\text{F}} = 1 + \sum_{l=1}^{\min\{l_{\text{ph1}}+1, L-1\}} V_l \tag{9}$$

inner products per image instead of the $V_{\text{tot}}$ inner products required using a global maximum similarity measure that considers all cells. Due to the weights of (8), per image $I$, along with the Voronoi-based encoding we also store the number of interest points per cell, comprising $V_{\text{tot}}$ additional values.

It is worth noting that further storage compaction of the Fast-VE is feasible using level projection. Via level projection, we can adhere to memory constraints of a practical deployment for very-large image datasets by only storing the PCA projected cell descriptors for the last level, $L-1$ and computing the cell descriptors for levels $0, ..., L-2$ at runtime by aggregating smaller-cell descriptors. Given that such storage compaction is of secondary importance in the overall unquantized and quantized Fast-VE design, we include its details as supplementary information in Appendix A.

## IV. Product Quantization for Efficient VE Search

Given that quantized descriptor representations offer significantly-higher compaction than unquantized ones, we extend VE and Fast-VE to quantized representations via a specially-designed product quantization framework.

### A. Product Quantization based on Symmetric Distance Computation for Voronoi-based Encoding

We consider PQ based on SDC for the proposed VE approach[1], where both the query vector $\boldsymbol{y}$ and test dataset vector $\boldsymbol{t}$ are quantized [46]. We opted for SDC-based rather than ADC-based PQ because ADC-based methods require the precomputation and storage of distances between VE query and test vectors, which is not feasible in a large-scale image retrieval system where potentially any image could form a query.

In SDC-based PQ, the nearest neighbour to $\boldsymbol{y}$ can be approximated by optimizing the distance function $d(q(\boldsymbol{y}), q(\boldsymbol{t}))$. The distance function is typically the squared Euclidean distance [46]:

$$\begin{aligned} d(q(\boldsymbol{y}), q(\boldsymbol{t})) &= \|q(\boldsymbol{y}) - q(\boldsymbol{t})\|^2 \\ &= \sum_{m=1,\ldots,M} d(q_m(\boldsymbol{y}_m), q_m(\boldsymbol{t}_m)) \\ &= \sum_{m=1,\ldots,M} \|q_m(\boldsymbol{y}_m) - q_m(\boldsymbol{t}_m)\|^2, \end{aligned} \tag{10}$$

with $M$ the number of subquantizer blocks of (4).

The key intuition behind the modified PQ for VE is to treat the constituent Voronoi cells as images and apply PQ on each query and test cell. A single PQ codebook $\mathbf{C}$ is learned using K-means clustering on a training set. Each cell descriptor from the test dataset is thus considered as a concatenation of $M$ subvectors $\widetilde{\boldsymbol{\phi}} = \left[\widetilde{\boldsymbol{\phi}}_1, \ldots, \widetilde{\boldsymbol{\phi}}_M\right]$ of $D' = D/M$ elements each, with each subvector being encoded from its corresponding subcodebook $\mathbf{C}_m$. In this way, there is also no dependency on the level $l$, as we quantize the cell descriptors from a single PQ codebook. All possible distance values between the $i$th and $j$th subcodebook vectors in the $m$-th subspace, $d(\boldsymbol{c}_{m,i}, \boldsymbol{c}_{m,j})$, are pre-computed and stored in a $Z' \times Z'$ lookup table, thus enabling efficient ANN search by simple lookup table accesses.

As the subspaces are orthogonal, we $L_2$-normalize the product quantization of each cell descriptor's subquantizer block $m$, $1 \leq m \leq M$, by normalizing the columns of the PQ subcodebooks $\mathbf{C}_m$ individually before computing and storing the distance values. For the $i$th subcodebook vector in the $m$-th subspace, the normalization term is given by $\sqrt{M}\,\|\boldsymbol{c}_{m,i}\|$. As such, the distance value to be stored between the $i$th and $j$th subcodebook vectors is

$$d(\boldsymbol{c}_{m,i}, \boldsymbol{c}_{m,j}) = \frac{\langle \boldsymbol{c}_{m,i}, \boldsymbol{c}_{m,j} \rangle}{M\,\|\boldsymbol{c}_{m,i}\|\,\|\boldsymbol{c}_{m,j}\|}. \tag{11}$$

Quantizing from the normalized PQ subcodebooks, the distance function between a subspace normalized and quantized query cell descriptor $q(\boldsymbol{y})$ and a subspace normalized and quantized test cell descriptor $q(\boldsymbol{t})$ is now simply

$$d(q(\boldsymbol{y}), q(\boldsymbol{t})) = \sum_{m=1,\ldots,M} \langle q_m(\boldsymbol{y}_m), q_m(\boldsymbol{t}_m) \rangle, \tag{12}$$

which is analogous to the squared Euclidean distance of normalized vectors. Importantly, this bounds the similarity score between -1 and 1, which facilitates performance comparisons.

Fig. 2 illustrates two indicative examples of SDC-based PQ on the $D = 128$ and $D = 2048$ dimensional VLAD, both with and without subspace normalization. The retrieval performance is measured in terms of mean average precision (mAP) on the Holidays dataset [36], using whole-image queries. It is evident from the results of the figure that, for given dimension $D' = D/M$, subspace normalization actually improves retrieval performance, effectively peaking close to $M = D/4$. At this block size, the subspace dimensionality is sufficient such that each subspace is optimally regularized. In addition, we observe that the performance margin between the VLAD descriptor and its subspace normalized counterpart increases significantly with dimension $D$.

Essentially, we want the block size to be large enough that we encode over a sufficient number of bits; however, beyond a certain block size, we end up normalizing over too few dimensions. In this regard, it is interesting to consider the limit case, where $M = D$, i.e., $D' = 1$. There, subspace normalization results in storing just the sign per cell descriptor. In this extreme case, the similarity between cells can be computed very efficiently by using the Hamming distance, i.e., without accessing any lookup tables.

Concerning storage requirements, assuming that the components of the unquantized cell representation are kept as 32-

[1]refer to Section II-D for nomenclature and symbol definitions

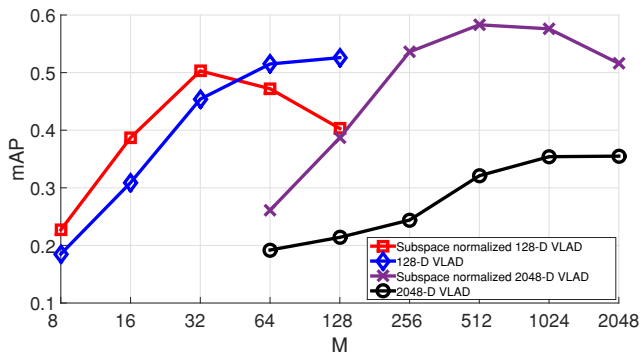



Fig. 2. Plot of mean average precision (mAP) with varying number of PQ blocks, $M$, for PQ VLAD descriptors on the Holidays dataset.

bit floating-point numbers, their offline storage requirement is $D \times 32$ bits per test image. On the other hand, our product-quantized cell descriptor requires $V_{\text{tot}} \times B$ bits per test image, which is independent of the dimension, $D$. In addition, for the entire test dataset, the total storage requirement for the quantization lookup tables is $Z' \times Z' \times M$. As the test dataset grows in size, this value becomes negligible in comparison to the storage requirement for the product-quantized descriptors.

Finally, with regards to the search complexity, the inner products have been replaced by read accesses to the look-up tables. As such, the product quantized Fast-VE now has an upper bound on complexity of $M \times V_{\text{F}}$ reads, which is independent of the descriptor dimension per cell.

### *B. Optimal Bit Allocation via Whitening and Subspace Normalization*

Given the presence of multiple cells in the Voronoi-based encoding, it is important to derive an appropriate bit allocation strategy that minimizes the quantization distortion.

**Assumption 1.** *We consider successive samples of each subspace-normalized VE component (dimension) $\widetilde{\phi}_i$ $(1 \leq i \leq D/M)$ to be modelled by independent, normally-distributed, random variables, with corresponding variance $\sigma_i$.*

Under Assumption 1, the normalized random vectors $\widetilde{\boldsymbol{\phi}}_m$ of all subspaces $m$, $1 \leq m \leq M$, can then be represented by independent and identically-distributed multivariate Gaussians[2], with corresponding diagonal covariance matrices $\boldsymbol{\Sigma}_m$. The rate-distortion function for independent, normally-distributed random variables [62] can be extended to the multivariate case in order to derive the optimal bit allocation strategy for VE. This leads to the following proposition.

**Proposition 1.** *Under Assumption 1, optimal bit allocation after subspace normalization in VE can be achieved by balancing the variances of the subspaces.*

*Proof:* See Appendix B. ■

[2]The Gaussian assumption is necessary for some of the theoretical derivations, but is also proven to hold in practice [60], [61].

Indeed, recent work [61] employs an optimized product quantization (OPQ) that effectively leads to balanced subspace variances by assigning principal components to a subspace with the objective of balancing the product of eigenvalues per subspace. This corresponds to performing a permutation of the principal components to achieve balanced variances. Jegou *et al.* [38] propose balancing the component variance with a random orthogonal rotation, but this removes the decorrelation achieved by PCA. A different approach is proposed by Brandt *et al.* [60]: one can achieve a constant quantization distortion per subspace by varying the number of bits assigned to each principal component, at the cost of increased training and runtime complexity. Finally, Spyromitros-Xioufis *et al.* [63] consider the effects of applying a random orthogonal rotation on PCA-projected and whitened VLAD vectors prior to product quantization. However, whitening inherently balances the subspace variances by setting $\boldsymbol{\Sigma}_m$ to the identity matrix for all $m$, which also preserves decorrelation and mitigates descriptor bias from visual word/component co-occurrences [40], [41]. As such, we propose a simple and effective solution for the bit allocation that adheres with the theoretical result of Proposition 1: we use a whitening approach after PCA (and prior to the product quantization), together with the subspace normalization described in the previous section (and shown to be beneficial by the tests of Fig. 2). Specifically, per cell, we can express the relationship between a projected descriptor $\widetilde{\boldsymbol{\phi}}_m$ and its whitened and normalized counterpart $\widehat{\boldsymbol{\phi}}_m$ in the $m$-th subspace as:

$$\widehat{\boldsymbol{\phi}}_m = \frac{\boldsymbol{\Lambda}_{\text{S},m} \widetilde{\boldsymbol{\phi}}_m}{\left\| \boldsymbol{\Lambda}_{\text{S},m} \widetilde{\boldsymbol{\phi}}_m \right\|}, \tag{13}$$

where $\boldsymbol{\Lambda}_{\text{S},m} = \text{diag}(\lambda_{a+1}^{-\frac{1}{2}}, \lambda_{a+2}^{-\frac{1}{2}}, \ldots, \lambda_{a+D'}^{-\frac{1}{2}})$ is the diagonal subspace matrix of eigenvalues of the training-set covariance matrix, with $\lambda_i$ associated with the $i$-th largest eigenvector and $a = D'(m-1)$.

The advantage of using whitening and normalization against previous approaches is that there is no need for any additional pre-processing, such as learning a rotation matrix or variability in the bit allocation across the principal components. We term our approach *whitening & normalization based product quantization* (WNPQ).

## V. Experimental Evaluation

### *A. Datasets*

We measure performance on the Holidays and Caltech Cars (Rear) test image datasets. For both datasets, a set of predefined queries and hand-annotated ground truth is used.

**Caltech + Stanford Cars** [37], [64]: This test dataset consists of 1155 (360 ×) 240) photographs of cars taken from the rear. Subsequently, we test on a subset of 416 images from the Caltech Cars (Rear) dataset, from which we select 10 images and perform three tests: *(i)* we mimic a surveillance test by selecting only the license plates as ROI-queries; *(ii)* we select as mid-scale ROI-queries a section of the car trunk, and *(iii)* use the whole images as queries.

An example of the query subset is given in the left part of Fig. 3. For the license plate test, we manually create "good" and "junk" ground-truth files over matching images [15]; "junk" ground truth comprises any image in which the query (i.e., the license plate) is barely visible or not distinguishable by the interest point detector. To provide a more rigorous and diversified test, we combine the Caltech Cars subset with another independent set of 1000 distractor images from the Stanford Cars dataset [64], comprising various car models and orientations, giving the Caltech + Stanford Cars dataset.

**Holidays** [36]: The Holidays test dataset consists of 1491 images, mainly consisting of holiday photos. There are 500 "whole image" provided queries of a distinct scene or object. In order to test on a smaller scale, we also select salient regions from a subset of 40 query images as ROI queries into our system. An example ROI query with its corresponding matching image set is shown in the right part of Fig. 3.

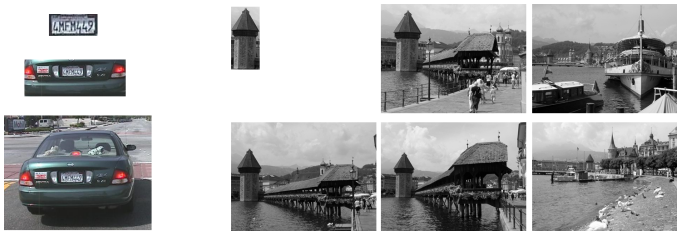


Fig. 3. (Left) Example queries for the Caltech Cars dataset. (Right) Example ROI query (top left) and matching image set for the Holidays dataset (remaining images).

### B. Setup

Unless stated otherwise, all vectors are whitened and re-normalized post-PCA. The retrieval performance is measured by creating a ranked list and computing the mAP over all queries. Matching complexity is defined as the number of multiply-accumulate (MAC) operations for unquantized descriptors, or the number of look-up table reads for quantized descriptors. Per descriptor, we report the matching complexity averaged over all tests and normalized to the baseline 128-D descriptor complexity, along with the descriptor storage size, in bytes.

**Caltech + Stanford Cars**: Due to the specificity of the Caltech + Stanford Cars dataset, together with the lower ROI resolutions, using a deep CNN pre-trained on ImageNet is not a viable option. For example, ImageNet (ILSVRC2012 dataset [59]) does not contain any substantial number of images (and associated labels) corresponding to car license plates; therefore, the pre-trained deep CNN descriptor will not be suitable for such images. For these reasons, we have established that, in this dataset, the utilized deep CNN descriptor is outperformed by VLAD descriptor variants, particularly on license-plate queries. Thus, we use this dataset to test how the proposed Voronoi-based encoding performs with the "shallow" learned VLAD descriptor of Subsection III-A.1.

For the VLAD computation, we follow the design of Subsection II-A. The PCA projection matrix, visual word centers and PQ codebook are learned on an independent dataset of 2000 car images from the Stanford Cars dataset [64]. For Fast-VVLAD, we set: $K = 64$, $L = 3$, $V_1 = V_2 = 3$, with 128-D VLAD per cell and compile a ranked list from the relevant similarity score. For VLAD, we use: 128-D, 768-D and 1664-D sizes, in order to align the VLAD matching complexity with that of the Fast-VVLAD descriptor. We set $L = 3$ for the Multi-VLAD descriptor, such that it is inline with our Voronoi partitioning. This results in a (128 $\times$ 14)-D descriptor size per image [21], as derived from (3).

*WNPQ Parameter Selection:* We set $M = 32$ for the quantized 128-D VLADs. For the 768-D and 1664-D quantized VLADs, we respectively set the block size to $M = 96$ and $M = 208$. For the quantized VVLAD, we set $M = 32$ for all cell VLADs. Finally, quantized Multi-VLAD also uses $M = 32$. These settings for the block size were chosen to align the matching complexity of the quantized 1664-D VLAD with that of the quantized Fast-VVLAD, whilst providing the 768-D VLAD as a solution with mid-range complexity. Notably, we fix $Z' = 256$ for all experiments. Higher values for $Z'$ increase the computational load for each block quantizer, whilst increasing the storage requirement of the look-up tables ($\mathbb{O}(Z'^2)$), which is an important detriment as these tables need to be sufficiently small to fit in cache memory [46].

**Holidays**: The Holidays dataset provides a less controlled test for our system. The scenes in the Holidays dataset are better represented by a deep CNN architecture trained on ImageNet, particularly due to their high resolution. Similar to prior work [65], we have confirmed that deep CNNs substantially outperform VLAD descriptors for this dataset. Therefore, we use this dataset to test how the proposed Voronoi-based encoding performs with the deep CNN descriptor of Subsection III-A.2.

For the utilized CNN-S architecture [58], all images and image partitions are resized to 224 $\times$ 224 and fed into the network after subtracting an average image. The final feature descriptor is 512-D, which can then be normalized, PCA-projected to 128-D and whitened. However, following a similar approach to the instance retrieval pipeline on the VLAD descriptor, we normalize, sign-square root and re-normalize the feature descriptor prior to PCA and whitening, with the intention of minimising the burstiness of dimensions and thus adding to descriptor invariance [38]. It is worth noting that contrary to recent works [65], [66], we do not manually rotate the images in the Holidays dataset as we do not deem this to be a fair representation of data 'in the wild'.

*Parameter selection for the Voronoi partitioning:* For the VE of all cases, the FAST corner detector [57] is used, and we learn the PCA projection matrix and PQ codebook on a subset of 4000 images from the ILSVRC-2010 validation set. For Fast-VDCNN, we set: $L = 3$, $V_1 = V_2 = 3$, with a 128-D CNN feature descriptor per cell and compile a ranked list from the relevant similarity score. We compare this with a 128-D and the full unprojected 512-D CNN feature descriptors.

For Multi-CNN, we use the same grid partitioning as Multi-VLAD, with $L = 3$, thus producing a $(128 \times 14)$-D size per image.

*WNPQ Parameter Selection:* We set $M = 32$ for the quantized 128-D CNN feature descriptors. For the 512-D quantized CNN feature descriptor, we set the block size to $M = 128$. For the quantized Fast-VDCNN, we set $M = 32$ for all cell descriptors. Finally, quantized Multi-CNN also uses $M = 32$. As with the VLAD descriptors, we fix $Z' = 256$ for all experiments to keep the storage requirement for the lookup tables constant.

### C. Results with Unquantized Descriptors

This section summarises performance using unquantized descriptors on the Caltech + Stanford Cars and Holidays dataset.

**Caltech + Stanford Cars**: Table II summarizes the retrieval performance of all unquantized VLAD methods on the Caltech + Stanford Cars dataset. The first observation is that the Fast-VVLAD descriptor offers competitive performance to the larger 1664-D VLAD, whilst decreasing the matching complexity by more than 50%. In addition, Fast-VVLAD performs significantly better on license plate queries than both the 128-D VLAD and its 768-D VLAD complexity counterpart, yielding respective mAP gains of over 200% and 41%. Fast-VVLAD maintains consistently-good mAP even with the larger ROIs of car trunks and whole-image queries, and is only outperformed on whole-image queries by VLAD by (up to) a 7% margin. Finally, Fast-VVLAD maintains competitive performance to Multi-VLAD on all query types, whilst offering lower dimensionality and matching complexity.

**Holidays**: Table III summarises the retrieval performance for the 500 whole-image queries and 40 smaller ROI queries on the Holidays dataset. Interestingly, the Fast-VDCNN remains competitive on whole image queries. This is attributed to the Fast-VDCNN similarity score of (8) that considers all partition levels, which provides robustness against false positives. The Fast-VDCNN is found to outperform Multi-CNN for whole image queries and maintain very competitive performance on ROI queries, while offering more than 50% reduction in the matching complexity[3]. Fast-VDCNN was also found to substantially outperform the lower dimensional CNN feature descriptors for ROI queries (gains exceeding 50% in mAP). Given that the utilized CNN-S descriptor derived from Layer 13 is limited to 512 dimensions [58], we also benchmarked using the first fully connected layer (FC1), which allows for a large 4096-D feature descriptor. Nevertheless, the FC1 descriptor performed significantly worse than our 512-D Layer 13 descriptors for both query ROI and whole images, scoring mAP of 28.3% and 71.4%, respectively. This serves as an additional validation for our choice for the utilized CNN layer.

[3]We have also validated that this saving translates to practical runtime saving: by adding a large distractor set (thereby scaling the dataset size to 150K images), we found that Fast-VDCNN based retrieval is 40% faster than Multi-CNN retrieval, with execution time comparable to the baseline 512-D CNN feature descriptor.

TABLE II
COMPLEXITY AND MAP RESULTS FOR THE CALTECH + STANFORD CARS IMAGE DATASET [37], [64].

| | $D_{\text{tot}}$ | Matching Complexity | Descriptor Storage (bytes) | License Plates | Trunk | Whole Image |
|---|---|---|---|---|---|---|
| VLAD [38] | 128 | 1 | 512 | 0.148 | 0.669 | 0.729 |
| | 768 | 6 | 3.07k | 0.348 | 0.739 | 0.780 |
| | 1664 | 13 | 6.66k | **0.512** | 0.722 | **0.785** |
| Proposed Fast-VVLAD | $128 \times 13$ | 6.55 | 6.66k | 0.490 | 0.745 | 0.728 |
| Multi-VLAD [21] | $128 \times 14$ | 14 | 7.17k | 0.493 | **0.780** | 0.732 |

TABLE III
COMPLEXITY AND MAP RESULTS FOR THE HOLIDAYS DATASET [36].

| | $D_{\text{tot}}$ | Matching Complexity | Descriptor Storage (bytes) | Query ROI | Whole Image |
|---|---|---|---|---|---|
| CNN (Layer 13) | 128 | 1 | 512 | 0.339 | 0.757 |
| | 512 | 4 | 2.05k | 0.369 | **0.767** |
| Proposed Fast-VDCNN | $128 \times 13$ | 6.11 | 6.66k | 0.674 | 0.761 |
| Multi-CNN [21], [58] | $128 \times 14$ | 14 | 7.17k | **0.678** | 0.737 |

### D. Results with Quantized Descriptors

We now consider performance when integrating quantization into all approaches under consideration.

**WNPQ against other quantization methods:** We first consider the performance of the proposed WNPQ method against other state-of-the-art methods, namely the parametric optimized product quantization (OPQ) [61] and product quantization with a random rotation pre-processing (RRPQ) [38]. As the OPQ and RRPQ descriptors are not normalized, we use the squared Euclidean distance metric for these methods and compare retrieval performance on both datasets. The results of Fig. 4 show that the proposed WNPQ method outperforms RRPQ and, for the majority of the tests, also outperforms OPQ. Essentially, the WNPQ maintains its high retrieval performance when the dimensionality is increased from 128-D to the 768-D and 1664-D VLAD descriptors. To ensure a fair comparison, and because the proposed WNPQ was shown to provide for the best overall performance, we use it to quantize all the descriptors under comparison.

**Caltech + Stanford Cars:** Table IV summarises the performance of the various descriptors with WNPQ on the Caltech + Stanford Cars dataset. On whole images, coupled with the aggregated similarity score, Fast-VVLAD offers superior performance to the 128-D VLAD, with an mAP gain of 6%. The 1664-D VLAD, which is now of comparable complexity to the Fast-VVLAD, is outperformed on the small license plate queries, with mAP gain of 9%, but remains superior for whole-image queries. However, it is worth mentioning that the gain from Fast-VVLAD on small queries outweighs any loss on larger queries, thus making it favorable. Finally, the quantized Multi-VLAD offers marginally superior mAP to Fast-VVLAD, albeit at the cost of twice the matching

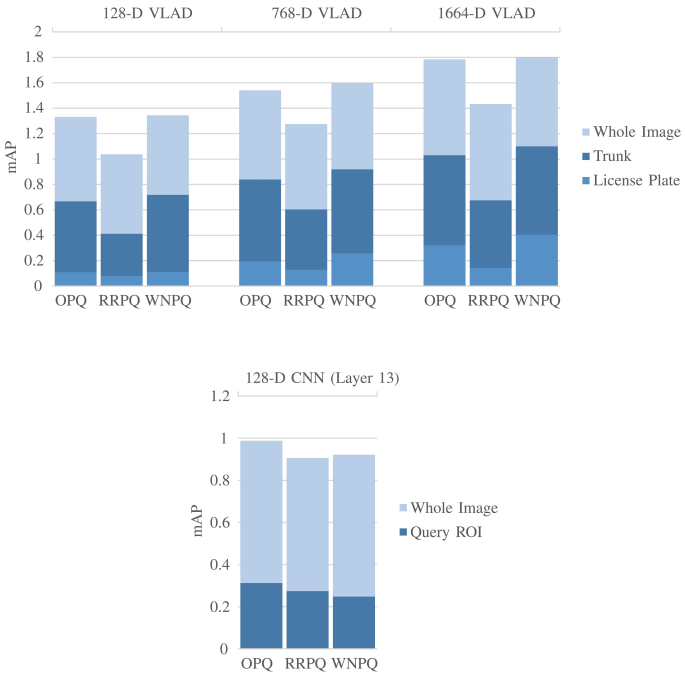



Fig. 4. Comparison of quantization methods. Top: Caltech + Stanford Cars dataset: mid blue = license plate, light blue = whole image, dark blue = trunk. Bottom: Holidays (128-D): dark blue = query ROI, light blue = whole image.

TABLE IV
COMPLEXITY AND MAP RESULTS FOR THE CALTECH + STANFORD CARS IMAGE DATASET WITH WNPQ [37], [64].

| | $D_{\text{tot}}$ | Matching Complexity | Descriptor Storage (bytes) | License Plates | Trunk | Whole Image |
|---|---|---|---|---|---|---|
| WNPQ VLAD | 128 | 1 | 32 | 0.112 | 0.606 | 0.626 |
| | 768 | 3 | 96 | 0.257 | 0.663 | 0.677 |
| | 1664 | 6.5 | 208 | 0.404 | 0.696 | **0.702** |
| Proposed WNPQ Fast-VVLAD | 128 × 13 | 6.43 | 416 | 0.440 | 0.713 | 0.661 |
| WNPQ Multi-VLAD | 128 × 14 | 14 | 448 | **0.449** | **0.769** | 0.652 |

complexity and higher descriptor storage size[4].

**Holidays:** For the Holidays dataset, the quantized Fast-VDCNN maintains its mAP gain on query ROI over the quantized 128-D and 512-D CNN feature descriptors, while the descriptor storage has been reduced by a factor of 16 compared to its unquantized counterpart. In addition, the Fast-VDCNN still performs better than quantized Multi-CNN on whole image queries, with an mAP gain of 4%.

### *E. Further Improvements on Whole-image Search*

The experimental results of the previous section show that Fast-VVLAD and Fast-VDCNN clearly outperform their counterparts for ROI image search, while being competitive for whole-image search. The performance on whole image queries is primarily controlled by the dimension of the level-0 (whole image) component. For experiments in the previous section, we set the dimension uniformly across all components of the Voronoi-based descriptor, i.e., 128-D descriptor per cell. As a result, mAP for the Voronoi-based descriptors on whole images is comparable to that of the 128-D reference descriptors. One option to tailor performance towards whole image queries or smaller ROI queries is by tapering the dimension across levels; we leave this as a topic for future study.

TABLE V
COMPLEXITY AND MAP RESULTS FOR THE HOLIDAYS DATASET WITH WNPQ [36].

| | $D_{\text{tot}}$ | Matching Complexity | Descriptor Storage (bytes) | Query ROI | Whole Image |
|---|---|---|---|---|---|
| WNPQ CNN (Layer 13) | 128 | 1 | 32 | 0.248 | 0.674 |
| | 512 | 4 | 128 | 0.280 | **0.706** |
| Proposed WNPQ Fast-VDCNN | 128 × 13 | 6.13 | 416 | 0.550 | 0.684 |
| WNPQ Multi-CNN | 128 × 14 | 14 | 448 | **0.603** | 0.656 |

Another approach to boost performance for whole image queries is by accounting for multiple scales in *both* the query and dataset images. In other words, rather than applying Voronoi partitioning *only* on the dataset images, we can also apply Voronoi partitioning *on the query image* over multiple levels and submit each of the $V_{\text{tot}}$ query partitions as a subquery. Notably, using the Fast-VDCNN for the dataset image encodings, each subquery is matched only against representative cells in the dataset images (i.e., between 4 to 7 cells), which are determined by the adaptive search proposed in Section III-B. The inner product between the original query image and a dataset image is taken as the average inner product over all $V_{\text{tot}}$ subqueries. While this incurs linear increase in the search complexity (by $V_{\text{tot}}$), this scales better than the quadratic search complexity achieved by Carlsson *et al.* [30], [32], where exhaustive search amongst all subqueries is carried out.

Table VI compares the retrieval performance of the proposed Fast-VDCNN descriptor against the current state-of-the-art on the Holidays dataset that use networks pre-trained on ImageNet. The Fast-VDCNN descriptor is generated under the configuration of Section V-B, albeit now also partitioning the queries with $V_1 = V_2 = 2$ and resizing image partitions to $448 \times 448$. Beyond benchmarking against the grid-based spatial search method of Carlsson *et al.* [32], we also compare our results with the recently-proposed CNN+VLAD [33], CKN-mix [45], the hybrid FV-NN approach of Peronnin *et al.* [27], as well as lower-dimensional but more computationally-intensive proposals [5] that perform competitively [65]–[67]. Evidently, the additional scale and location invariance provided by the Voronoi partitioning leads to the proposed Fast-VDCNN achieving competitive performance to other CNN

[4]It is worth noting that, given we use a single PQ codebook for quantizing all cell components of Fast-VVLAD and Fast-VDCNN, all quantized based systems have a $Z' \times Z' \times M$ bit cost for storing the look-up tables. This means that, for example, although the 1664-D VLAD offers a lower storage size to Fast-VVLAD, there is an additional 1.7MB cost to store the look-up table, versus 262kB for Fast-VVLAD. However, as mentioned previously, the significance of this additional storage cost diminishes when increasing the test dataset size.

[5]In particular: the SPoC descriptor [65] offers the best performance to dimensionality, but utilizes a deeper and a more computationally heavy CNN (144M parameters vs 76M parameters for our architecture) and a larger image input size, the R-MAC based descriptor uses Siamese learning with supervised whitening, and NetVLAD requires additional processing (soft assignment and normalizations within the NetVLAD layer) to encode VLAD from the network activations. On the contrary, under the chosen configuration, the proposed Fast-VDCNN approach allocates only 128 dimensions per cell and accesses between 4 to 7 cells for each image subquery.

TABLE VI
COMPARISON OF WHOLE-IMAGE RETRIEVAL PERFORMANCE (MAP) WITH STATE-OF-THE-ART FOR THE HOLIDAYS DATASET [36]. THE PROPOSED APPROACH ALLOCATES 128 DIMENSIONS PER PARTITION CELL.

| | $D_{\text{tot}}$ | Whole Image |
|---|---|---|
| Proposed Fast-VDCNN | 1.66K (128) | 0.821 |
| FV-NN (Peronnin *et al.*) [27] | 4K | **0.835** |
| CNN + VLAD [33] | 2K | 0.802 |
| CNN (Carlsson *et al.*) [32] | 4K-15K | 0.769 |
| CKN-mix [45] | 4K | 0.829 |
| SPoC (w/o center prior) [65] | 256 | 0.802 |
| R-MAC [66] | 512 | 0.825 |
| NetVLAD [67] | 256 | 0.799 |

derived frameworks and hybrid variants, without manually rotating the images, and despite the fact that our feature descriptor is built directly from a pre-trained network and incurs modest computational and storage requirements.

## VI. CONCLUSION

We proposed a novel descriptor design, termed Voronoi-based encoding, for region-of-interest image retrieval. We have shown how VE could fit into a practical ROI-based retrieval system via the proposed fast search, memory-efficient design, product-quantization based lossy compression techniques, and robust similarity scoring mechanisms. We test retrieval performance on two datasets, using VLAD and a deep CNN as our descriptor basis. Our results show that our approach is descriptor agnostic; the proposed Fast-VVLAD and Fast-VDCNN maintain competitive retrieval performance over diverse ROI queries on two datasets and significantly improve on the retrieval performance (or implementation efficiency) of their respective descriptor variants with a grid spatial search, when dealing with smaller ROI queries. Moreover, improved geometric invariance results in competitive retrieval performance to the current state-of-the-art on whole image queries.

## APPENDIX A
## LEVEL PROJECTION FOR VE STORAGE COMPACTION

In order to decrease the storage requirements for unquantized Voronoi-based encoded (VE) representations, the descriptor over two constituent cells $x$ and $y$ (i.e., spatially-neighboring cells belonging to the same cell of the upper level), can be approximated as:

$$\widetilde{\phi}_{x\cup y} = \widetilde{\phi}_x + \widetilde{\phi}_y. \tag{14}$$

This holds because both PCA and whitening are linear mappings, therefore, if we do not consider the vector truncation and subsequent $L_2$ normalization of the individual cell vectors, the additivity property holds in the projected domain as well. Given that directionality is preserved under normalization, (14) provides an approximation to the normalized encoding computed directly over the two cells. Therefore, we can trade-off computation for memory by solely storing the last-level PCA-projected descriptors (level $L-1$) and computing all other cell encodings for all lower levels at runtime via repetitive application of (14) amongst constituent cells and renormalizing before carrying out the similarity measurement of (2). This is an appealing proposition for practical systems because vectorized addition and scaling for normalization is extremely inexpensive in modern SIMD-based architectures. As such, this approach requires storing only $\prod_{l=1}^{L-1} V_l$ cell descriptors, instead of $V_{\text{tot}}$ cell vectors. Naturally, there is a dependency on the projection error, which will evidently be greater with less dimensions retained post-PCA.

We can integrate product quantization with a modified level projection for quantized VE storage compaction. As before, we only store the last-level (quantized) descriptors offline. However, as the inner product satisfies the distributive law, we should now directly approximate the inner product between a query encoding and a level $l-1$ cell descriptor as an $L_1$ normalized summation:

$$S_{\text{ROI},l-1} = \frac{1}{V_l}\sum_{i=1}^{V_l} S_{\text{ROI},l_i\in l-1} \tag{15}$$

where each inner product is read from a look up table.

## APPENDIX B
## PROOF OF PROPOSITION 1

*Proof:* In order to optimize the bit allocation to the various descriptions (subspaces), we optimize the rate distortion expression:

$$R(E) = \min_{\sum E_m = E} \sum_{m=1}^{M} \ln \frac{|\mathbf{\Sigma}_m|}{E_m} \tag{16}$$

where $E_m$ is the product of the subcomponent distortions, $|\mathbf{\Sigma}_m|$ is the determinant of the covariance matrix $\mathbf{\Sigma}_m$ and $E$ is the overall distortion value. The minimum rate for given $E$ is derived when all distortions $E_m$ are equal, i.e., $E_m = E/M$.

Using results derived from rate distortion theory [68], $E_m$ for the $m$-th subspace can be approximated by:

$$E_m \approx \ |\mathbf{\Sigma}_m| \prod_{i=1}^{D'} h_{im} 2^{-2b_{im}} = |\mathbf{\Sigma}_m|\, h'_m 2^{-2B'_m} \tag{17}$$

where $h_{im} = \frac{1}{12}\left\{\int_{-\infty}^{\infty} f_{im}(x)\ dx\right\}^3$ is a variable determined by the univariate Gaussian of the normalized components, $f_{im}(x)$, and $b_{im}$ is the average number of bits encoded per dimension. Due to the independence property, the product of $h_{im}$ in the $m$-th subspace yields the variable $h'_m$, which is now determined by the multivariate Gaussian distribution for the normalized subspace random vectors. This distribution is independent of subspace, and, as such, $h'_m$ is constant for all $m$. Similarly, if the size of the bit encoding and block dimension $D'$ is fixed per subspace, then $B'_m$ is a constant for all $m$. For $E_m$ to be equal for all $m$, $|\mathbf{\Sigma}_m|$ must be constant, independent of subspace. ■

## References


[1] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Rec. (CVPR)*, 2012, pp. 2911–2918.

[2] S. Bai, S. Sun, X. Bai, Z. Zhang, and Q. Tian, "Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity," in *European Conference on Computer Vision*. Springer, 2016, pp. 592–608.

[3] A. Chadha and Y. Andreopoulos, "Region-of-interest retrieval in large image datasets with voronoi vlad," in *Int. Conf. on Computer Vision Syst.* Springer, 2015, pp. 218–227.

[4] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013.

[5] X. Xu, W. Geng, R. Ju, Y. Yang, T. Ren, and G. Wu, "Obsir: Object-based stereo image retrieval," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[6] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. of the IEEE Int. Conf. on Comput. Vis.*, 2015, pp. 1116–1124.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[8] V. Spiliotopoulos *et al.*, "Quantization effect on vlsi implementations for the 9/7 dwt filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 2001, ICASSP'01*, vol. 2. IEEE, 2001, pp. 1197–1200.

[9] Y. Andreopoulos and M. van der Schaar, "Incremental refinement of computation for the discrete wavelet transform," *IEEE Trans. on Signal Process.*, vol. 56, no. 1, pp. 140–157, 2008.

[10] Y. Andreopoulos *et al.*, "A new method for complete-to-overcomplete discrete wavelet transforms," in *Proc. 14th IEEE Int. Conf. on Digital Signal Process., DSP 2002*, vol. 2. IEEE, 2002, pp. 501–504.

[11] ——, "A local wavelet transform implementation versus an optimal row-column algorithm for the 2d multilevel decomposition," in *Proc. IEEE Int. Conf. on Image Process., ICIP 2001*, vol. 3. IEEE, 2001, pp. 330–333.

[12] N. Kontorinis *et al.*, "Statistical framework for video decoding complexity modeling and prediction," *IEEE Trans. on Circ. and Syst. for Video Technol.*, vol. 19, no. 7, pp. 1000–1013, 2009.

[13] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. Ninth IEEE Int. Conf. on Comput. Vis.* IEEE, 2003, pp. 1470–1477.

[14] J. Philbin *et al.*, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Rec.*, 2008, pp. 1–8.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2007, pp. 1–8.

[16] I. González-Díaz, C. E. Baz-Hormigos, and F. Diaz-de Maria, "A generative model for concurrent image retrieval and ROI segmentation," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 169–183, 2014.

[17] Z. Zhong, J. Zhu, and C. Hoi, "Fast object retrieval using direct spatial matching," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1391–1397, 2015.

[18] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 648–659, 2015.

[19] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Rec.*, 2010, pp. 3304–3311.

[20] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2007.

[21] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2013, pp. 1578–1585.

[22] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1297–1308, 2015.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2014, pp. 580–587.

[24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[25] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2014, pp. 1725–1732.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] F. Perronnin and D. Larlus, "Fisher vectors meet neural networks: A hybrid classification architecture," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2015, pp. 3743–3752.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.* IEEE, 2009, pp. 248–255.

[29] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Europ. Conf. in Comput. Vis. (ECCV)*. Springer, 2014, pp. 584–599.

[30] H. Azizpour, A. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," 2014.

[31] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[32] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn. Workshops*, 2014, pp. 806–813.

[33] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Europ. Conf. in Comput. Vis. (ECCV)*. Springer, 2014, pp. 392–407.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[35] V. Chandrasekhar, J. Lin, O. Morère, H. Goh, and A. Veillard, "A practical guide to cnns and fisher vectors for image instance retrieval," *arXiv preprint arXiv:1508.02496*, 2015.

[36] H. Jegou *et al.*, "Hamming embedding and weak geometric consistency for large scale image search," in *Europ. Conf. in Comput. Vis.* Springer, 2008, pp. 304–317.

[37] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, vol. 2, 2003, pp. II–264.

[38] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, 2012.

[39] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.*, 2010, pp. 3384–3391.

[40] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening," in *Proc. Europ. Conf. in Comput. Vis. (ECCV)*. Springer, 2012, pp. 774–787.

[41] O. Chum and J. Matas, "Unsupervised discovery of co-occurrence in sparse high dimensional data," in *Proc. IEEE Int. Conf. on Comput. Vis. and Pat. Recogn. (CVPR)*. IEEE, 2010, pp. 3416–3423.

[42] S. Lazebnik *et al.*, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Rec.*, vol. 2, 2006, pp. 2169–2178.

[43] E. Mantziou, S. Papadopoulos, and Y. Kompatsiaris, "Large-scale semi-supervised learning by approximate Laplacian eigenmaps, VLAD and pyramids," in *14th Int. Workshop on Image Anal. for Mult. Interactive Services (WIAMIS)*. IEEE, 2013, pp. 1–4.

[44] R. Zhou, Q. Yuan, X. Gu, and D. Zhang, "Spatial pyramid VLAD," in *IEEE Vis. Comm. and Image Proc. Conf.* IEEE, 2014, pp. 342–345.

[45] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: an unsupervised approach," *International Journal of Computer Vision*, to appear.

[46] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011.

[47] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[48] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009, pp. 1753–1760.

[49] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.* IEEE, 2008, pp. 1–8.

[50] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu, "Complementary hashing for approximate nearest neighbor search," in *2011 IEEE Int. Conf. on Comput. Vis. (ICCV)*. IEEE, 2011, pp. 1631–1638.

[51] Y. Li, R. Wang, H. Liu, H. Jiang, S. Shan, and X. Chen, "Two birds, one stone: Jointly learning binary code for large-scale face image retrieval and attributes prediction," in *Proc. of the IEEE Int. Conf. on Comp. Vis.*, 2015, pp. 3819–3827.

[52] D. Song, W. Liu, R. Ji, D. A. Meyer, and J. R. Smith, "Top rank supervised binary coding for visual search," in *Proc. of the IEEE Int. Conf. on Comput. Vis.*, 2015, pp. 1922–1930.

[53] T. Ji, X. Liu, C. Deng, L. Huang, and B. Lang, "Query-adaptive hash code ranking for fast nearest neighbor search," in *Proc. of the ACM Int. Conf. on Multimedia*. ACM, 2014, pp. 1005–1008.

[54] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[55] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. of Comput. Vis.*, vol. 65, no. 1-2, pp. 43–72, 2005.

[56] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. Europ. Conf. in Comput. Vis. (ECCV)*. Springer, 2002, pp. 128–142.

[57] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Europ. Conf. in Comput. Vis.* Springer, 2006, pp. 430–443.

[58] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[59] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[60] J. Brandt, "Transform coding for fast approximate nearest neighbor search in high dimensions," in *Proc. IEEE Int. Conf. on Comput. Vis. and Patt. Recogn.* IEEE, 2010, pp. 1815–1822.

[61] T. Ge, K. He, Q. Ke, and J. Sun, "Optimized product quantization," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 36, no. 4, pp. 744–755, 2014.

[62] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[63] E. Spyromitros-Xioufis, S. Papadopoulos, I. Y. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "A comprehensive study over VLAD and product quantization in large-scale image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1713–1728, 2014.

[64] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *IEEE Int. Conf. on Comput. Vis. Workshops (ICCVW)*. IEEE, 2013, pp. 554–561.

[65] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proc. of the IEEE Int. Conf. on Comput. Vis.*, 2015, pp. 1269–1277.

[66] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," *arXiv preprint arXiv:1604.02426*, 2016.

[67] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *arXiv preprint arXiv:1511.07247*, 2015.

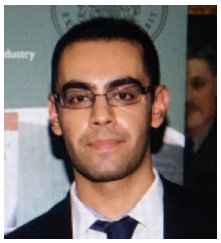
[68] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[69] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[70] O. Chum *et al.*, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE Int. Conf. on Comput. Vis.*, 2007, pp. 1–8.

[71] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2, pp. 4–29, 1984.

[72] G. Shakhnarovich, T. Darrell, and P. Indyk, "Nearest-neighbor methods in learning and vision: Theory and practice, chapter 3," 2006.

**Aaron Chadha** obtained a BA and MEng in Information and Computer Engineering from the University of Cambridge, graduating with a Distinction. He joined the Department of Electronic and Electrical Engineering at University College London in 2014, where he is currently working towards a PhD in the area of image/video classification and retrieval under the supervision of Dr Yiannis Andreopoulos. He was awarded with an Industrial Fellowship from the Royal Commission of the Exhibition of 1851 in 2016.

**Yiannis Andreopoulos** (M'00–SM'14) obtained the Electrical Engineering Diploma and an MSc in Signal and Image Processing Systems from the University of Patras, Greece, and the PhD in Applied Sciences from the Vrije Universiteit Brussel, Belgium. He is Reader in Data and Signal Processing Systems in the Department of Electronic and Electrical Engineering of University College London (U.K.). His research interests are in wireless sensor networks, error-tolerant computing and multimedia systems. He received the 2007 Most-Cited Paper Award from the Elsevier EURASIP Signal Processing: Image Communication journal and a best paper award from the 2009 IEEE Workshop on Signal Processing Systems. He was Special Sessions Co-Chair of the 10th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2009) and Programme Co-Chair of the 18th International Conference on Multimedia Modeling (MMM 2012) and the 9th International Conference on Body Area Networks (BODYNETS 2014). He has been an Associate Editor of the IEEE Transactions on Multimedia, the IEEE Signal Processing Letters and Image and Vision Computing (Elsevier).