Higher-order Occurrence Pooling on Mid- and Low-level Features: Visual Concept Detection

Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, Krystian Mikolajczyk

▶ **To cite this version:**

Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, Krystian Mikolajczyk. Higher-order Occurrence Pooling on Mid- and Low-level Features: Visual Concept Detection. [Technical Report] 2013, pp.20. hal-00922524

HAL Id: hal-00922524

https://inria.hal.science/hal-00922524

Submitted on 27 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Higher-order Occurrence Pooling on Mid- and Low-level Features: Visual Concept Detection

Piotr Koniusz, Fei Yan, Philippe-Henri Gosselin, Krystian Mikolajczyk

Abstract—In object recognition, the Bag-of-Words model assumes: i) extraction of local descriptors from images, ii) embedding these descriptors by a coder to a given visual vocabulary space which results in so-called mid-level features, iii) extracting statistics from mid-level features with a pooling operator that aggregates occurrences of visual words in images into so-called signatures. As the last step aggregates only occurrences of visual words, it is called as First-order Occurrence Pooling. This paper investigates higher-order approaches. We propose to aggregate over co-occurrences of visual words, derive Bag-of-Words with Second- and Higher-order Occurrence Pooling based on linearisation of so-called Minor Polynomial Kernel, and extend this model to work with adequate pooling operators. For bi- and multi-modal coding, a novel higher-order fusion is derived. We show that the well-known Spatial Pyramid Matching and related methods constitute its special cases. Moreover, we propose Third-order Occurrence Pooling directly on local image descriptors and a novel pooling operator that removes undesired correlation from the image signatures. Finally, Uni- and Bi-modal First-, Second-, and Third-order Occurrence Pooling are evaluated given various coders and pooling operators. The proposed methods are compared to other approaches (e.g. Fisher Vector Encoding) in the same testbed and attain state-of-the-art results.

Index Terms—Bag-of-Words, Mid-level features, First-order, Second-order, Co-occurrence, Pooling Operator, Sparse Coding

1 Introduction

Bag-of-Words [1], [2] (BoW) is a popular approach which transforms local image descriptors [3], [4], [5] into image representations that are used in retrieval and classification [1], [2]. To date, a number of its variants have been developed and reported to produce state-of-the-art results: Kernel Codebook [6], [7], [8], [9] a.k.a. Soft Assignment and Visual Word Uncertainty, Approximate Locality-constrained Soft Assignment [10], [11], Sparse Coding [12], [13], Local Coordinate Coding [14], Approximate Locality-constrained Linear Coding [15], and Laplacian Sparse Coding [16]. We refer to this group as standard BoW. Recently, Super Vector Coding [17], Vector of Locally Aggregated Descriptors [18], Fisher Vector Encoding [19], [20], and Vector of Locally Aggregated Tensors [21] have emerged as better performers compared to *e.g.* Sparse Coding [13]. The main hallmarks of this second group of methods, in contrast to standard BoW, are: i) encoding descriptors with respect to the centres of clusters that these descriptors are assigned to, ii) extracting second-order statistics from mid-level features to complement the first-order cues, iii) pooling that benefits from Power Normalisation [22], [20] which counteracts so-called *burstiness* [23], [11].

Various models of BoW have been evaluated in several publications [24], [25], [26], [27], [28], [29], [11]. A recent review of coding schemes [25] includes Hard Assignment, Soft Assignment, Approximate Locality-constrained Linear Coding, Super Vector Coding, and Fisher Vector Encoding. Moreover, the role played by pooling during the generation of image signatures has been studied [28], [29], [11] leading to promising improvements in object category recognition. A detailed comparison of BoW [11] shows that the choice of pooling influences substantially the classification performance of various coders. Their evaluations highlight that the standard BoW approaches perform noticeably worse compared to the second group of methods distinguished above, *e.g.* Fisher Vector Encoding.

To date, the pooling step employed by standard BoW aggregates only occurrences of visual words in the mid-level features (First-order Occurrence Pooling). Max-pooling [13] is often used to perform the aggregation whilst the coding step varies [7], [13], [14]. In this paper, we study the standard BoW model according to the listed above hallmarks and propose changes that make it outperform other approaches. The analysis of First-, Second-, and Third-order Occurrence Pooling in the BoW model constitutes the main contribution of this work. In more detail:

1) We propose to aggregate co-occurrences rather than occurrences of visual words in mid-level features to improve the expressiveness of a visual dictionary. We call this Second- and Higher-order Occurrence Pooling and derive it by linearisation of so-called Minor Polynomial Kernel. A generalisation from Average to Max-pooling is also proposed.

- *F. Yan, K. Mikolajczyk are at Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford, UK, GU2 7XH.*
- *P. Koniusz graduated from CVSSP, he is currently with INRIA LEAR, Rhône-Alpes, France, 38330. See http://claret.wikidot.com*
- *P. H. Gosselin works at INRIA TexMex, Rennes, France, 35042.*

2) Evaluations of First-, Second-, and Third-order Occurrence Pooling are performed on Sparse Coding (SC), Approximate Locality-constrained Linear Coding (LLC), and Approximate Locality-constrained Soft Assignment (LcSA). In most of experiments, we resign from Spatial Pyramid Matching [30], [13] in favour of Spatial Coordinate Coding introduced in [31], evaluated in [11], and employed recently by Fisher Vector Encoding [32].
3) For the aggregation step, we employ a pooling operator called $@n$ that reduces the coding noise in BoW [11]. Then, we compare it to Max-pooling [13] and Analytical pooling such as Power Normalisation [20] (Gamma) and *theoretical expectation of Max-pooling* [29] (MaxExp). For the best performance, we use the pooling variants that account for the interdependence of overlapping descriptors [11].
4) A linearisation for fusing bi- and multi-modal cues in Second- and Higher-order Occurrence Pooling is proposed, evaluated on the grey and colour mid-level features, and compared to other fusions.
5) For further evaluation of the proposed fusion, a novel residual descriptor, used as an auxiliary cue, is developed to exploit the quantisation error of SC, LLC, and LcSA coding. Moreover, Spatial [30], [13] and Dominant Angle Pyramid Matching [31], [11] are shown as special cases of our second-order fusion on standard BoW. Thus, we attribute their robustness to the implicit use of second-order cues.
6) In contrast to the coder-based methods, we also propose Third-order Occurrence Pooling on the low-level local image descriptors and a novel pooling operator based on Higher Order Singular Value Decomposition [33], [34] and Power Normalisation to counteract so-called *correlated burstiness*.
7) We compare our methods in the common testbed to Fisher Vector Encoding [20], [32] (FV), Vector of Locally Aggregated Tensors [21] (VLAT), First-order Occurrence based Spatial Coordinate Coding [31], [11] (SCC), Spatial Pyramid Matching [30], [13] (SPM), Dominant Angle Pyramid Matching [31], [11] (DoPM), and Second-order Pooling from [35]. We attain state-of-the-art results on Pascal VOC07, Caltech101, Flower102, ImageCLEF11, 15 Scenes, and Pascal VOC10 Action Recognition datasets.

Our method is somewhat inspired by Vector of Locally Aggregated Tensors [21] in terms of how we model co-occurrences. However, we distinguish the coding and pooling steps in the proposed model to incorporate arbitrary coders and pooling operators. This also differs from a recently proposed Second-order Pooling applied in the problem of semantic segmentation [35]: i) we perform pooling on the mid-level features to preserve the data manifold learned during the coding step whilst the latter method acts on the low-level descriptors, ii) we provide a derivation of Second- and Higher-order Occurrence Pooling, iii) we use a pooling operator developed for BoW. Another take on building rich statistics from the mid-level features are 2D histogram representations [36]. They employ arbitrary statistics to each particular type of coder, *e.g.* sum between pairs of mid-level coefficients. In contrast, our approach results from the analytical solution to the kernel linearisation problems.

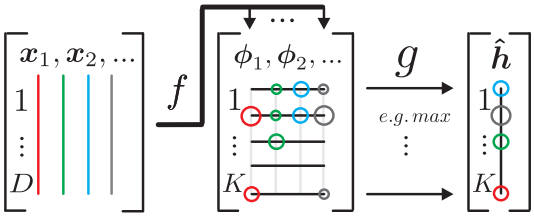


Fig. 1. Overview of Bag-of-Words. The local descriptors x are extracted from an image and coded by f that operates on columns. Circles of various sizes illustrate values of mid-level coefficients. Pooling g aggregates visual words from the mid-level features ϕ along rows.

The reminder of section 1 introduces the standard model of Bag-of-Words in section 1.1. The coders and pooling operators used in this study are presented in sections 1.2 and 1.3, respectively. The rest of this paper is organised as follows. Section 2 describes Uni-modal BoW with Higher-order Occurrence Pooling. Section 3 proposes Bi- and Multi-modal BoW with Second- and Higher-order Occurrence Pooling and its several extensions. Section 4 explains Third-order Occurrence Pooling on the low-level descriptors. Section 5 details our experiments. Section 6 draws the conclusions.

1.1 Bag-of-Words Model

Let us denote the descriptor vectors as $\boldsymbol{x}_n \in \mathbb{R}^D$ such that $n = 1, ..., N$, where N is the total descriptor cardinality for the entire image set $\mathcal{I}$, and D is the descriptor dimensionality. Given any image $i \in \mathcal{I}$, $\mathcal{N}^i$ denotes a set of its descriptor indices. We drop the superscript for simplicity and use $\mathcal{N}$. Therefore, $\{\boldsymbol{x}_n\}_{n\in\mathcal{N}}$ denotes a set of descriptors for an image $i \in \mathcal{I}$. Next, we assume $k = 1, ..., K$ visual appearance prototypes $\boldsymbol{m}_k \in \mathbb{R}^D$ a.k.a. visual vocabulary, words, centres, atoms, and anchors. We form a dictionary $\boldsymbol{\mathcal{M}} = \{\boldsymbol{m}_k\}_{k=1}^K$, where $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{D\times K}$ can also be seen as a matrix formed by visual words as its columns. Figure 1 gives a simple illustration of Bag-of-Words. Following the formalism of [28], [11], we express the standard BoW approaches (indicated in the introduction) as a combination of the mid-level coding and pooling steps, followed by the ℓ_2 norm normalisation:

$$\boldsymbol{\phi}_n = f(\boldsymbol{x}_n, \boldsymbol{\mathcal{M}}), \quad \forall n \in \mathcal{N} \tag{1}$$

$$\hat{h}_k = g\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) \tag{2}$$

$$\boldsymbol{h} = \hat{\boldsymbol{h}}/\|\hat{\boldsymbol{h}}\|_2 \tag{3}$$

Equation (1) represents a mid-level feature mapping $f : \mathbb{R}^D \to \mathbb{R}^K$, *e.g.* Sparse Coding. It quantifies the

image content in terms of visual vocabulary $\boldsymbol{\mathcal{M}}$. Each descriptor $\boldsymbol{x}_n$ is embedded into the visual vocabulary space resulting in mid-level features $\boldsymbol{\phi}_n \in \mathbb{R}^K$.

Equation (2) represents the pooling operation, *e.g.* Average or Max-pooling. The role of g is to aggregate occurrences of visual words in mid-level features, and therefore in an image. Formally, function $g : \mathbb{R}^{|\mathcal{N}|} \rightarrow \mathbb{R}$ takes all mid-level feature coefficients ϕ_{kn} for visual word $\boldsymbol{m}_k$ given image i to produce a k^{th} coefficient in vector $\hat{\boldsymbol{h}} \in \mathbb{R}^K$. Note that $\boldsymbol{\phi}_n$ denotes an n^{th} mid-level feature vector while ϕ_{kn} denotes its k^{th} coefficient. We do not include pooling over cells of Spatial Pyramid Matching to maintain simplicity. SPM compatible formulation can be found in [11].

Equation (3) normalises signature $\hat{\boldsymbol{h}}$. Then, signatures $\boldsymbol{h}_i, \boldsymbol{h}_j \in \mathbb{R}^K$ for $i, j \in \mathcal{I}$ form a linear kernel $Ker_{ij} = (\boldsymbol{h}_i)^T \cdot \boldsymbol{h}_j$ used by a classifier, *e.g.* KDA [37].

This model of BoW assumes First-order Occurrence Pooling and often uses SC, LLC, and LcSA coders that will be now described. The same model can accommodate FV and VLAT after minor changes.

1.2 Mid-level Coders

Below is the introduction of the mid-level coders f used in this work. For clarity, we abbreviate $\boldsymbol{x}_n$ to $\boldsymbol{x}$ and $\boldsymbol{\phi}_n$ to $\boldsymbol{\phi}$ where possible.

Sparse Coding [12], [13] expresses each descriptor $\boldsymbol{x}$ as a sparse linear combination of the visual words contained in $\boldsymbol{\mathcal{M}}$. This is achieved by optimising the following cost function with respect to $\boldsymbol{\phi}$:

$$\begin{aligned} &\boldsymbol{\phi} = \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \boldsymbol{\mathcal{M}}\bar{\boldsymbol{\phi}} \right\|_2^2 + \alpha \|\bar{\boldsymbol{\phi}}\|_1 \\ &s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0 \end{aligned} \tag{4}$$

A low number of non-zero coefficients in $\boldsymbol{\phi}$, referred to as sparsity, is induced with the ℓ_1 norm and adjusted by constant α. We impose a non-negative constraint on $\boldsymbol{\phi}$ for compatibility with Analytical pooling.

Approximate Locality-constrained Linear Coding addresses the non-locality phenomenon [15] that can occur in SC and is formulated as follows:

$$\begin{aligned} &\boldsymbol{\phi}^* = \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \boldsymbol{\mathcal{M}}\left(\boldsymbol{x}, l\right) \bar{\boldsymbol{\phi}} \right\|_2^2 \\ &s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0,\ \ \mathbf{1}^T \bar{\boldsymbol{\phi}} = 1 \end{aligned} \tag{5}$$

Descriptor $\boldsymbol{x}$ is coded with its l-nearest neighbours found in dictionary $\boldsymbol{\mathcal{M}}$ by NN search. For every $\boldsymbol{x}$, a new compact dictionary is formed and used: $\boldsymbol{\mathcal{M}}\left(\boldsymbol{x}, l\right) = NN\left(\boldsymbol{x}, l, \boldsymbol{\mathcal{M}}\right) \in \mathbb{R}^{D \times l}$. Constant $l \ll K$ influences how localised the coding becomes. Lastly, the resulting $\boldsymbol{\phi}^* \in \mathbb{R}^l$ of length l is re-projected into the full length nullified mid-level feature $\boldsymbol{\phi} \in \mathbb{R}^K$. For every atom of index $i = 1, ..., l$ in $\boldsymbol{\mathcal{M}}\left(\boldsymbol{x}, l\right)$, we set $\boldsymbol{\phi}_{i'} = \boldsymbol{\phi}_i^*$ based on index $1 \leq i' \leq K$ of corresponding atoms in $\boldsymbol{\mathcal{M}}$ known from NN search.

Approximate Locality-constrained Soft Assignment [10] is derived from Mixture of Gaussians [38] with parameters $\theta = (\theta_1, ..., \theta_K) = ((\boldsymbol{m}_1, \sigma), ..., (\boldsymbol{m}_K, \sigma))$ and mixing probabilities $w_1 = ... = w_K = 1$. The component membership probability is used as a coder [11]:

$$\phi_k = \begin{cases} \frac{G(\boldsymbol{x}; \boldsymbol{m}_k, \sigma)}{\sum_{\boldsymbol{m}' \in \boldsymbol{\mathcal{M}}(\boldsymbol{x}, l)} G(\boldsymbol{x}; \boldsymbol{m}', \sigma)} & \text{if } \boldsymbol{m}_k \in \boldsymbol{\mathcal{M}}\left(\boldsymbol{x}, l\right) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Note that ϕ_k is computed from the l-nearest Gaussian components of $\boldsymbol{x}$ that are found in dictionary $\boldsymbol{\mathcal{M}}$ by NN search. This prevents non-locality of coding [15]. Parameter $l \ll K$ dictates how localised the solution is, σ is the smoothing factor of Gaussian kernel G [7].

Fisher Vector Encoding is used in this work for comparison. The coding step can be isolated from its common formulation given in [19], [20]. FV uses Mixture of Gaussians [38] as a dictionary. K components $\theta = (\theta_1, ..., \theta_K) = ((w_1, \boldsymbol{m}_1, \boldsymbol{\sigma}_1), ..., (w_K, \boldsymbol{m}_K, \boldsymbol{\sigma}_K))$ are used, each consisting of mixing probability, mean, and on-diagonal standard deviation. The first and second order statistics $\boldsymbol{\phi}_k, \boldsymbol{\psi}_k \in \mathbb{R}^D$ are isolated:

$$\boldsymbol{\phi}_k = (\boldsymbol{x} - \boldsymbol{m}_k)/\boldsymbol{\sigma}_k,\ \ \boldsymbol{\psi}_k = \boldsymbol{\phi}_k^2 - 1 \tag{7}$$

Furthermore, concatenation of per-cluster statistics $\bar{\boldsymbol{\phi}}_k \in \mathbb{R}^{2D}$ forms the mid-level feature $\boldsymbol{\phi} \in \mathbb{R}^{2KD}$:

$$\boldsymbol{\phi} = \left[\bar{\boldsymbol{\phi}}_1^T, ..., \bar{\boldsymbol{\phi}}_K^T\right]^T,\ \ \bar{\boldsymbol{\phi}}_k = \frac{p\left(\boldsymbol{m}_k | \boldsymbol{x}, \theta\right)}{\sqrt{w_k}} \begin{bmatrix} \boldsymbol{\phi}_k \\ \boldsymbol{\psi}_k / \sqrt{2} \end{bmatrix} \tag{8}$$

The expression $p\left(\boldsymbol{m}_k | \boldsymbol{x}, \theta\right)$ is the membership probability of mean $\boldsymbol{m}_k$ being selected given descriptor $\boldsymbol{x}$ and parameters θ. Note that the above formulation is compatible with equation (1) except for $\boldsymbol{\phi}$ to be $2KD$ rather than K long. Moreover, $\boldsymbol{\phi}$ also contains second-order statistics unlike codes of SC, LLC, and LcSA.

Vector of Locally Aggregated Tensors [21] also has a distinct coding step yielding the first and second order statistics $\boldsymbol{\phi}_k \in \mathbb{R}^D$ and $\boldsymbol{\Psi}_k \in \mathbb{R}^{D \times D}$ per cluster:

$$\boldsymbol{\phi}_k = \boldsymbol{x} - \boldsymbol{m}_k,\ \ \boldsymbol{\Psi}_k = \boldsymbol{\phi}_k {\boldsymbol{\phi}_k}^T - \boldsymbol{C}_k \tag{9}$$

However, only the second order matrices $\boldsymbol{\Psi}_k$ are deployed to form the mid-level features after normalisation with per-cluster covariance matrices $\boldsymbol{C}_k$. As $\boldsymbol{\Psi}_k$ are symmetric, the upper triangles and diagonals are extracted and unfolded into vectors with operator $u_:$, and concatenated for all k-means clusters $k = 1, ..., K$:

$$\boldsymbol{\phi} = \left[u_:(\boldsymbol{\Psi}_1)^T, ..., u_:(\boldsymbol{\Psi}_K)^T\right]^T \tag{10}$$

Note that this formulation is compatible with equation (1) except for $\boldsymbol{\phi}$ to be $KD(D+1)/2$ rather than K long.

1.3 Pooling Operators

BoW aggregates occurrences of visual words represented by the coefficients of mid-level feature vectors with a pooling operator g given by equation (2). The operators used in this work are described below.

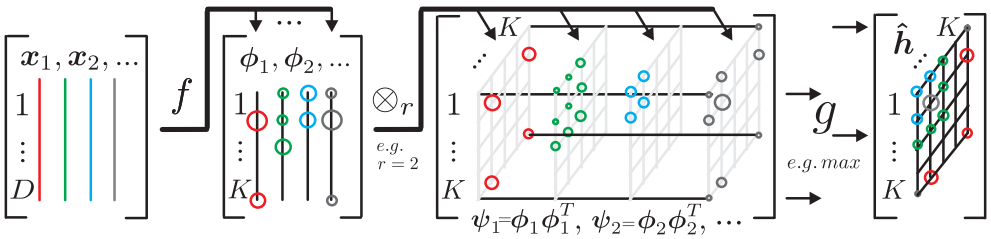


Fig. 2. Uni-modal Bag-of-Words with Second-order Occurrence Pooling (order $r=2$). The local descriptors $\boldsymbol{x}$ are extracted from an image and coded by f that operates on columns. Circles of various sizes illustrate values of mid-level coefficients. Self-tensor product $\otimes_r$ computes co-occurrences of visual words for every mid-level feature $\boldsymbol{\phi}$. Pooling g aggregates visual words from the co-occurrence matrices $\boldsymbol{\psi}$ along the direction of stacking. For the purpose of illustration, the unfolding operator $u_:$ from equation (16) is not used.

Average pooling counts the number of descriptor assignments per cluster k and normalises such counts by the number of descriptors in the image [2]. It works with SC, LLC, LcSA, FV, VLAT and is defined as:

$$\hat{h}_k = \operatorname{avg}\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) = \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\phi_{kn} \tag{11}$$

Max-pooling [13], [28], [29], [10] selects the largest value from $|\mathcal{N}|$ mid-level feature coefficients responding to visual word $\boldsymbol{m}_k$. This is repeated for $k=1,...,K$:

$$\hat{h}_k = \max\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) = \max\left(\phi_{kn^{(1)}}, \phi_{kn^{(2)}}, ...\right) \tag{12}$$

To detect occurrences of visual words, Max-pooling is often combined with SC, LLC, and LcSA coders. It is not applicable to FV or VLAT, as their mid-level feature coefficients do not represent visual words.

MaxExp represents a *theoretical expectation of Max-pooling* [29] inspired by a statistical model. The mid-level feature coefficients for a given $\boldsymbol{m}_k$ are presumed to be drawn at random from Bernoulli distribution under the i.i.d. assumption. Binomial distribution dictates that, given exactly $\bar{N}=|\mathcal{N}|$ trials, the probability of *at least one visual word* $\boldsymbol{m}_k$ *present in image* i is:

$$\hat{h}_k = 1-(1-h_k^*)^{\bar{N}}, \quad h_k^* = \operatorname{avg}\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) \tag{13}$$

Moreover, we generalised this operator to account for the feature interdependence [11]. As the degree of statistical dependence between features is unknown, parameter $\bar{N}\leq|\mathcal{N}|$ has to be found by cross-validation. MaxExp is typically used with SC, LLC, and LcSA as constraint $0\leq h_k^*\leq 1$ does not hold for FV or VLAT.

Power Normalisation a.k.a. Gamma [22], [20], [23] approximates the statistical model of MaxExp [11]. It is used by SC, LLC, LcSA, FV, VLAT, and defined as:

$$\hat{h}_k = \operatorname{sgn}(h_k^*)\,|h_k^*|^\gamma, \quad h_k^* = \operatorname{avg}\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) \tag{14}$$

The degree of statistical dependence between features is adjusted by $0<\gamma\leq 1$ found during cross-validation.

Improved pooling (@n) was designed to suppress the low values of mid-level feature coefficients that were considered as a noise and called *leakage* [11]. Given SC, LLC, and LcSA coders, *leakage* was shown to misrepresent chosen visual words. Moreover, the @n was shown to exploit the descriptor interdependence and led to consistent classification improvements. This operator is a trade-off between Max-pooling and an Analytical pooling, *e.g.* MaxExp is used in this work:

$$h_k^* = \operatorname{avg}\operatorname{srt}\left(\{\phi_{kn}\}_{n\in\mathcal{N}},@n\right) = \operatorname{avg}\left[\operatorname{srt}\left(\{\phi_{kn}\}_{n\in\mathcal{N}},@n\right)\right]$$
$$\hat{h}_k = 1-(1-h_k^*)^{\bar{N}} \tag{15}$$

The @n largest mid-level features are selected by partial sort algorithm srt and averaged by avg. Parameter $1\leq@n\leq|\mathcal{N}|$ adjusts the trade-off, while $\bar{N}$ remains as defined for MaxExp. The mid-level feature coefficients for any given $\boldsymbol{m}_k$ are presumed to be drawn at random from a Bernoulli distribution under the i.i.d. assumption. However, this is only an approximation as ϕ_{kn} are typically non-negative real numbers such that $0\leq\phi_{kn}\leq 1$. Note that the pool of the largest @n coefficients only is available. Binomial distribution dictates that, given exactly $\bar{N}=@n$ trials, equation (15) yields the probability of *at least one visual word* $\boldsymbol{m}_k$ *present in the* @n *largest mid-level feature coefficients*. Given that large ϕ_{kn} represent visual word $\boldsymbol{m}_k$ and the small ones are the noise, this formulation yields improved estimates. This does not hold for FV or VLAT as their small ϕ_{kn} are not related to leakage.

2 Uni-modal Bag-of-Words with Higher-Order Occurrence Pooling

Higher-order BoW is introduced below. Its derivation is given in sections 2.1 and 2.2. The benefits of Second-order Occurrence Pooling are detailed in section 2.3.

Bag-of-Words typically use First-order Occurrence Pooling with the coding and pooling operators from section 1. In contrast, FV and VLAT benefit from the second-order statistics. Thus, we equip BoW with the second- or higher-order statistics. Figure 2 illustrates the proposed BoW with Second-order Occurrence Pooling. First, we perform coding represented by equation (1), then embed the second- or higher-order statistics by replacing equation (2) with two steps:

$$\boldsymbol{\psi}_n = u_:\left(\otimes_r\boldsymbol{\phi}_n\right) \tag{16}$$
$$\hat{h}_k = g\left(\{\psi_{kn}\}_{n\in\mathcal{N}}\right) \tag{17}$$

Equation (16) represents self-tensor product $\otimes_r$ performed on every mid-level feature vector $\boldsymbol{\phi}_n$ resulting from f, where $r \geq 1$ is a chosen rank (or order). This is done in order to compute co-occurrences (or higher-order occurrences) of visual words in every mid-level feature. Given $r = 1$, the above formulation becomes reduced to the standard BoW as $\boldsymbol{\psi}_n = \boldsymbol{\phi}_n = \otimes_1(\boldsymbol{\phi}_n)$. Moreover, as the resulting $\otimes_{r>1}$ are symmetric, only non-redundant coefficients are retained and unfolded into vectors with operator $u_:$. Specifically, one can extract: i) the upper triangle and diagonal for $\otimes_2$, ii) the upper pyramid and diagonal plane for $\otimes_3$, iii) the upper simplex and diagonal hyperplane for $\otimes_{r\geq 3}$. The dimensionality of self-tensor product after removing repeated coefficients and unfolding is $K^{(r)} = \binom{K+r-1}{r}$.

Equation (17) performs pooling, as in equation (3). However, this time g aggregates co-occurrences or higher-order occurrences of visual words in mid-level features for $r \geq 2$. Formally, function $g : \mathbb{R}^{|\mathcal{N}|} \rightarrow \mathbb{R}$ takes k^{th} co-occurrence (or higher-order occurrence) coefficients ψ_{kn} for all $n \in \mathcal{N}$ from image i to produce a k^{th} coefficient in vector $\hat{\boldsymbol{h}} \in \mathbb{R}^{K^{(r)}}$, where $k = 1, ..., K^{(r)}$.

Lastly, the normalisation from equation (3) is applied to $\hat{\boldsymbol{h}}$. The resulting signatures $\boldsymbol{h}$ are of dimensionality $K^{(r)}$ depending only on the dictionary size K and rank r. In contrast, sizes of FV and VLAT signatures depend on K and D (descriptor dimensionality).

2.1 Linearisation of Minor Polynomial Kernel

BoW with Higher-order Occurrence Pooling can be derived analytically by performing the following steps: i) defining a kernel function on a pair of mid-level features, $\boldsymbol{\phi}$ and $\bar{\boldsymbol{\phi}}$, referred to as Minor Kernel, ii) summing over all pairs of mid-level features formed from a given pair of images, iii) normalising sums by the feature counts and, iv) normalising the resulting kernel. First, we define Minor Polynomial Kernel:

$$ker\left(\boldsymbol{\phi}, \bar{\boldsymbol{\phi}}\right) = \left(\beta \boldsymbol{\phi}^T \bar{\boldsymbol{\phi}} + \lambda\right)^r \tag{18}$$

We chose $\beta = 1$ and $\lambda = 0$, while $r \geq 1$ denotes the polynomial degree (it is also the order of occurrence pooling). Equation (18) can be rewritten by using the dot product $\left\langle \boldsymbol{\phi}, \bar{\boldsymbol{\phi}} \right\rangle$ of a pair of mid-level features:

$$ker\left(\boldsymbol{\phi}, \bar{\boldsymbol{\phi}}\right) = \left\langle \boldsymbol{\phi}, \bar{\boldsymbol{\phi}} \right\rangle^r \tag{19}$$

We assume $\boldsymbol{\phi}$ and $\bar{\boldsymbol{\phi}}$ are the ℓ_2 norm normalised. We define a kernel function between two sets of mid-level features $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_n\}_{n\in\mathcal{N}}$ and $\bar{\boldsymbol{\Phi}} = \left\{\bar{\boldsymbol{\phi}}_{\bar{n}}\right\}_{\bar{n}\in\bar{\mathcal{N}}}$ given two sets of descriptor indexes $\mathcal{N}$ and $\bar{\mathcal{N}}$ from two images:

$$\begin{aligned} Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right) &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} ker\left(\boldsymbol{\phi}_n, \bar{\boldsymbol{\phi}}_{\bar{n}}\right) \\ &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} \left\langle \boldsymbol{\phi}_n, \bar{\boldsymbol{\phi}}_{\bar{n}} \right\rangle^r \\ &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} \left(\sum_{k=1}^{K} \phi_{kn} \bar{\phi}_{k\bar{n}}\right)^r \end{aligned} \tag{20}$$

Moreover, the rightmost summation in equation (20) can be expressed as a dot product of two self-tensor products of order r. Similar considerations were previously shown in [39]. Thus, this leads to:

$$\begin{aligned} \left(\sum_{k=1}^{K} \phi_{kn} \bar{\phi}_{k\bar{n}}\right)^r &= \sum_{k^{(1)}=1}^{K} ... \sum_{k^{(r)}=1}^{K} \phi_{k^{(1)}} \bar{\phi}_{k^{(1)}} \cdot ... \cdot \phi_{k^{(r)}} \bar{\phi}_{k^{(r)}} \\ &= \left\langle u_:^*(\otimes_r \boldsymbol{\phi}_n), u_:^*\left(\otimes_r \bar{\boldsymbol{\phi}}_{\bar{n}}\right) \right\rangle \end{aligned} \tag{21}$$

Operator $u_:^*$ unfolds an r dimensional tensor into a vector. Now, equations (20) and (21) can be combined:

$$\begin{aligned} Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right) &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} \left\langle u_:^*(\otimes_r \boldsymbol{\phi}_n), u_:^*\left(\otimes_r \bar{\boldsymbol{\phi}}_{\bar{n}}\right) \right\rangle \\ &= \left\langle \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} u_:^*(\otimes_r \boldsymbol{\phi}_n), \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} u_:^*\left(\otimes_r \bar{\boldsymbol{\phi}}_{\bar{n}}\right) \right\rangle \\ &= \left\langle \underset{n\in\mathcal{N}}{\text{avg}}\left[u_:^*(\otimes_r \boldsymbol{\phi}_n)\right], \underset{\bar{n}\in\bar{\mathcal{N}}}{\text{avg}}\left[u_:^*\left(\otimes_r \bar{\boldsymbol{\phi}}_{\bar{n}}\right)\right] \right\rangle \end{aligned} \tag{22}$$

We denote $\text{avg}_{n\in\mathcal{N}}\, \boldsymbol{v}_n$ as a mean vector over $\{\boldsymbol{v}_n\}_{n\in\mathcal{N}}$. Moreover, kernel $Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right)$ is normalised to ensure that self-similarity $Ker\left(\boldsymbol{\Phi}, \boldsymbol{\Phi}\right) = Ker\left(\bar{\boldsymbol{\Phi}}, \bar{\boldsymbol{\Phi}}\right) = 1$. This is achieved by applying a well-known formula:

$$Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right) := \frac{Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right)}{\sqrt{Ker\left(\boldsymbol{\Phi}, \boldsymbol{\Phi}\right)} \sqrt{Ker\left(\bar{\boldsymbol{\Phi}}, \bar{\boldsymbol{\Phi}}\right)}} \tag{23}$$

We replace the unfolding operator $u_:^*$ with previously defined $u_:$ to remove the redundant coefficients from the symmetric self-tensor products and perform unfolding. It is easy to verify that, for Average pooling, the model derived in equation (22) is identical to BoW defined by steps in equations (1), (16), (17), and (3).

2.2 Beyond Average Pooling for Higher-order Occurrence Statistics

This section provides a generalisation of Higher-order Occurrence Pooling to work with Max-pooling that benefits classification. Several evaluations demonstrated that Average pooling performs worse than Max-pooling [13], [29], [11]. We note that Average pooling counts all occurrences of a given visual word in an image. Hence, it quantifies areas spanned by repetitive patterns that are unlikely to appear in the same quantities in a collection of images. However, Max-pooling was shown to be a lower bound of the likelihood of *at least one visual word* $\boldsymbol{m}_k$ *being present in image* i [10]. Thus, Max-pooling acts as a detector of visual words in an image and performs well.

First, we assume two sets of mid-level features $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_n\}_{n\in\mathcal{N}}$ and $\bar{\boldsymbol{\Phi}} = \left\{\bar{\boldsymbol{\phi}}_{\bar{n}}\right\}_{\bar{n}\in\bar{\mathcal{N}}}$ and their descriptor indexes $\mathcal{N}$ and $\bar{\mathcal{N}}$ from a given pair of images. We also define $\max_{n\in\mathcal{N}} v_n = \max\left(\{v_n\}_{n\in\mathcal{N}}\right)$ and $\max_{n\in\mathcal{N}} \boldsymbol{v}_n$ as a vector formed from element-wise $\max\left(\{v_{1n}\}_{n\in\mathcal{N}}\right), \max\left(\{v_{2n}\}_{n\in\mathcal{N}}\right), ...$ over all $\boldsymbol{v}_n$.

The standard BoW with Max-pooling and Polynomial Kernel of degree r is given in equation (24) which

is then expanded in equation (25) and simplified to a dot product between two vectors in equation (26). Thus, it forms a linear kernel. A simple lower bound of this kernel is proposed in equation (27). Note that it represents Higher-order Occurrence Pooling with Max-pooling operator. We further express it as a dot product between two vectors in equation (28).

$$Ker\left(\boldsymbol{\Phi},\bar{\boldsymbol{\Phi}}\right)=\left\langle \hat{\boldsymbol{h}},\bar{\hat{\boldsymbol{h}}}\right\rangle^{r},\ and\ \begin{cases}\hat{h}_k=\max\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right)\\ \bar{\hat{h}}_k=\max\left(\{\bar{\phi}_{kn}\}_{\bar{n}\in\bar{\mathcal{N}}}\right)\end{cases}$$

$$=\left(\sum_{k=1}^{K}\max_{n\in\mathcal{N}}(\phi_{kn})\cdot\max_{\bar{n}\in\bar{\mathcal{N}}}\left(\bar{\phi}_{k\bar{n}}\right)\right)^{r} \tag{24}$$

$$=\sum_{k^{(1)}=1}^{K}\dots\sum_{k^{(r)}=1}^{K}\left(\begin{array}{r}\max_{n\in\mathcal{N}}(\phi_{k^{(1)}n})\cdot\dots\cdot\max_{n\in\mathcal{N}}(\phi_{k^{(r)}n})\cdot\\ \cdot\max_{\bar{n}\in\bar{\mathcal{N}}}\left(\bar{\phi}_{k^{(1)}\bar{n}}\right)\cdot\dots\cdot\max_{\bar{n}\in\bar{\mathcal{N}}}\left(\bar{\phi}_{k^{(r)}\bar{n}}\right)\end{array}\right) \tag{25}$$

$$=\left\langle u_{:}^{*}\left[\otimes_r\max_{n\in\mathcal{N}}\left(\boldsymbol{\phi}_n\right)\right],u_{:}^{*}\left[\otimes_r\max_{\bar{n}\in\bar{\mathcal{N}}}\left(\bar{\boldsymbol{\phi}}_{\bar{n}}\right)\right]\right\rangle \tag{26}$$

$$\geq\sum_{k^{(1)}=1}^{K}\dots\sum_{k^{(r)}=1}^{K}\left(\begin{array}{r}\max_{n\in\mathcal{N}}(\phi_{k^{(1)}n}\cdot\dots\cdot\phi_{k^{(r)}n})\cdot\\ \cdot\max_{\bar{n}\in\bar{\mathcal{N}}}\left(\phi_{k^{(1)}\bar{n}}\cdot\dots\cdot\bar{\phi}_{k^{(r)}\bar{n}}\right)\end{array}\right) \tag{27}$$

$$=\left\langle\max_{n\in\mathcal{N}}\left[u_{:}^{*}(\otimes_r\boldsymbol{\phi}_n)\right],\max_{\bar{n}\in\bar{\mathcal{N}}}\left[u_{:}^{*}(\otimes_r\bar{\boldsymbol{\phi}}_{\bar{n}})\right]\right\rangle \tag{28}$$

We note that Max-pooling violates bi-linearity of equation (28) in contrast to Average pooling which preserves bi-linearity of equation (22). Breaking bi-linearity due to Power Normalisation in [21] led to improvements over the bi-linearity preserving counterpart in [39]. Therefore, formulations with Average pooling, *e.g.* equation (22), are only convenient for performing the linearisation task. Then, Average pooling has to be replaced with a suitable operator.

Moreover, we observed that the signatures from the lower bound formulation in equation (28) have lower normalised entropy compared to the signatures from equation (26). We also verified this analytically for $K=2$ and $r=2$. Therefore, the signatures from the lower bound formulations are more refined.

In practice, we use Second- and Higher-order Occurrence Pooling with the $@n$ operator. Under minor changes, it can be shown as a lower bound of the standard BoW model with the $@n$ operator and Polynomial Kernel. Its signatures have also lower normalised entropy compared to the signatures from the standard BoW. We replace operator $u_{:}^{*}$ with $u_{:}$ and apply ℓ_2 norm to these signatures, as in section 2.1.

Next, an interesting probabilistic difference between the BoW models in equations (26) and (28) can be shown. We first consider Max-pooling in standard BoW with a linear kernel. If mid-level feature coefficients ϕ_{kn} are drawn from a feature distribution under the i.i.d. assumption given a visual word $\boldsymbol{m}_k$, the likelihood of *at least one visual word* $\boldsymbol{m}_k$ *being present in image* i [10] is an upper bound of Max-pooling:

$$1-\prod_{n\in\mathcal{N}}(1-\phi_{kn})\geq\max\left(\{\phi_{kn}\}_{n\in\mathcal{N}}\right) \tag{29}$$

We now derive upper bounds of Max-pooling for the BoW models in equations (26) and (28). We denote the folded image signature from equation (26) as tensor $\boldsymbol{T}=\otimes_r\max_{n\in\mathcal{N}}(\boldsymbol{\phi}_n)\in\mathbb{R}^{K^r}$. Coefficient-wise, this becomes:

$$T_{k^{(1)},\dots,k^{(r)}}=\prod_{s=1}^{r}\max\left(\{\phi_{k^{(s)}n}\}_{n\in\mathcal{N}}\right) \tag{30}$$

Each coefficient of image signature of BoW with Max-pooling and Polynomial Kernel is upper bounded by the probability of *visual words* $\boldsymbol{m}_{k^{(1)}},\dots,\boldsymbol{m}_{k^{(r)}}$ *jointly occurring at least once after pooling in image* i:

$$\prod_{s=1}^{r}\left(1-\prod_{n\in\mathcal{N}}(1-\phi_{k^{(s)}n})\right)\geq T_{k^{(1)},\dots,k^{(r)}} \tag{31}$$

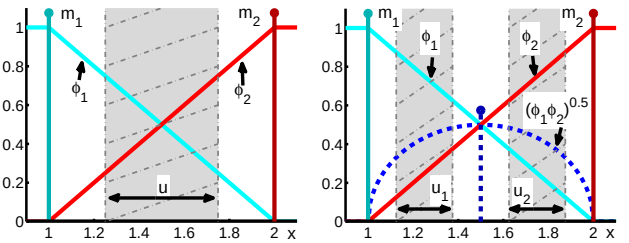
The folded image signature from equation (28) forms tensor $\boldsymbol{T}'=\max_{n\in\mathcal{N}}(\otimes_r\boldsymbol{\phi}_n)\in\mathbb{R}^{K^r}$. Coefficient-wise, it is:

$$T'_{k^{(1)},\dots,k^{(r)}}=\max\left(\left\{\prod_{s=1}^{r}\phi_{k^{(s)}n}\right\}_{n\in\mathcal{N}}\right) \tag{32}$$

Again, we note that every coefficient of image signature of Higher-order Occurrence Pooling with Max-pooling operator is upper bounded by the probability of *visual words* $\boldsymbol{m}_{k^{(1)}},\dots,\boldsymbol{m}_{k^{(r)}}$ *jointly occurring before pooling in at least one mid-level feature* $\boldsymbol{\phi}_n$:

$$1-\prod_{n\in\mathcal{N}}\left(1-\prod_{s=1}^{r}\phi_{k^{(s)}n}\right)\geq T'_{k^{(1)},\dots,k^{(r)}} \tag{33}$$

The joint occurrence of visual words computed in equation (33) per mid-level feature before pooling is more informative compared to the joint occurrence after pooling in equation (31) as, it can be thought of as adding new elements to the visual dictionary. This will be demonstrated in the next section.



(*a*) first-order (*b*) second-order

Fig. 3. Uncertainty in Max-pooling. Mid-level feature coefficients ϕ_1 and ϕ_2 are produced by LLC ($l=2$) for descriptors $1\leq x\leq 2$ given visual words $m_1=1$ and $m_2=2$. (*a*) First-order Occurrence Pooling results in the pooling uncertainty u (the grey area). See text for explanations. (*b*) Second-order statistics produce co-occurrence component $(\phi_1\phi_2)^{0.5}$ that has a maximum for x indicated by the dashed stem. This component limits the pooling uncertainty. The square root is applied to preserve the linear slopes, *e.g.* $(\phi_1\phi_1)^{0.5}=\phi_1$.

2.3 Interpretation of the Joint Occurrence of Visual Words on the Mid-level Feature Level

This section provides intuitive considerations on Second-order Occurrence Pooling. We argue that the joint occurrence of visual words in the mid-level features benefits Max-pooling (and other related operators) by limiting its pooling uncertainty as detailed below. Figure 3 illustrates the mid-level coefficients produced with LLC ($l=2$) for descriptors $1 \leq x \leq 2$. Two one dimensional visual words are used.

Figure 3 (a) shows two linear slopes comprised of coding values ϕ_1 and ϕ_2 for any $1 \leq x \leq 2$. Imagine that we draw randomly a number of descriptors from this interval, obtain ϕ_1 and ϕ_2 from the plot, and apply Max-pooling. Note that the role of pooling is to aggregate the mid-level features into an image signature and preserve information about the descriptors. If we were to draw several times $x_n = 1.5$, we would obtain $\phi_{1n} = \phi_{2n} = 0.5$ for all n. Applying Max-pooling would result in $max(\{\phi_{1n}\}_{n\in\mathcal{N}}) = max(\{\phi_{2n}\}_{n\in\mathcal{N}}) = 0.5$. From this information, one can infer that the only descriptors that could produce such signature are $x_n = 0.5$. Therefore, if $max(\{\phi_{1n}\}_{n\in\mathcal{N}}) \rightarrow 0.5$ and $max(\{\phi_{2n}\}_{n\in\mathcal{N}}) \rightarrow 0.5$, uncertainty in position of descriptors x_n results in $u \rightarrow 0$. However, it takes only two descriptors $x_1 = 1$ and $x_2 = 2$ to mask presence of other descriptors from range $1 < x < 2$. In this case, Max-pooling results in $max(\{\phi_{1n}\}_{n\in\mathcal{N}}) = max(\{\phi_{2n}\}_{n\in\mathcal{N}}) = 1$. One can infer that $x_1 = 1$ and $x_2 = 2$ were present amongst the descriptors. However, other descriptors $1 < x < 2$ could have been also present, *e.g.* $x_3 = 1.25$, $x_4 = 1.5$, and $x_5 = 1.75$. However, there is nothing in the produced signature indicating this. Thus, as $max(\{\phi_{1n}\}_{n\in\mathcal{N}}) \rightarrow 1$ and $max(\{\phi_{2n}\}_{n\in\mathcal{N}}) \rightarrow 1$, uncertainty in position of descriptors x_n results in $u \rightarrow 1$. Both these cases seem undesirable, *e.g.* if all $x_n = 1.5$ then there are no other descriptors in the image. If $x_1 = 1$ and $x_2 = 2$ then another descriptors are masked during Max-pooling.

Figure 3 (b) extends the above example with the second-order statistics. Co-occurrence of ϕ_1 and ϕ_2 results in coefficient $\phi_1\phi_2$. We applied the square root to these statistics to preserve the linear slopes of ϕ_1 and ϕ_2 in the plot, *e.g.* we plotted $(\phi_1\phi_2)^{0.5}$ as a dashed curve instead of $\phi_1\phi_2$. Its maximum occurs for descriptor $x = 1.5$ (the dashed stem). If two descriptors $x_1 = 1$ and $x_2 = 2$ are drawn, they cannot fully mask other descriptors from range $1 < x < 2$. Max-pooling for these descriptors results in $max(\{\phi_{1n}\}_{n\in\mathcal{N}}) = max(\{\phi_{2n}\}_{n\in\mathcal{N}}) = 1$ and $max(\{\phi_{1n}\phi_{2n}\}_{n\in\mathcal{N}}) = 0$. Note that drawing a third descriptor $x_3 = 1.5$ would result in $max(\{\phi_{1n}\phi_{2n}\}_{n\in\mathcal{N}}) = 0.5$ and mark its presence in the image signature. Thus, second-order statistics appear to increase the dictionary resolution. This limits the uncertainty of Max-pooling such that $u_1 + u_2 \leq u$.

Figure 4 illustrates the mid-level coefficients ϕ_1, ϕ_2, ϕ_3, ϕ_4 produced with SC ($\alpha = 1$) for $\boldsymbol{x} = [x_1, x_2]^T \in [-5, 5]^2$, and the corresponding co-occurrence coefficients $(\phi_1\phi_2)^{0.5}$, $(\phi_2\phi_3)^{0.5}$, $(\phi_3\phi_4)^{0.5}$, $(\phi_4\phi_1)^{0.5}$. We applied the square root to these statistics to preserve the linear slopes of ϕ_1, ϕ_2, ϕ_3, and ϕ_4. The maxima of the co-occurrence functions are indicated by the dashed stems. They can be seen as the additional elements of the visual dictionary. Note that $(\phi_1\phi_3)^{0.5} = (\phi_2\phi_4)^{0.5} = 0$ for any $\boldsymbol{x} \in [-5, 5]^2$. This demonstrates that the subspace learned with SC is preserved in the second-order statistics in contrast to 2D histogram representations [36] that compute sum between all pairs of mid-level feature coefficients.

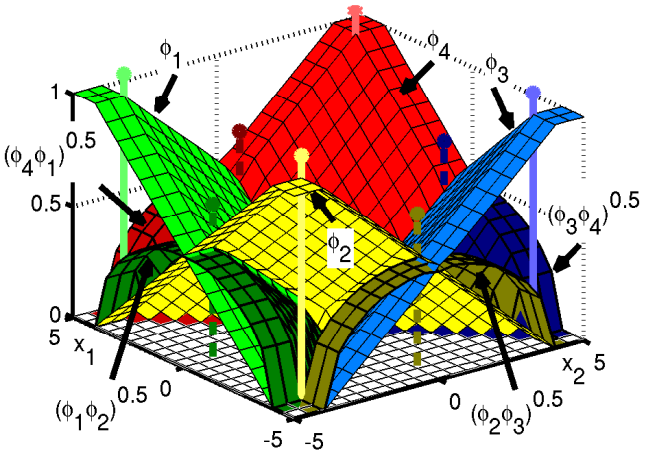
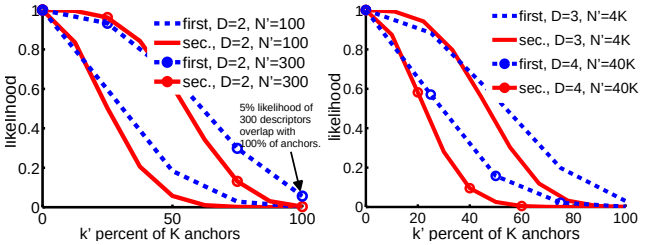


Fig. 4. Co-occurrence coefficients. Mid-level feature coefficients $\phi_1, ..., \phi_4$ are produced by SC ($\alpha = 1$) for descriptors $\boldsymbol{x} = [x_1, x_2]^T \in [-5, 5]^2$ and arbitrarily chosen $k = 1, ..., 4$ visual words $\boldsymbol{m}_k \in [-5, 5]^2$ indicated by the solid line stems. The second-order statistics produce co-occurrence components $(\phi_1\phi_2)^{0.5}$, $(\phi_2\phi_3)^{0.5}$, $(\phi_3\phi_4)^{0.5}$, and $(\phi_4\phi_1)^{0.5}$ with maxima for $\boldsymbol{x}$ indicated by the dashed stems. The remaining co-occurrence coefficients are equal 0, *e.g.* $(\phi_1\phi_3)^{0.5} = 0$. This shows that the subspace learned with SC is preserved.

We illustrated earlier that if a descriptor overlaps with an anchor from the dictionary, other near-by descriptors may be not represented in the final signature. Thus, we perform an experiment to quantify



(*a*) (*b*)

Fig. 5. Saturation effect in Max-pooling for the first- and second-order pooling ('first' and 'sec'). Descriptor space $[-5, 5]^D$ is quantised into 21^D values. We draw from it N' values given the uniform distribution. (*a*) Likelihood that at least k' percent of $K = 4$ anchors will overlap with N' descriptors given $D = 2$. (*b*) Simulation for $D = 3$ and $D = 4$. Note that the second-order pooling suffers less from the saturation effect in all cases.

this behaviour. We illustrate the likelihood that at least k' percent of $K = 4$ anchors will overlap with N' descriptors in figure 5. These descriptors are drawn at random from descriptor space $[-5, 5]^D$ quantised into 21^D values. We consider an anchor to overlap with a descriptor if their both quantised values are the same. Figure 5 (a) shows that if $N' = 300$ descriptors are drawn given $D = 2$, the likelihood they will overlap with all 4 anchors is 5%. For $N' = 100$ descriptors this amounts to 0%. Furthermore, the second-order statistics contribute additional 4 non-zero coefficients that increase resolution of the visual dictionary (see figure 4). Therefore, it is more likely that the descriptors will overlap with at least one anchor for the second- rather than the first-order cases. However, it is less likely that the descriptors will overlap with all anchors for the second-order cases compared to the first-order representations. This demonstrates that the second-order statistics improve capabilities of Max-pooling (and related pooling operators). Lastly, figure 5 (b) demonstrates the same behaviour in higher dimensional spaces as, for $D=3$ and $D=4$, there are 5 and 6 non-zero second-order coefficients, respectively. Similar trend follows in $D=128$ dimensional spaces.

3 Bag-of-Words for Bi- and Multi-modal Codes with Second- and Higher-Order Occurrence Pooling

Bi- and Multi-modal extensions of BoW are proposed below. Their derivations are provided in section 3.3. Sections 3.1 and 3.2 outline the early and late fusion of cues for BoW (used for comparisons on the grey and colour features). Section 3.4 presents SPM and DoPM as special cases of our bi-modal fusion. A Residual Descriptor is proposed in section 3.5 to further demonstrate robustness of the bi-modal fusion.

Grey scale and colour cues are often combined due to their complementary nature that benefits the object category recognition [5], [40], [20], [41], [42], [43], [44], [45] and visual concept detection [46], [47], [48], [49], [11]. Some approaches employ so-called early fusion of modalities that occurs on the descriptor level [5], [40], [31]. Another methods perform coding and pooling steps on various modalities first, followed by so-called late fusion which involves combining multiple kernels [41], [42], [44], [45], [40], [48].

The Second- and Higher-order Occurrence Pooling are characterised by their ability to capture the joint occurrence of visual words per mid-level feature as formulated in equation (33) of section 2.2. This ability extends to bi- and multi-modal scenarios. Each modality is represented by its mid-level features in the joint occurrence statistics. Moreover, linearisation of Minor Polynomial Kernel yields so-called cross-term which captures the joint occurrence of visual words between mid-level features of various modalities, *e.g.* spatially corresponding grey and colour features.

3.1 The Early Fusion in Bag-of-Words

We showed in [31] that the early fusion of modalities can be thought of as a trade-off between the quantisation losses of linearly coded signals. With the means of Sparse Coding, we showed that such a trade-off can be implemented by concatenating modalities on the descriptor level without explicitly redesigning the coding method. Such a fusion of descriptors with their spatial coordinates is called Spatial Coordinate Coding [31]. It improves the classification performance and limits the size of image signatures due to bypassed Spatial Pyramid Matching [30]. A similar fusion on descriptor level was also used in recognition with discriminatively trained Gaussian Mixtures [50] and by Joint Sparse Coding [51]. Below, we generalise [31] to work with arbitrary Q modalities:

$$\boldsymbol{\phi} = \arg\min_{\bar{\boldsymbol{\phi}}} \sum_{q=1}^{Q} \beta^{(q)} \left\| \boldsymbol{x}^{(q)} - \boldsymbol{\mathcal{M}}^{(q)} \bar{\boldsymbol{\phi}} \right\|_2^2 + \alpha \|\bar{\boldsymbol{\phi}}\|_1 \quad s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0 \tag{34}$$

Sparse Coding [12], [13] is extended in equation (34) by combining Q terms for quantisation loss with the sparsity term. Weights $\beta^{(1)}, ..., \beta^{(Q)}$ determine the impact of features $\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(Q)}$ and dictionaries $\boldsymbol{\mathcal{M}}^{(1)}, ..., \boldsymbol{\mathcal{M}}^{(Q)}$ in this multi-modal trade-off. One can also impose $\beta^{(1)}+...+\beta^{(Q)}=1$. Equation (34) is further rewritten to reduce this problem to ordinary SC:

$$\boldsymbol{\phi} = \arg\min_{\bar{\boldsymbol{\phi}}} \sum_{q=1}^{Q} \left\| \sqrt{\beta^{(q)}} \boldsymbol{x}^{(q)} - \sqrt{\beta^{(q)}} \boldsymbol{\mathcal{M}}^{(q)} \bar{\boldsymbol{\phi}} \right\|_2^2 + \alpha \|\bar{\boldsymbol{\phi}}\|_1 \quad s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0 \tag{35}$$

Vector $\boldsymbol{x}$ and dictionary $\boldsymbol{\mathcal{M}}$ for ordinary SC can be formed by concatenation across Q modalities:

$$\boldsymbol{x} = \left[\sqrt{\beta^{(1)}} \boldsymbol{x}^{(1)^T}, ..., \sqrt{\beta^{(Q)}} \boldsymbol{x}^{(Q)^T} \right]^T$$
$$\boldsymbol{\mathcal{M}} = \left[\sqrt{\beta^{(1)}} \boldsymbol{\mathcal{M}}^{(1)^T}, ..., \sqrt{\beta^{(Q)}} \boldsymbol{\mathcal{M}}^{(Q)^T} \right]^T \tag{36}$$

Spatial Coordinate Coding [31] is often used in this work. The descriptor vectors $\boldsymbol{x}$ are augmented with their spatial positions $\boldsymbol{x}^s = [c^x/w, c^y/h]^T$ that are normalised by the image width and height. Thus $\boldsymbol{x} := [\sqrt{\beta^s}\boldsymbol{x}^{s^T}, \sqrt{1-\beta^s}\boldsymbol{x}^T]^T$. The trade-off between the visual appearance and spatial bias is balanced by β^s, which is determined by cross-validation.

Opponent SIFT is comprised of two modalities. The orientations of gradients are extracted from the luminance and chromaticity maps to form two ℓ_2 norm normalised vectors $\boldsymbol{x}$ and $\boldsymbol{x}^c$. Vector $\boldsymbol{x}$ is augmented with the spatial and colour terms $\boldsymbol{x}^s$ and $\boldsymbol{x}^c$ that are balanced by β^s and β^c. This results in $\boldsymbol{x} := [\sqrt{\beta^s}\boldsymbol{x}^{s^T}, \sqrt{1-\beta^s-\beta^c}\boldsymbol{x}^T, \sqrt{\beta^c}\boldsymbol{x}^{c^T}]^T$ that is used in comparisons to the extension proposed in section 3.3.

3.2 The Late Fusion in Bag-of-Words

Fusing multiple modalities can be performed by coding and pooling them first, forming the kernels, and

linearly combining them [41], [42], [45], [48], [40], [49]:

$$Ker_{ij} = \sum_{q=1}^{Q} \beta^{(q)} Ker_{ij}^{(q)} \tag{37}$$

Weights $\beta^{(1)}, ..., \beta^{(Q)}$ determine the impact of kernels $Ker^{(1)}, ..., Ker^{(Q)}$. One can further impose that $\beta^{(1)}+...+\beta^{(Q)}=1$. There are various approaches to learning weights. However, given a small number of modalities, these weights can be easily found by cross-validation and result in performance on a par with MKL [45], [40], [48]. We use such a fusion only for comparisons with the extension given in section 3.3.

3.3 Linearisation of Minor Polynomial Kernel for Bi- and Multi-modal Codes

The proposed BoW with Higher-order Occurrence Pooling for bi- and multi-modal codes can be derived in the following four steps: i) defining a kernel function referred to as Minor Kernel on Q pairs of mid-level features, one pair $\left(\boldsymbol{\phi}_n^{(q)}, \bar{\boldsymbol{\phi}}_{\bar{n}}^{(q)}\right)$ per modality $q=1, ..., Q$, ii) summing over pairs of mid-level features from a given pair of images, iii) normalising with respect to the feature count, iv) normalising the final kernel. First, we define Minor Polynomial Kernel:

$$ker\left(\left\{\left(\boldsymbol{\phi}^{(q)}, \bar{\boldsymbol{\phi}}^{(q)}\right)\right\}_{q=1}^{Q}\right) = \left(\sum_{q=1}^{Q} \beta^{(q)} \boldsymbol{\phi}^{(q)^T} \bar{\boldsymbol{\phi}}^{(q)} + \lambda\right)^r \tag{38}$$

We chose $\lambda = 0$, while $\beta^{(1)}, ..., \beta^{(Q)}$ are weights determining the impact of modalities, and $r \geq 1$ denotes the polynomial degree (the order of occurrence pooling). One can further impose $\beta^{(1)}+...+\beta^{(Q)}=1$. Equation (38) can be rewritten with the dot-product:

$$ker\left(\left\{\left(\boldsymbol{\phi}^{(q)}, \bar{\boldsymbol{\phi}}^{(q)}\right)\right\}_{q=1}^{Q}\right) = \left(\sum_{q=1}^{Q} \beta^{(q)} \left\langle \boldsymbol{\phi}^{(q)}, \bar{\boldsymbol{\phi}}^{(q)} \right\rangle\right)^r \tag{39}$$

We assume that $\boldsymbol{\phi}^{(q)}$ and $\bar{\boldsymbol{\phi}}^{(q)}$ are the ℓ_2 norm normalised. We also define a kernel function between two sets of mid-level features $\boldsymbol{\Phi} = \left\{\left\{\boldsymbol{\phi}_n^{(q)}\right\}_{n\in\mathcal{N}}\right\}_{q=1}^{Q}$ and $\bar{\boldsymbol{\Phi}} = \left\{\left\{\bar{\boldsymbol{\phi}}_{\bar{n}}^{(q)}\right\}_{\bar{n}\in\bar{\mathcal{N}}}\right\}_{q=1}^{Q}$ given descriptor indexes $\mathcal{N}$ and $\bar{\mathcal{N}}$ from two images and given Q modalities:

$$\begin{aligned} Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right) &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} ker\left(\left\{\left(\boldsymbol{\phi}^{(q)}, \bar{\boldsymbol{\phi}}^{(q)}\right)\right\}_{q=1}^{Q}\right) \\ &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} \left(\sum_{q=1}^{Q} \beta^{(q)} \left\langle \boldsymbol{\phi}^{(q)}, \bar{\boldsymbol{\phi}}^{(q)} \right\rangle\right)^r \\ &= \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \frac{1}{|\bar{\mathcal{N}}|} \sum_{\bar{n}\in\bar{\mathcal{N}}} \left(\sum_{q=1}^{Q} \beta^{(q)} \sum_{k=1}^{K} \phi_{kn}^{(q)} \bar{\phi}_{k\bar{n}}^{(q)}\right)^r \end{aligned} \tag{40}$$

Bi-modal Second-order Occurrence Pooling is first derived by linearising the above kernel by setting parameters $Q = 2$ (two coders) and $r = 2$ (second-order). We denote $\beta^{(1)} = \beta$ and $\beta^{(2)} = 1 - \beta$. Thus, Minor Polynomial Kernel from equation (39), also appearing in equation (40), can be rewritten as:

$$\begin{aligned} &\left(\beta \sum_{k=1}^{K} \phi_{kn}^{(1)} \bar{\phi}_{k\bar{n}}^{(1)} + (1-\beta) \sum_{k=1}^{K} \phi_{kn}^{(2)} \bar{\phi}_{k\bar{n}}^{(2)}\right)^2 \qquad (41) \\ &= \beta^2 \left(\sum_{k=1}^{K} \phi_{kn}^{(1)} \bar{\phi}_{k\bar{n}}^{(1)}\right)^2 + (1-\beta)^2 \left(\sum_{k=1}^{K} \phi_{kn}^{(2)} \bar{\phi}_{k\bar{n}}^{(2)}\right)^2 \\ &\quad + 2\beta(1-\beta) \left(\sum_{k=1}^{K} \phi_{kn}^{(1)} \bar{\phi}_{k\bar{n}}^{(1)}\right) \left(\sum_{k=1}^{K} \phi_{kn}^{(2)} \bar{\phi}_{k\bar{n}}^{(2)}\right) \\ &= \beta^2 \left\langle u_:^*\left(\boldsymbol{\phi}_n^{(1)} \boldsymbol{\phi}_n^{(1)^T}\right), u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)^T}\right) \right\rangle \qquad (42) \\ &\quad + 2\beta(1-\beta) \left\langle u_:^*\left(\boldsymbol{\phi}_n^{(1)} \boldsymbol{\phi}_n^{(2)^T}\right), u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)^T}\right) \right\rangle \qquad (43) \\ &\quad + (1-\beta)^2 \left\langle u_:^*\left(\boldsymbol{\phi}_n^{(2)} \boldsymbol{\phi}_n^{(2)^T}\right), u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)^T}\right) \right\rangle \qquad (44) \end{aligned}$$

Minor Polynomial Kernel in equation (41) is linearised for order $r = 2$ with three dot product terms in equations (42), (43), and (44). Substituting Minor Polynomial Kernel in equation (40) by these terms yields:

$$\begin{aligned} &Ker\left(\boldsymbol{\Phi}, \bar{\boldsymbol{\Phi}}\right) = Ker_{ij} \\ &= \beta^2 \left\langle \underset{n\in\mathcal{N}}{\text{avg}}\left[u_:^*\left(\boldsymbol{\phi}_n^{(1)} \boldsymbol{\phi}_n^{(1)^T}\right)\right], \underset{\bar{n}\in\bar{\mathcal{N}}}{\text{avg}}\left[u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)^T}\right)\right] \right\rangle \qquad (45) \\ &+ 2\beta(1-\beta) \left\langle \underset{n\in\mathcal{N}}{\text{avg}}\left[u_:^*\left(\boldsymbol{\phi}_n^{(1)} \boldsymbol{\phi}_n^{(2)^T}\right)\right], \underset{\bar{n}\in\bar{\mathcal{N}}}{\text{avg}}\left[u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(1)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)^T}\right)\right] \right\rangle \qquad (46) \\ &+ (1-\beta)^2 \left\langle \underset{n\in\mathcal{N}}{\text{avg}}\left[u_:^*\left(\boldsymbol{\phi}_n^{(2)} \boldsymbol{\phi}_n^{(2)^T}\right)\right], \underset{\bar{n}\in\bar{\mathcal{N}}}{\text{avg}}\left[u_:^*\left(\bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)} \bar{\boldsymbol{\phi}}_{\bar{n}}^{(2)^T}\right)\right] \right\rangle \qquad (47) \end{aligned}$$

Note that the final kernel for the two coders is comprised of three dot product terms. Equations (45) and (47) represent simply Second-order Occurrence Pooling for coders $q = 1$ and $q = 2$. They are identical with the uni-modal coding given by equation (22) in section 2.1. However, equation (46) represents the cross-term that captures co-occurrences between visual words of mid-level features $\phi_{kn}^{(1)}$ and $\phi_{k'n}^{(2)}$ from two coders. This term will be shown to improve results in section 5.4.

In practice, we use Second-order Occurrence Pooling and the $@n$ operator as in section 2.2. We replace the unfolding operator $u_:^*$ in equations (45, 47) with $u_:$ that removes the redundant coefficients from the symmetric self-tensor products and performs unfolding. The image signatures are the ℓ_2 norm normalised.

Bi-modal Higher-order Occurrence Pooling can be derived from expansion of Minor Polynomial Kernel in equation (39) using Binomial theorem:

$$\Big(\beta a + \big(1-\beta\big) b\Big)^r = \sum_{s=0}^{r} \binom{r}{s} \Big(\beta a\Big)^{r-s} \Big(\big(1-\beta\big) b\Big)^s \tag{48}$$

Two coders $Q = 2$ and order $r \geq 2$ are assumed, and substitutions $a = \left\langle \boldsymbol{\phi}^{(1)}, \bar{\boldsymbol{\phi}}^{(1)} \right\rangle$ and $b = \left\langle \boldsymbol{\phi}^{(2)}, \bar{\boldsymbol{\phi}}^{(2)} \right\rangle$ are made. The derivations follow the same reasoning as for Bi-modal Second-order Occurrence Pooling. We

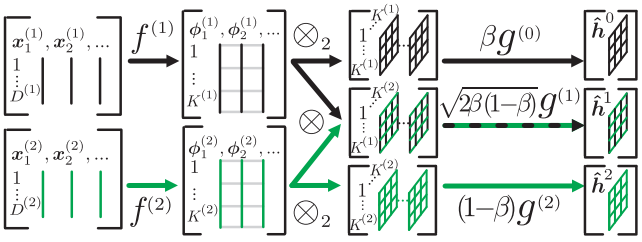


Fig. 6. Bi-modal Bag-of-Words with Second-order Occurrence Pooling. Two types of local descriptors $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are extracted from an image and coded by coders $f^{(1)}$ and $f^{(2)}$. Self-tensor product $\otimes_2$ computes co-occurrences of visual words in every mid-level feature $\boldsymbol{\phi}^{(1)}$ and $\boldsymbol{\phi}^{(2)}$, respectively. Moreover, tensor product $\otimes$ captures co-occurrences of visual words between $\boldsymbol{\phi}^{(1)}$ and $\boldsymbol{\phi}^{(2)}$ (cross-term operation). Pooling g aggregates co-occurring visual words. For clarity, the unfolding operator $u_:$ from equation (50) is dropped.

skip them for clarity and define Bag-of-Words with Bi-modal Higher-order Occurrence Pooling:

$$\begin{aligned}\boldsymbol{\phi}_n^{(1)} &= f^{(1)}\left(\boldsymbol{x}_n^{(1)}, \boldsymbol{\mathcal{M}}^{(1)}\right)\\ \boldsymbol{\phi}_n^{(2)} &= f^{(2)}\left(\boldsymbol{x}_n^{(2)}, \boldsymbol{\mathcal{M}}^{(2)}\right)\end{aligned}\ ,\quad \forall n \in \mathcal{N} \tag{49}$$

$$\boldsymbol{\psi}_n^s = u_:\left[\left(\otimes_{r-s}\,\boldsymbol{\phi}_n^{(1)}\right)\left(\otimes_s\,\boldsymbol{\phi}_n^{(2)}\right)\right],\ s=0,...,r \tag{50}$$

$$\hat{h}_k^s = \binom{r}{s}^{\frac{1}{2}}(1-\beta)^{\frac{s}{2}}\beta^{\frac{r-s}{2}} g^{(s)}\left(\{\psi_{kn}^s\}_{n\in\mathcal{N}}\right),\ k=1,...,K^{(r,s)} \tag{51}$$

$$\boldsymbol{h} = \hat{\boldsymbol{h}}/\|\hat{\boldsymbol{h}}\|_2\ ,\ \hat{\boldsymbol{h}} = \left[\hat{\boldsymbol{h}}^{\mathbf{0}^T},...,\hat{\boldsymbol{h}}^{\boldsymbol{r}^T}\right]^T \tag{52}$$

Figure 6 illustrates the above model given Bi-modal BoW with Second-order Occurrence Pooling.

Equation (49) represents the coding step for two coders $f^{(1)}: \mathbb{R}^{D^{(1)}} \rightarrow \mathbb{R}^{K^{(1)}}$ and $f^{(2)}: \mathbb{R}^{D^{(2)}} \rightarrow \mathbb{R}^{K^{(2)}}$ that embed descriptors $\boldsymbol{x}_n^{(1)} \in \mathbb{R}^{D^{(1)}}$ and $\boldsymbol{x}_n^{(2)} \in \mathbb{R}^{D^{(2)}}$ representing two modalities into the visual vocabulary spaces given by dictionaries $\boldsymbol{\mathcal{M}}^{(1)} \in \mathbb{R}^{D^{(1)}\times K^{(1)}}$ and $\boldsymbol{\mathcal{M}}^{(2)} \in \mathbb{R}^{D^{(2)}\times K^{(2)}}$. This results in two groups of mid-level features $\boldsymbol{\phi}_n^{(1)} \in \mathbb{R}^{K^{(1)}}$ and $\boldsymbol{\phi}_n^{(2)} \in \mathbb{R}^{K^{(2)}}$ given the descriptor indexes $n \in \mathcal{N}$ of image $i \in \mathcal{I}$. Moreover, the coders used can be of different types, the descriptor dimensionality $D^{(1)}$ may differ from $D^{(2)}$, and dictionary sizes $K^{(1)}$ and $K^{(2)}$ may differ.

Equation (50) represents the joint occurrence of visual words in $\boldsymbol{\phi}_n^{(1)}$ or $\boldsymbol{\phi}_n^{(2)}$, or the cross-modal joint occurrence of visual words per mid-level pair $(\boldsymbol{\phi}_n^{(1)}, \boldsymbol{\phi}_n^{(2)})$, depending on k and s. It results from an expansion of Minor Polynomial Kernel in equation (39) according to Binomial theorem. A similar expansion was performed in equations (41-44) for $r=2$. However, we moved weight β inside the dot product and conveniently appended them to the pooling operator in equation (51). Thus, only vectors $\boldsymbol{\psi}_n^s$ that would appear inside the dot product expressions are given. Note that equation (50) replaces the unfolding operator $u_:^*$ with $u_:$ that both unfolds tensors and removes the redundant coefficients resulting from the symmetries which occur in self-tensors $\otimes_{r-s}\boldsymbol{\phi}_n^{(1)}$ and $\otimes_s\boldsymbol{\phi}_n^{(2)}$ if $r-s\geq 2$ or $s\geq 2$. The dimensionality of $\boldsymbol{\psi}_n^s$ after removing repeated coefficients and unfolding is $K^{(r,s)} = K^{(r-s)}K^{(s)} = \binom{K+r-s-1}{r-s}\binom{K+s-1}{s}$.

Equation (51) represents pooling that aggregates the joint occurrences or the cross-modal joint occurrences of visual words. Function $g^{(s)}: \mathbb{R}^{|\mathcal{N}|} \rightarrow \mathbb{R}$ takes the k^{th} joint occurrence (or the cross-modal joint occurrence) from ψ_{kn}^s for all $n \in \mathcal{N}$ given image i to produce a k^{th} coefficient in vector $\hat{\boldsymbol{h}}^s \in \mathbb{R}^{K^{(r,s)}}$. The weighting factors preceding $g^{(s)}$ result from Binomial expansion.

Equation (52) concatenates various joint occurrence statistics and also performs the ℓ_2 norm normalisation.

Bi-modal Second-order Occurrence Pooling in equations (45), (46), and (47) can also be readily derived from Bi-modal Higher-order Occurrence Pooling. If $r=2$, then equation (50) results in three terms:

$$\boldsymbol{\psi}_n^0 = u_:^*\left(\boldsymbol{\phi}_n^{(1)}\boldsymbol{\phi}_n^{(1)T}\right), \hat{h}_k^0 = \beta\,\mathrm{avg}\left(\left\{\psi_{kn}^0\right\}_{n\in\mathcal{N}}\right) \tag{53}$$

$$\boldsymbol{\psi}_n^1 = u_:^*\left(\boldsymbol{\phi}_n^{(1)}\boldsymbol{\phi}_n^{(2)T}\right), \hat{h}_k^1 = \sqrt{2\beta(1-\beta)}\,\mathrm{avg}\left(\left\{\psi_{kn}^1\right\}_{n\in\mathcal{N}}\right) \tag{54}$$

$$\boldsymbol{\psi}_n^2 = u_:^*\left(\boldsymbol{\phi}_n^{(2)}\boldsymbol{\phi}_n^{(2)T}\right), \hat{h}_k^2 = (1-\beta)\,\mathrm{avg}\left(\left\{\psi_{kn}^2\right\}_{n\in\mathcal{N}}\right) \tag{55}$$

Employing Average pooling for the step in equation (51) is done by replacing $g^{(s)}$ with avg for $s=0,1,2$. Pooling over $\boldsymbol{\psi}_n^0$, $\boldsymbol{\psi}_n^1$, and $\boldsymbol{\psi}_n^2$ from equations (53), (54), and (55) results in $\hat{\boldsymbol{h}}^0$, $\hat{\boldsymbol{h}}^1$, and $\hat{\boldsymbol{h}}^2$ per image. Forming three kernels $\langle\hat{\boldsymbol{h}}_i^0, \hat{\boldsymbol{h}}_j^0\rangle$, $\langle\hat{\boldsymbol{h}}_i^1, \hat{\boldsymbol{h}}_j^1\rangle$, and $\langle\hat{\boldsymbol{h}}_i^2, \hat{\boldsymbol{h}}_j^2\rangle$ given images i and j and adding such kernels is equivalent to operations in equations (45), (46), and (47).

Multi-modal Higher-order Occurrence Pooling can be readily derived by expanding Minor Polynomial Kernel in equation (39) using Multinomial theorem. This fusion can be also performed by concatenating the mid-level features of index n from Q coders:

$$\boldsymbol{\phi}_n = \left[\sqrt{\beta^{(1)}}\boldsymbol{\phi}_n^{(1)T}, \sqrt{\beta^{(2)}}\boldsymbol{\phi}_n^{(2)T}, ..., \sqrt{\beta^{(Q)}}\boldsymbol{\phi}_n^{(Q)T}\right]^T \tag{56}$$

Such formed mid-level features $\boldsymbol{\phi}_n$ can be fed to equation (16) to form tensors leading to Bi- and Multi-modal Second- and Higher-order Occurrence Pooling.

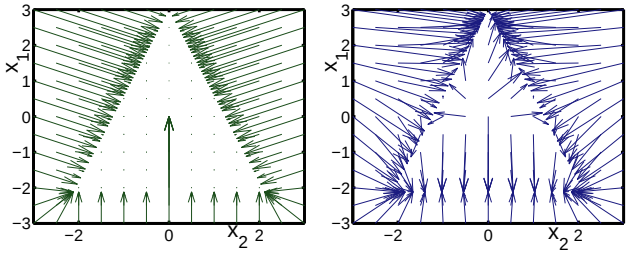
3.4 Special Cases of Bi-modal Second-order Occurrence Pooling: Pyramid Matching Techniques

Below, Spatial Pyramid Matching [30], [13] (SPM) is shown as a special case of Bi-modal Second-order Occurrence Pooling. We employ two coders: $f^{(1)}$ is *e.g.* SC, LLC, LcSA, while $f^{(2)}$ produces a binary vector with assignments of descriptors to spatial partitions:

$$\boldsymbol{\phi}_n^{(1)} = f\left(\boldsymbol{x}_n, \boldsymbol{\mathcal{M}}\right) \tag{57}$$

$$\boldsymbol{\phi}_n^{(2)} = \left[\oplus_{t=1}^{\overline{T}}\oplus_{z_x=0}^{Z_t-1}\oplus_{z_y=0}^{\overline{Z}_t-1}\mathbb{1}\left(\left\lfloor\frac{Z_t c_n^x}{w}\right\rfloor = z_x\right)\mathbb{1}\left(\left\lfloor\frac{\overline{Z}_t c_n^y}{h}\right\rfloor = z_y\right)\right]^T$$

Equation (57) uses the operator $\oplus_{t=1}^{\overline{T}}$ denoting concatenation over $\overline{T}$ levels of spatial quantisation. Operators $\oplus_{z_x=0}^{Z_t-1}$ and $\oplus_{z_y=0}^{\overline{Z}_t-1}$ concatenate binary values over



(*a*) SC, α=1 (*b*) LcSA, σ^2=4, l=2

Fig. 7. Illustration of Residual Descriptors. Flow of the descriptors from their original positions $\boldsymbol{x}$ denoted by the grid points to the corresponding reconstructed positions $\hat{\boldsymbol{x}}$ pointed to by the arrows. (*a*) SC: optimal reconstruction within the triangular region (no displacement). (*b*) LcSA: poor reconstruction due to low $l=2$.

vertical and horizontal partitions $z_x = 0, ..., Z_t - 1$ and $z_y = 0, ..., \overline{Z}_t - 1$, where vectors $\boldsymbol{Z}$ and $\overline{\boldsymbol{Z}}$ define the numbers of splits for each pyramid level $t = 1, ..., \overline{T}$. Binary indicator $\mathbb{1}(z_l = z_r)$ returns 1 if $z_l = z_r$, 0 otherwise. Next, $0 \leq c_n^x < w$ and $0 \leq c_n^y < h$ are the spatial coordinates of descriptor $\boldsymbol{x}_n$, w and h are the image width and height, and $\lfloor . \rfloor$ is the floor operator.

SPM (*e.g.* variant from [13]) can be obtained by simply applying Bi-modal Second-order Occurrence Pooling, extracting the cross-modal joint occurrence of visual words that form $\boldsymbol{\psi}_n^1$, and suppressing the joint occurrence of visual words in $\boldsymbol{\psi}_n^0$ and $\boldsymbol{\psi}_n^2$:

$$\boldsymbol{\psi}_n^0 = [\,], \boldsymbol{\psi}_n^1 = u_{:}^{*}\left(\boldsymbol{\phi}_n^{(1)} \boldsymbol{\phi}_n^{(2)T}\right), \boldsymbol{\psi}_n^2 = [\,] \tag{58}$$

The parameters for SPM with $1{\times}1, 3{\times}1, 1{\times}3$, and $2{\times}2$ spatial splits are $\overline{T}$=4, $\boldsymbol{Z}$=[1 3 1 2]^T and $\overline{\boldsymbol{Z}}$=[1 1 3 2]^T. SPM gathers second-order statistics by quantifying co-occurrences between visual words in the mid-level features and spatial locations that are quantised at several levels of quantisation. Thus, SPM enhances the visual vocabulary with a spatial vocabulary: similar visual appearances can take various meanings based on their spatial locations. We stress that Bi-modal Second-order Occurrence Pooling entails three terms $\boldsymbol{\psi}_n^0$, $\boldsymbol{\psi}_n^1$, and $\boldsymbol{\psi}_n^2$ rather than just $\boldsymbol{\psi}_n^1$. Thus, we will evaluate the three-term based SPM model as ordinary SPM appears to be a simplified second-order model.

By analogy to SPM, Dominant Angle Pyramid Matching [31], [11] can be obtained by re-defining the coder in equation (58) to exploit orientations of dominant edges from the local descriptors instead of spatial coordinates. BoW schemes like BossaNova [52] can be also derived by employing: i) the descriptor assignment to l-nearest k-means clusters as the first coder, ii) the descriptor assignment to radial zones defined over k-means clusters as the second coder.

3.5 Complementing Coder by Residual Descriptor

We now present the Residual Descriptor (RD) that is used along with a chosen coder (*e.g.* SC, LLC, or LcSA) to address its quantisation loss. RD is not related to the bi-modal fusion, however, by its means an interesting property of Bi-modal Second-order Occurrence Pooling can be shown. SC and LLC optimise a trade-off between a quantisation loss (defined below) and a chosen regularisation penalty, *e.g.* sparsity or locality as in equations (4, 5). Measuring the quality of quantisation in such mappings follows the theory of Linear Coordinate Coding [14]. The linear approximation of descriptor $\boldsymbol{x}$ given dictionary $\boldsymbol{\mathcal{M}}$ and coder f that produces mid-level feature $\boldsymbol{\phi}$ is $\hat{\boldsymbol{x}} = \boldsymbol{\mathcal{M}} f(\boldsymbol{x}) = \boldsymbol{\mathcal{M}}\boldsymbol{\phi}$. The quantisation loss a.k.a quantisation error is defined as:

$$\xi^2 = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2. \tag{59}$$

However, ξ^2 quantifies only the magnitude of such an error. Hence, we define Residual Descriptor vector:

$$\boldsymbol{\xi} = \boldsymbol{x} - \hat{\boldsymbol{x}} \tag{60}$$

Residual Descriptors are illustrated in figure 7. Having coded descriptors $\boldsymbol{x} = [x_1, x_2]^T \in [-3, 3]^2$ with three atoms $\boldsymbol{m}_1 = [0, 3]^T$, $\boldsymbol{m}_2 \approx [-2, -2]^T$, and $\boldsymbol{m}_3 \approx [2, -2]^T$ by SC and LcSA coders, the obtained codes $\boldsymbol{\phi}$ are projected back to the descriptor space: $\hat{\boldsymbol{x}} = \boldsymbol{\mathcal{M}}\boldsymbol{\phi}$. The resulting quantisation artifacts, used by us as RD, are visualised by displacements between each descriptor $\boldsymbol{x}$ and its approximation $\hat{\boldsymbol{x}}$. Plot (a) shows SC for regularisation $\alpha = 1$ (good trade-off). Plot (b) shows LcSA with large quantisation errors due to low $l=2$.

The displacements in figure 7 are shown with respect to descriptors $\boldsymbol{x}$. However, encoding the magnitude and orientation of the quantisation error given equation (60) does not indicate which descriptors are the source of errors. Hence, we propose to use Bi-modal Second-order Occurrence Pooling framework to combine both mid-level features $\boldsymbol{\phi}$ and vectors $\boldsymbol{\xi}$:

$$\boldsymbol{\phi}_n^{(1)} = f\left(\boldsymbol{x}_n, \boldsymbol{\mathcal{M}}\right), \;\; \boldsymbol{\phi}_n^{(2)} = \boldsymbol{x}_n - \boldsymbol{\mathcal{M}}\boldsymbol{\phi}_n^{(1)} \tag{61}$$

In this formulation, the cross-term captures co-occurrences between visual words of mid-level feature $\boldsymbol{\phi}^{(1)}$ of descriptor $\boldsymbol{x}$ and directions of the corresponding residual error $\boldsymbol{\xi}$. This associates the error with the descriptor and helps us correct for the coding artifacts. We demonstrate later that the cross-term resulting from this formulation is very informative if combined with self-tensors as proposed in section 3.3.

4 Pooling the Low-level Descriptors

Recent advances in visual categorisation resulted in a coder-free approach to semantic segmentation [35]. Such a method employs the autocorrelation matrix formed by Average pooling the outer products of the local image descriptors. Hence, the coding step is bypassed. In our evaluations, this approach performed well on the Caltech101 dataset whilst the results on challenging PascalVOC07 were less competitive. We go beyond the second-order approach and propose Third-order Occurrence Pooling directly on the third-order autocorrelation tensor. The approach from [35]

employs the Log-euclidean framework to the autocorrelation matrix. This is not directly applicable to the higher-order tensors as there is no clear notion of their logarithm. Therefore, we propose a concept of *correlated burstiness* that helps us replace the matrix logarithm from [35] by Power Normalisation. Specifically, Second-order Occurrence Pooling can be performed by Singular Value Decomposition of the autocorrelation matrix, applying Power Normalisation to its eigenvalues, and reassembling the matrix. Third-order Occurrence Pooling is proposed to use Higher Order Singular Value Decomposition (HOSVD) [33], [34]. Power Normalisation is then performed on its eigenvalues from so-called core tensor and the autocorrelation tensor is reassembled. This extension results in significant improvements over the approach from [35] and is expressed in the following steps:

$$\boldsymbol{\phi}_n = \boldsymbol{x}_n,\ \forall n \in \mathcal{N} \tag{62}$$

$$\boldsymbol{H} = \underset{n\in\mathcal{N}}{\operatorname{avg}}\left(\boldsymbol{\Phi}_n\right),\ \boldsymbol{\Phi}_n = \otimes_r \boldsymbol{\phi}_n \tag{63}$$

$$\left(\boldsymbol{E}; \boldsymbol{A}_1, ..., \boldsymbol{A}_r\right) = \operatorname{hosvd}(\boldsymbol{H}) \tag{64}$$

$$\hat{\boldsymbol{E}} = g_e\left(\boldsymbol{E}\right) \tag{65}$$

$$\hat{\boldsymbol{H}} = \hat{\boldsymbol{E}} \times_1 \boldsymbol{A}_1 \cdots \times_r \boldsymbol{A}_r \tag{66}$$

$$\hat{h}_k = g_f\left(h_k^*\right),\ \boldsymbol{h}^* = u_:\left(\hat{\boldsymbol{H}}\right) \tag{67}$$

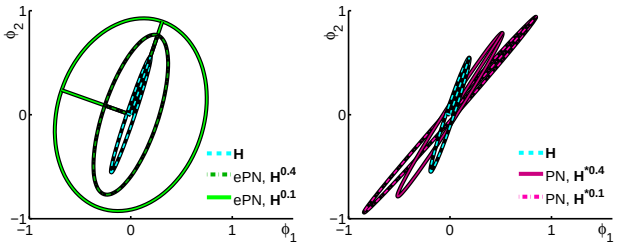
Equation (62) represents the coder-free step, however, we apply a PCA projection $\boldsymbol{\phi}_n = \operatorname{pcaproj}\left(\boldsymbol{x}_n\right)$ to $\boldsymbol{x}_n \in \mathbb{R}^D$ and obtain $\boldsymbol{\phi}_n \in \mathbb{R}^K$ such that $K \leq D$ to reduce the size of image signatures. We use: i) no spatial information, ii) append it on the descriptor level (Spatial Coordinate Coding) to the PCA projection, or iii) add Spatial Pyramid Matching by equation (56), *e.g.* $\boldsymbol{\phi}_n^{(1)} = \operatorname{pcaproj}\left(\boldsymbol{x}_n\right)$, $\boldsymbol{\phi}_n^{(2)}$ is from equation (57).

Equation (63) performs Average pooling as discussed in section 2.1. In detail, the higher-order autocorrelation tensor $\boldsymbol{H} \in \mathbb{R}^{K^r}$ (an r^{th}-order equivalent of the autocorrelation matrix) is computed in this equation by averaging over tensors $\boldsymbol{\Phi}_n \in \mathbb{R}^{K^r}$ of order r formed from *e.g.* $\boldsymbol{\phi}_n = \operatorname{pcaproj}\left(\boldsymbol{x}_n\right)$ given $n \in \mathcal{N}$.

Equations (64-66) and (67) represent two stage pooling that performs the eigenvalue- and coefficient-wise corrections such as Power Normalisation, respectively. We first detail the equations in this model and then explain the rationale behind the eigenvalue correction.

Equation (64) performs HOSVD with the operator $\operatorname{hosvd} : \mathbb{R}^{K^r} \rightarrow \left(\mathbb{R}^{R^r}; \mathbb{R}^{K\times R}, ..., \mathbb{R}^{K\times R}\right)$. It takes the higher-order autocorrelation tensor $\boldsymbol{H} \in \mathbb{R}^{K^r}$ and outputs the core tensor $\boldsymbol{E} \in \mathbb{R}^{R^r}$ of eigenvalues and the orthonormal factor matrices $\boldsymbol{A}_1, ..., \boldsymbol{A}_r \in \mathbb{R}^{K\times R}$ that are thought of as the principal components in each mode $1, ..., r$. If $R < K$, HOSVD becomes the truncated decomposition [34], thus, we decided to keep $R = K$.

Equation (65) performs eigenvalue-wise pooling $g_e : \mathbb{R}^{R^r} \rightarrow \mathbb{R}^{R^r}$ by applying element-wise corrections to eigenvectors from the core tensor. Note that even for symmetric $\boldsymbol{H}$ the eigenvalues may be negative. We use Power Normalisation, *e.g.* $g_e\left(\boldsymbol{E}\right) = \operatorname{sgn}(\boldsymbol{E})\left|\boldsymbol{E}\right|^\gamma$.



(*a*) (*b*)

Fig. 8. Whitening of the autocorrelation matrix $\boldsymbol{H}$. (*a*) The eigenvalue- and (*b*) coefficient-wise Power Normalisation steps (*ePN*) and (*PN*) are shown. See $\boldsymbol{H}^{0.4}$, $\boldsymbol{H}^{0.1}$, $\boldsymbol{H}^{*0.4}$, and $\boldsymbol{H}^{*0.1}$, $(^*)$ is the element-wise power. The axes of ellipses show their principal components.

Equation (66) reassembles the pooling corrected higher-order autocorrelation tensor $\hat{\boldsymbol{H}} \in \mathbb{R}^{K^r}$ by applying so-called k-mode product $\times_k$ (detailed in [34]) to tensors $\hat{\boldsymbol{E}}$ and $\boldsymbol{A}_1$, ..., $\boldsymbol{A}_r$, where $k = 1, ..., r$.

Equation (67) uses the previously defined operator $u_:$ to remove the redundant coefficients from the symmetric tensor $\hat{\boldsymbol{H}}$ as the symmetry of $\boldsymbol{H}$ is preserved. Next, the coefficient-wise pooling correction is applied to $\boldsymbol{h}^*$ of dimensionality $K^{(r)}$ for each h_k^* such that $k = 1, ..., K^{(r)}$. Note that $K^{(r)}$ is defined in section 2. Operator $g_f : \mathbb{R} \rightarrow \mathbb{R}$ performs Power Normalisation $g_f\left(h\right) = \operatorname{sgn}(h)\left|h\right|^\gamma$. The MaxExp correction given on the left-hand side of equation (13) can be also adapted to work with negative values and $h_k^* > 1$.

The normalisation in equation (3) is applied to $\hat{\boldsymbol{h}}$.

4.1 Two Stage Pooling

The burstiness is defined in [23] as "*the property that a given visual element appears more times in an image than a statistically independent model would predict*". The Analytical pooling operators have been advocated as a remedy to this phenomenon [11]. Moreover, corrections such as Power Normalisation: i) are recognised to act on Average pooling by correcting its output, ii) they act similarly to the probability of *at least one particular visual word being present in an image* (MaxExp from section 1.3), iii) they are applied to each visual element separately (these elements are assumed to be i.i.d.), therefore iv) they can be also thought of as performing whitening on the i.i.d. visual elements. We argue that the i.i.d. assumption does not hold in the real data and propose to simply perform Power Normalisation, MaxExp, or a similar correction on the eigenvalues from HOSVD (or SVD if $r = 2$) instead. For instance, local image descriptors extracted from a brick wall result in vertical and horizontal orientations of gradients that co-occur due to the underlying texture pattern. Thus, it makes sense to treat the burstiness of these gradients by correcting a principal component associated with them rather than process them separately. Figures 8 (a, b) show the difference between the eigenvalue (ePN) and coefficient-wise

(PN) Power Normalisation. The autocorrelation matrix $\boldsymbol{H}$ built from $2D$ features ϕ was used. Its principal components show that ϕ are correlated. The principal components of $\boldsymbol{H}^{0.4}$ and $\boldsymbol{H}^{0.1}$ show the data being whitened to a desired degree (the ellipses become more isotropic). On the contrary, element-wise Power Normalisation (*PN*) fails to whiten the correlated data.

Lastly, the element-wise operator g_f helps fine-tune the correction. The two stage pooling can be applied to the mid-level features or the low-level descriptors.

5 Experimental Section

Uni-modal First-, Second-, and Third-order Occurrence Pooling are compared to FV and VLAT in section 5.2. The coding and pooling are evaluated in sections 5.3 and 5.6. Experiments on Bi-modal Second-order Occurrence Pooling are in section 5.4. Second- and Third-order Occurrence Pooling for the low-level descriptors are compared in section 5.5. The PascalVOC07 [58], Caltech101 [59], Flower102 [42], ImageCLEF11 [46], 15 Scenes [30], and PascalVOC10 Action Recognition [58] sets are used in evaluations.

5.1 Experimental Arrangements and Datasets

The PascalVOC07 [58] set consists of 20 classes of objects of varied nature, *e.g. human*, *cat*, *chair*, *train*, *bottle*. This is a challenging collection of images with objects that appear at variable scales and orientations, often in difficult visual contexts and backgrounds, being frequently partially occluded. The training, validation, and testing splits are provided. The Caltech101 [59] set consists of 101 classes represented by objects which are aligned to the centres of images as well as a separate background class. The evaluations are performed with 15 and 30 training images per class. The Flower102 [42] set of 102 flower classes was used for further evaluations. A single split into the training and testing sets is supplied for this corpus. ImageCLEF11 Photo Annotation [46] is a challenging collection of images represented by 99 concepts of a varied nature, including complex topics, *e.g. party life*, *funny*, *work*, *birthday*. Unlike sets of objects, this challenge aims at annotation labels that correspond to human-like understanding of a scene [47], [60]. Only the visual annotation was used for this dataset. The 15 Scenes dataset [30] consists of 15 classes of indoor and outdoor scenes, *e.g. bedroom*, *kitchen*, *coast*. They contain from 200 to 400 images each. The PascalVOC10 Action Recognition [58] set (PascalVOC10AR) provides bounding boxes which delineate instances of human actions. There are 9 categories in total, *e.g. phoning*, *reading*, *using computer*. To best use the images in ImageCLEF11 and PascalVOC10AR, their training sets were doubled by left-right flipping training images [25]. Table 1 lists the experimental parameters for all datasets and the best results from the literature.

Dictionaries. Online Dictionary Learning was used to train dictionaries for Sparse Coding [61]. Dictionary learning proposed for Approximate Locality-constrained Linear Coding [15] was used for the LLC coder. Furthermore, we adapted such a method to work with Approximate Locality-constrained Soft Assignment as it outperformed LcSA with dictionaries formed by k-means. Size-wise, we used between 4K to 40K for First-, 300 to 1600 for Second-, and 100 to 200 for Third-order Occurrence Pooling. Fisher Vector Encoding [19], [20], [32] and Vector of Locally Aggregated Tensors [21] were used in comparisons, GMM and k-means dictionaries with 64 to 4096 and 64 to 512 atoms were employed, respectively.

Descriptors. Opponent SIFT was extracted on dense grids. The grey scale components (128D) were used for uni-modal BoW. The colour components (144D) were additionally used for bi-modal BoW. PCA was applied for FV and VLAT (80D for the grey and 120D for the grey and opponent components). Lastly, PCA (60-120D) was employed to control the signature sizes when pooling the low-level descriptors (no coding).

Spatial bias. Spatial relations in images were exploited mainly by Spatial Coordinate Coding [31]

Dataset	Splits no.	Training+validation samples	Test samples	Total images	Dict. size	Descr. type	State-of-the-art results	
PascalVOC07	1x	2501+2510=5011	4952	9963	100-1600	Opp. SIFT	Sánchez [32]	66.3
Caltech101	10x	12+3=15/24+6=30 (per class)	rest	9144	300-800	SIFT	Yang [44], 30im.	84.3
Flower102	1x	1020+1020=2040	6149	8189	300-1600	Opp. SIFT	Awais [53]	80.3
ImageCLEF11	1x	6K+2K=8K (+8K flip)	10K	18K (+8K)	800	Opp. SIFT	Binder [49]	38.8
15Scenes	5x	60+40=100 (per class)	rest	4485	400-800	SIFT	Gao [16]	89.8
PascalVOC10AR	1x	301+307=608 (+608 flip)	613	1221 (+608)	400-800	Opp. SIFT	Yao [54]	65.1

	Descr. interval (px)	Radii (px)	Descr. per img.	Coding	Spatial/other schemes	Order	Kernel types	State-of-the-art results	
PascalVOC07	4,6,8,10,12,14,16	12,16,24,32,40,48,56	19420	SC/LLC/LcSA/none	none/SCC/SPM*/DoPM*	1*,2,3	linear	Zhou [17]	64.0
Caltech101	4,6,8,10	16,24,32,40	5200	SC/none	SCC/SPM*	1*,2	linear	Kulkarni, Li [55]	83.3
Flower102	6,9,12,15	16,24,32,40	14688	SC	SCC/DoPM*	1*,2	linear	Zhang [56]	76.9
ImageCLEF11	4,6,8,10,12,14,16	12,16,24,32,40,48,56	19642	SC	SCC	2,3	linear/χ^2_{RBF}	Koniusz [11]	38.4
15Scenes	3,4,6,8,10,12,14,16	10,12,16,24,32,40,48,56	12650	SC/none	none/SCC/SPM	2,3	linear	Avila [52]	88.9
PascalVOC10AR	4,6,8,10,12,14,16	12,16,24,32,40,48,56	5660	SC/none	none/SCC/SPM	2,3	linear	Yao [57]	64.6

TABLE 1 (*) the first-order BoW with SPM/DoPM is used for comparisons - it was proposed in [31], [11]
Summary of the datasets, descriptor parameters, and various experimental details.

described in section 3.1. Spatial [30] and Dominant Angle [31], [11] Pyramid Matching were additionally used to: i) obtain comparative results on the standard BoW (first-order), ii) evaluate the proposed special cases of SPM and DoPM given in section 3.4. This special case of SPM was also used when pooling the low-level descriptors as explained in section 4. SPM used 3 levels of coarseness with 1x1, 1x3, 3x1, and 2x2 grids on PascalVOC07, PascalVOC10AR and 15 Scenes, and 4 levels with 1x1, 2x2, 3x3, and 4x4 grids on Caltech101. DoPM was used to exploit dominant edge bias given 5 levels of coarseness with 1, 3, 6, 9, and 12 grids on PascalVOC07, and 3 levels with 1, 2, and 3 grids on Flower102. Comparisons on the standard BoW (first-order) employed either SCC, SPM, or DoPM. By default, all experiments on DoPM used the descriptor coordinates appended at the descriptor level (SCC). Applying ordinary SPM to Second-order Occurrence Pooling on BoW performed worse than SCC, produced extremely large signatures, thus it is rarely reported on. Similar findings were presented in [32] for FV combined with SCC rather than SPM. Thus, we use FV and VLAT with SCC.

Coding and Pooling. We used SC for the most of experiments except for additional demonstrations of Second-order Occurrence Pooling with LLC and LcSA. The pooling operator $@n$ was used throughout experiments on BoW. A brief comparison on Max-pooling, MaxExp, and Power Normalisation is provided. Then, the two stage pooling from section 4.1 is evaluated on the mid- and low-level features. FV and VLAT were combined with Power Normalisation only as other operators are not directly applicable here. All comparative results on the standard BoW (first-order) used SC with the $@n$ operator. The coding and pooling parameters we determined by cross-validation.

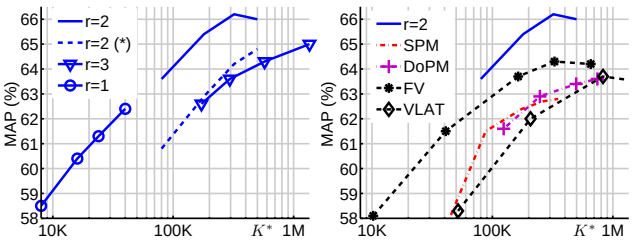
Kernels. Linear kernels $Ker_{ij} = (\boldsymbol{h}_i)^T \cdot \boldsymbol{h}_j$ were used, where $\boldsymbol{h}_i, \boldsymbol{h}_j \in \mathbb{R}^K$ are image signatures for $i, j \in \mathcal{I}$. χ^2 merged with RBF (χ^2_{RBF}) defined as $Ker_{ij} = exp\,[-\rho^2 \sum_k (h_{ki} - h_{kj})^2/(h_{ki} + h_{kj})]$ was used additionally on ImageCLEF11, $1/\rho$ is the RBF radius.

Classifiers. Multi-label KDA [37] was applied to PascalVOC07 and ImageCLEF11, as it performs well on these sets [37], [48]. Mean Average Precision [37] (MAP) is used to report their performance. Multi-class KDA [37] was applied to Clatech101, Flower102, 15 Scenes, and PascalVOC10AR. Mean Accuracy is reported on the first two sets, MAP on the rest.

5.2 Evaluating Uni-modal Bag-of-Words for First-, Second-, and Third-order Occurrence Pooling

This section presents how BoW described in section 2 performed in a practical classification scenario given order $r = 1$, 2, and 3, and the grey scale SIFT. Note that $r = 1$ renders BoW model from section 2 to be equivalent to the standard model in section 1.1.

Figure 9 (a) compares the classification performance of the proposed method for various orders r on the PascalVOC07 set (SCC is used). Second-order Occurrence Pooling is shown to outperform the first- and third-order cases. It attains 65.4%, 66.2%, and 66.0% MAP for $K = 600$, 800, and 1000 dictionary atoms that result in the signature lengths $K^* = 180300$, 320400, and 500500, respectively. Next, First-order Occurrence Pooling scores respectable 62.4% MAP for $K = K^* = 40000$ atoms (this is also the signature length). However, the coding step is computationally prohibitive for large visual dictionaries. It takes 815 and 1.5 seconds to code 1000 descriptors on a single 2.3GHz AMD Opteron core given $K = 40000$ and $K = 800$ atoms, respectively. Third-order Occurrence Pooling yields 65% MAP for $K = 200$ atoms resulting in the signature length $K^* = 1353400$. Our experiments suggest that the second-order case yields the highest results and provides an attractive trade-off between the tractability of coding and the signature lengths. Finally, Second-order Occurrence Pooling without any spatial information attains 64.8% MAP for $K = 1000$ atoms. This demonstrates the benefit of SCC.

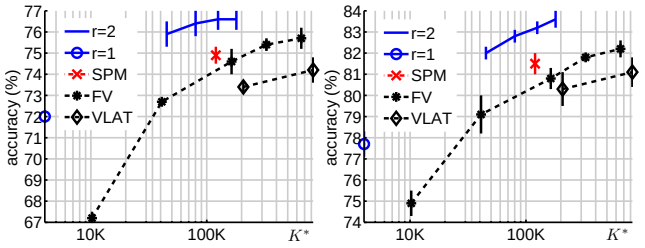


(*a*) VOC07, $r = 1, 2, 3$ (*b*) VOC07, $r = 2$ vs. rest

Fig. 9. Performance of Higher-order Occurrence Pooling compared to various approaches on the PascalVOC07 set. Results were plotted as functions of the signature length K^*. (*a*) First-, Second-, and Third-order Occurrence Pooling $r = 1, 2, 3$ with Spatial Coordinate Coding. Asterisk (*) denotes the case of order $r = 2$ without any spatial information. (*b*) The case of order $r = 2$ compared to SPM and DoPM ($r = 1$). Furthermore, results on FV and VLAT were also plotted.

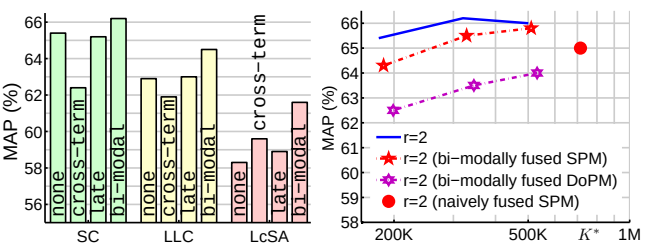
Figure 9 (b) compares Second-order Occurrence Pooling ($r = 2$, SCC is used) to the standard BoW ($r = 1$) combined with SPM and DoPM, respectively. FV and VLAT combined with SCC are also evaluated. BoW ($r = 1$) with SPM attains 62.8% MAP for $K = 32000$ atoms and results in the signature length $K^* = 352000$. BoW ($r = 1$) with DoPM yields 63.6% MAP and outperforms SPM by 0.8% for $K = 24000$ atoms and the signature length $K^* = 744000$. This is comparable to VLAT that attains 63.7% MAP for the signature length $K^* = 829440$. FV yields 64.3% MAP given the signature length $K^* = 327680$. With 66.2% MAP, Second-order Occurrence Pooling outperforms FV by 1.9% MAP for the comparable signature length.

The classification performance of Second-order Occurrence Pooling on the Caltech101 set is compared in figure 10 to the standard BoW ($r = 1$) combined with



(*a*) Caltech101 (15 samples) (*b*) 30 samples/class

Fig. 10. Performance of Second-order Occurrence Pooling compared to various approaches on the Caltech101 set. Results were plotted as functions of the signature length K^*. Standard BoW (order $r=1$, SCC and SPM), FV, and VLAT were evaluated on (*a*) 15, and (*b*) 30 training images per class, respectively.



(*a*) VOC07, Residual Desc. (*b*) VOC07, fusing SPM

Fig. 11. Evaluation of Bi-modal Second-order Occurrence Pooling (PascalVOC07). (*a*) Bars (*none*) show results for SC, LLC, and LcSA coders ($r=2$, 600 atoms). Residual Descriptors from section 3.5 were fused by the late fusion (*late*) from section 3.2 (note little improvement). The cross-term (section 2.1) is not the best on its own either. A larger gain is shown for Bi-modal Second-order Occurrence Pooling (*bi-modal*). (*b*) Special case SPM and DoPM proposed in section 3.4 were fused by Bi-modal Second-order Occurrence Pooling (*bi-modally*). SPM applied directly to the mid-level features (*naively*) is also evaluated for $r=2$.

SCC and SPM, respectively, and to FV and VLAT. We employ SCC for all methods except for SPM.

Figure 10 (a) provides evaluations on 15 training images per class. BoW ($r=1$) with SCC yields $72\pm0.3\%$ accuracy for $K=K^*=4000$ atoms (this is also the signature length). This offers very compact signatures and a good performance. BoW ($r=1$) with SPM yields $74.9\pm0.4\%$ accuracy for $K=4000$ atoms and the signature length $K^*=120000$. This represents a slight improvement over FV that yields $74.6\pm0.6\%$ accuracy given the signature length $K^*=163840$. Lastly, Second-order Occurrence Pooling yields $76.6\pm0.5\%$ given $K=500$ atoms and the signature length $K^*=125250$. This is a 2% improvement over FV given the comparable signature lengths. FV and VLAT yield $75.7\pm0.5\%$ and $74.2\pm0.6\%$ accuracy at best.

Figure 10 (b) provides evaluations given 30 training images per class. The comparison arrangements remain identical to those presented above. Second-order Occurrence Pooling scores $83.6\pm0.4\%$ accuracy given $K=600$ atoms and the signature length $K^*=180300$. This is a 2.8% improvement over FV that scores $80.8\pm0.5\%$ accuracy for the comparable signature length $K^*=163840$. BoW ($r=1$) with SPM yields $81.5\pm0.4\%$ accuracy for $K=4000$ atoms and the signature length $K^*=120000$. This also represents a small gain of 0.7% over FV. BoW ($r=1$) with SCC yields $77.7\pm0.6\%$ accuracy. FV and VLAT yield $82.2\pm0.4\%$ and $81.1\pm0.7\%$ accuracy at best.

5.3 Evaluations of SC, LLC, and LcSA given Unimodal Second-order Occurrence Pooling

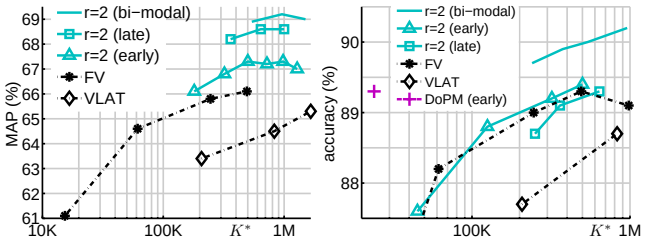
The coding step is now evaluated and demonstrated to have a significant impact on the performance of Second-order Occurrence Pooling. Evaluations of coding in the standard BoW ($r=1$) are provided in [11].

Figure 11 (a) demonstrates results on SC, LLC, and LcSA, all obtained on the PascalVOC07 set for $K=600$ dictionary atoms that resulted in the signature lengths $K^*=180300$. Bars called (*none*) show that SC yields 65.4%, LLC 62.9%, and LcSA 58.3% MAP. This is in agreement with the observation that the lower the quantisation loss of a coder is, the better the classification results are. We evaluated ξ^2 according to equation (59) for a subset of descriptors, summed over the individual ξ^2 for each descriptor, and observed that $\xi^2_{SC}<\xi^2_{LLC}<\xi^2_{LcSA}$. In section 5.4, we present results for Residual Descriptor from section 3.5 which exploits such quantisation effects. Finally, we note that the gap in performance between SC and LcSA is 7.1% MAP. We expect that the worse the quantisation properties of a coder are, the more distorted the joint occurrences of visual words on the mid-level feature level become. The gap between SC and LcSA is much smaller for the standard BoW ($r=1$) with SPM [11].

5.4 Evaluations of Bi-modal Bag-of-Words for Second-order Occurrence Pooling

This section presents the classification performance for BoW given order $r=2$ described in section 3 and illustrated in figure 6. The modalities to fuse are: i) the grey scale SIFT and Residual Descriptor proposed in section 3.5, ii) the grey scale SIFT and special case SPM and DoPM, respectively, as proposed in section 3.4, iii) the grey scale and colour components of SIFT.

We evaluate the following fusion schemes: a) Bi-modal Second-order Occurrence Pooling ($r=2$) outlined in section 3.3 and referred to as *bi-modal* in the plots, b) the early fusion explained in section 3.1 and referred to as *early*, c) the late fusion explained in section 3.2 and referred to as *late*. Also, we evaluate FV and VLAT, both using the early fusion. Moreover, for the proposed bi-modal fusion, equation (51) entails three terms $\hat{h}^s_k$ that are weighted by $w^s=\binom{r}{s}^{\frac{1}{2}}(1-\beta)^{\frac{s}{2}}\beta^{\frac{r-s}{2}}$ in equation (51), where $s=0,\ldots,2$. If $w^2\ll w^0$ or $w^0\ll w^2$, we remove $\hat{h}^2_k$ or $\hat{h}^0_k$ to shorten the signature as, either $\hat{h}^2_k$ or $\hat{h}^0_k$ becomes negligible.



(*a*) VOC07, fusing colour (*b*) Flower102, fusing col.

Fig. 12. Evaluation of Bi-modal Second-order Occurrence Pooling (*bi-modal*). The grey and opponent components of SIFT were fused in various ways given (*a*) PascalVOC07 and (*b*) Flower102 sets. The overall signature length K^* is indicated. Results for the early and late fusions from sections 3.1 and 3.2 are also provided for order $r=2$. Moreover, the early fusion was applied to FV, VLAT, and DoPM ($r=1$).

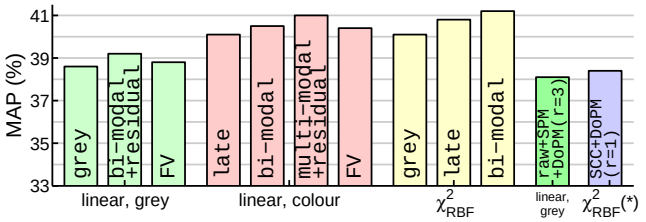


Fig. 13. Evaluation of Uni-modal (*grey*), Bi-modal (*bi-modal*) and Multi-modal (*multi-modal*) Second-order Occurrence Pooling (ImageCLEF11) given the linear and χ^2_{RBF} kernels. Various fusions of grey and colour SIFT components are shown (*colour*). FV uses early fusion (the colour case). Residual Descriptors (*residual*) use bi- and multi-modal fusions. For comparison, a coder-free approach for $r=3$ that uses SPM and DoPM is given (*raw+SPM+DoPM*). A result on SC combined with SCC and DoPM (*) for $r=1$ was taken from [11].

Residual Descriptor is combined with SC, LLC, and LcSA by the bi-modal and late fusions on the PascalVOC07 set given $K=600$ dictionary atoms. Figure 11 (a) shows the baseline performance for Second-order Occurrence Pooling (*grey*). The late fusion (*late*) of the Residual Descriptor resulted in loss for SC and a marginal improvement for LLC and LcSA. This is expected as the residual codes are not associated in such a fusion neither with the corresponding descriptors nor the mid-level features (see section 3.5). The cross-term is not sufficient on its own either to obtain top scores as equations (45) and (47) suggest that the self-tensors are also needed. However, capturing co-occurrences of Residual Descriptors with the corresponding features and performing the fusion (*bi-modal*) as described in section 3.3 results in a significant gain of 0.8%, 1.6%, and 3.3% MAP for SC, LLC, and LcSA, respectively. The greater the quantisation loss of the coder is, the larger the benefits from using RD are. Note also that SC attains 66.2% MAP with the overall signature length $K^*=265356$. The same score was presented in section 5.2 for the uni-modal second-order case with a longer signature length $K^*=320400$.

SPM and DoPM (the special case) proposed in section 3.4 were fused by Bi-modal Second-order Occurrence Pooling on the PascalVOC07 set. Figure 11 (b) demonstrates their performance (*bi-modally*) compared to SPM combined naively with Second-order Occurrence Pooling (*naively*). Bi-modally fused SPM scores 65.8% MAP giving a 0.8% improvement over the naively fused SPM which yields only 65.0% MAP. It also produces the signatures of length $K^*=510500$ (bi-modal case) compared to much longer 714780 (naive case). However, the uni-modal second-order case ($r=2$) from section 5.2 that employs SCC scores the highest. Section 3.4 explains that naive SPM turns the standard BoW ($r=1$) into a simplified second-order model. Once BoW ($r=1$) is extended by the co-occurrence statistics, the benefit of SPM becomes less obvious.

Fusing colour. The grey and opponent colour components of SIFT are fused for the following evaluations.

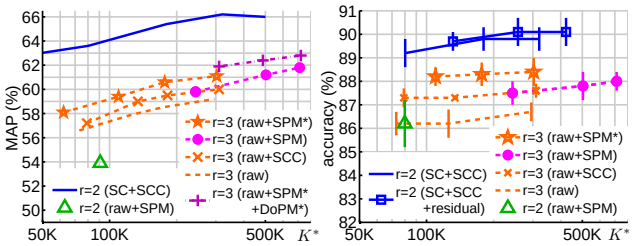
Figure 12 (a) compares the fusing schemes on the PascalVOC07. The proposed bi-modal fusion (*bi-modal*) scores 69.2% MAP for $K=800$ dictionary atoms. Note that one grey and one colour dictionary are used. This produces the signatures of length $K^*=960400$ as we removed all $\hat{h}_k^2$ as explained earlier in this section. The late fusion scores 68.6% MAP at its best for $K^*=640800$. This amounts to a 0.6% decline. The early fusion scores 67.3% MAP for $K=1000$ atoms that result in signature length $K^*=500500$. Lastly, FV and VLAT yield 65.6% and 64.8% MAP.

Figure 12 (b) details results on the Flower102 set. The bi-modal fusion (*bi-modal*) scores 90.2% MAP for $K=800$ dictionary atoms and the signature length $K^*=960400$. The late fusion scores 89.3% MAP at its best for $K^*=640800$. This amounts to a 0.9% decline over the bi-modal approach. The early fusion scores 89.4% MAP for for $K=1000$ atoms that result in the signature length $K^*=500500$. FV and VLAT yield 89.3% and 88.7% MAP. The standard BoW ($r=1$) with DoPM yields 89.3% MAP for $K=4000$ atoms that result in the signature length $K^*=24000$. This represents a good trade-off between the classification performance and the length of signatures.

Figure 13 presents performance of Uni-, Bi- and Multi-modal Second-order Occurrence Pooling on the ImageCLEF11 set. As ImageCLEF11 includes many abstract topics, *e.g. party life*, we compare the classification performance of linear and χ^2_{RBF} kernels.

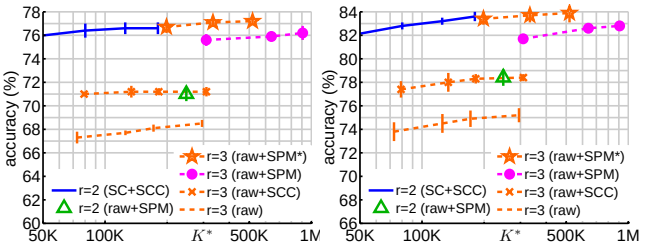
The experiments on the grey SIFT given the linear kernels (*linear, grey*) include uni-modal, bi-modal (Residual Descriptor is a second modality), and Fisher Vector Encoding approaches (*grey*, *bi-modal+residual*, and *FV*). They score 38.6%, 39.2%, and 38.8% MAP. $K=1000$ atoms were used (first two results). FV used $K=4096$ components. This produced the signature lengths $K^*=500500$, 628500, and 655360.

The fusion experiments on the grey and colour



(*a*) VOC07 (*b*) 15 Scenes

Fig. 14. Third-order Occurrence Pooling (r=3) on the low-level descriptors with no spatial cues (*raw*) was fused with SCC (*raw+SCC*), ordinary SPM (*raw+SPM*), bi-modally fused SPM (*raw+SPM**), and multi-modally fused SPM and DoPM (*raw+SPM*+DoPM**) on (*a*) PascalVOC07 and (*b*) 15 Scenes. Second-order Pooling ($r=2$) from [35] uses ordinary SPM (*raw+SPM*). Second-order Occurrence Pooling ($r=2$) on Sparse Coding uses Spatial Coordinate Coding (*SC+SCC*). It was then bi-modally fused with Residual Descriptor (*residual*). K^* is the signature length.



(*a*) Caltech101 (15 samples) (*b*) 30 samples/class

Fig. 15. Evaluation of Third-order Occurrence Pooling on the low-level descriptors. The Caltech101 set with (*a*) 15, and (*b*) 30 training images per class were used. The legend from figure 14 is used. Note the error bars.

modalities given the linear kernels (*linear, colour*) include the late, bi-modal, and multi-modal (Residual Descriptor is a third modality) methods (*late*, *bi-modal*, and *multi-modal+residual*). Fisher Vector Encoding (*FV*) uses the early fusion. These methods score 40.1%, 40.5%, 41.0%, and 40.4% MAP. $K=1000$ and 400 atoms were used per grey and colour modalities while $K=4096$ is used for FV. This produced the signature lengths K^*=580700, 900500, 1028500, and 983040.

A further improvement is observed given χ^2_{RBF} kernels (χ^2_{RBF}). This highlights χ^2_{RBF} as a good choice for datasets with complex topics. The uni-modal, late, and the bi-modal approaches (*grey*, *late*, and *bi-modal*) score 40.1%, 40.8%, and 41.2% MAP, respectively. This compares favourably to the late fusion of SCC and DoPM ($\chi^2_{RBF}(^*)$) given BoW (r=1) in [11].

5.5 Evaluating Low-level Descriptor Pooling

This section presents results on Third-order Occurrence Pooling of the low-level descriptors, detailed in section 4. Also, Second-order Pooling on the low-level descriptors from [35] is evaluated. These coder-free methods are compared to Second-order Occurrence Pooling on the mid-level features representing the BoW approach. Our third-order method uses the two stage pooling (eigenvalue- and coefficient-wise Power Normalisation). The latter step is often disabled as it is only for fine-tuning. Method [35] uses the matrix logarithm and coefficient-wise Power Normalisation $0<\gamma\leq 1$. The second-order BoW uses the @n operator. The grey features are used unless stated otherwise.

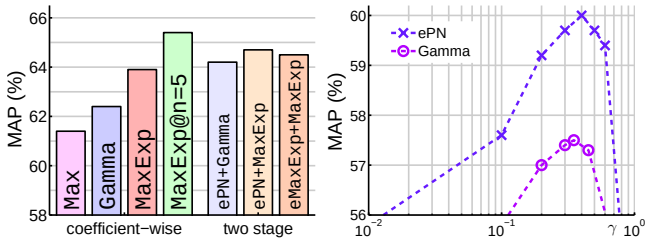
Figure 14 (a) shows that Second-order Occurrence Pooling ($r=2$) with the SC coder (*SC+SCC*) outperforms the coder-free methods on PascalVOC07. The result from section 5.2 indicates 66.2% MAP on this method. From the coder-free approaches, Third-order Occurrence Pooling ($r=3$) scores 61.1% MAP for bi-modally fused SPM (*raw+SPM**), 61.8% for ordinary SPM (*raw+SPM*), 60.0% for Spatial Coordinate Coding (*raw+SCC*), 59.2% for no spatial fusion (*raw*), and 62.8% MAP for multi-modally fused SPM and DoPM (*raw+SPM*+DoPM**). Second-order Pooling ($r=2$) with SPM (*raw+SPM*) from [35] yields 54.0% MAP. The signature lengths are K^*=320400, 713064, 302621, 310124, 295240, 721764, and 90816, respectively. For comparable signature lengths, bi-modal SPM and multi-modal SPM and DoPM outperformed ordinary SPM validating the benefit of our tensor level fusion.

Figure 14 (b) presents similar trends on 15 Scenes. Second-order Occurrence Pooling ($r=2$) with the SC coder and bi-modally fused Residual Descriptor (*SC+SCC+residual*) yields $90.1\pm0.6\%$ MAP. The coder-free Third-order Occurrence Pooling methods score $88.4\pm0.6\%$ (*raw+SPM**), $88.0\pm0.4\%$ (*raw+SPM*), $87.6\pm0.3\%$ (*raw+SCC*), and $86.7\pm0.4\%$ MAP (*raw*). Second-order Pooling from [35] gives $86.2\pm1.0\%$ MAP.

Figures 15 (a, b) show Third-order Occurrence Pooling on the low-level descriptors scoring $77.2\pm0.2\%$ and $83.9\pm0.8\%$ MAP on Caltech101 given 15 and 30 training images per class ($K^*=518665$, bi-modally fused SPM), and $76.2\pm0.6\%$ and $82.8\pm0.4\%$ MAP ($K^*=907536$, ordinary SPM). These results are similar to Second-order Occurrence Pooling with the SC coder reaching $76.6\pm0.5\%$ and $83.6\pm0.4\%$ MAP ($K^*=125250$), respectively. This suggests that datasets with little clutter and well aligned objects of fixed scale can be classified without coding if larger signatures are used. Second-order Pooling from [35] yields $71.0\pm0.6\%$ and $78.4\pm0.7\%$ MAP giving a 6% decline.

Not included in the plots, Second-order Occurrence Pooling (SC with SCC) and Third-order Occurrence Pooling (low-level descriptors with bi-modally fused SPM) score 65.0% and 63.5% MAP on PascalVOC10 AR given K^*=180300 and 302621. With the late colour fusion, these methods yield 66.5% and 66.0% MAP.

Lastly, Third-order Occurrence Pooling ($r=3$) on the low-level descriptors fused multi-modally with SPM and DoPM is evaluated on ImageCLEF11 in figure 13 (*raw+SPM+DoPM*). It scores 38.1% MAP.



(*a*) SC, $r=2$ (*b*) no coder, $r=3$

Fig. 16. Evaluation of the pooling operators on the PascalVOC07 set. (*a*) SC with a dictionary of 600 atoms was used. Max-pooling, Power Normalisation, MaxExp, and the $@n$ operators were combined with Second-order Occurrence Pooling. The eigenvalue Power Normalisation and MaxExp (*ePN* and *eMaxExp*) corrections (first stage) with coefficient-wise Power Normalisation and MaxExp (second stage) are shown. (*b*) Pooling the low-level descriptors: ePN vs. Gamma.

5.6 Evaluating the Pooling Operators

Below are the evaluations of pooling operators on the PascalVOC07 set. We use SCC for spatial information.

The classification results of Uni-modal Second-order Occurrence Pooling given SC with $K = 600$ visual words are shown in figure 16 (a). The best score of 65.4% MAP is attained by the $@n$ operator. Max-pooling yields 61.4% MAP. This 4% gap is consistent with evaluations in [11]. The proposed two stage pooling, eigenvalue- and coefficient-wise Power Normalisation (*ePN+Gamma*), scores 64.2% (1.8% increase over Gamma). Using MaxExp for the second stage (*ePN+MaxExp*) gives 64.7% (0.8% increase over MaxExp). We note that the noise limiting traits of the $@n$ operator, developed for ordinary BoW [11], also apply to Second-order Occurrence Pooling. Close scores of ePN and MaxExp suggest that occurrences of visual words from SC are largely statistically independent.

Figure 16 (b) verifies the impact of γ on whitening by eigenvalue Power Normalisation (*ePN*). Third-order Occurrence Pooling on the low-level descriptors is used. With coefficient-wise pooling switched off, ePN yielded 60.0% MAP outperforming coefficient-wise Power Normalisation (*Gamma*) which scored 57.5% MAP. This highlights that the choice of appropriate pooling is important, *e.g.* whitening has to be performed on tensors from the coder-free methods.

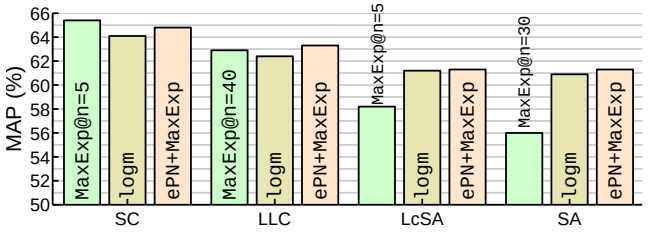
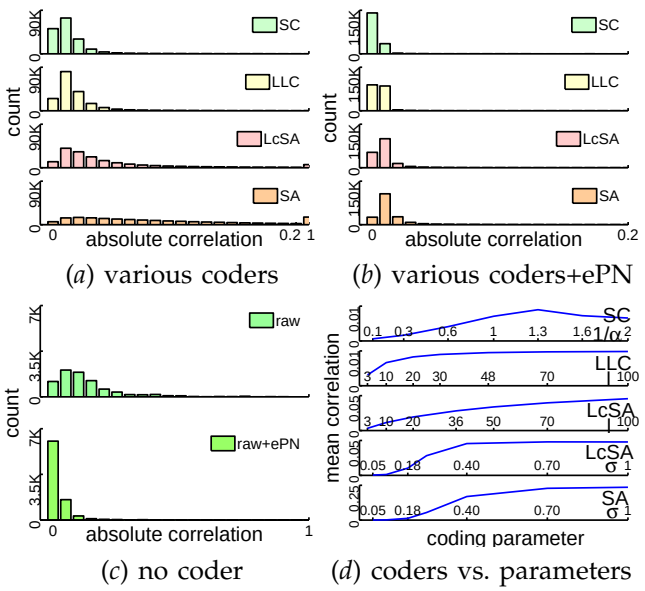


Fig. 17. Comparison of pooling operators $@n$, the matrix logarithm (*logm*), and the eigenvalue Power Normalisation combined with MaxExp (*ePN+MaxExp*), all on the PascalVOC07 set. Second-order Occurrence Pooling ($r=2$, 600 atoms), SCC, grey SIFT, and linear kernels were applied. Evaluations on SC, LLC, LcSA, and SA (Soft Assignment [7], [11]) coders reveal their varied ability to decorrelate the data.



(*a*) various coders (*b*) various coders+ePN

(*c*) no coder (*d*) coders vs. parameters

Fig. 18. Histogram of the absolute values of Pearson's correlation coefficients for the SC, LLC, LcSA, and SA coders (*a*) without and (*b*) with geometric pooling (ePN). The peaks show that $K = 600$ dimensions of mid-level features are decorrelated better by SC and LLC than LcSA or SA. Also, ePN decorrelates the data well for all coders. (*c*) Highly correlated coefficients of SIFT (*raw*) are decorrelated by ePN (*raw+ePN*). (*d*) Mean correlation w.r.t. the regularisation parameters α, l, and σ of coders. Note the stronger the regularisation is the more decorrelated the data becomes.

Figure 17 reveals a vital difference between the coders used in the experiments. We note that Sparse Coding with the $@n$ operator outperforms marginally the geodesic distance operators for SPD matrices such as the matrix logarithm (*logm*), used in [35] on low-level descriptors, and the eignevalue Power Normalisation combined with the element-wise MaxExp operator (*ePN+MaxExp*). This strongly suggests that SC is able to largely decorrelate mid-level feature responses to the visual words on the linear subspaces it selects. Therefore, coefficient-wise pooling like the $@n$ operator can be successfully employed for the SC coder. Similar observations hold in case of LLC. However, coefficient-wise pooling with the LcSA coder scores only 58.3% while the geodesic distance operators attain 61.3% MAP. This significant improvement of 3% MAP can be ascribed to decorrelation between dimensions of the mid-level features. It is demonstrated in [62] that LcSA does not take into account dependency between visual words selected for a to-be-coded descriptor whilst LLC does. Moreover, Soft Assignment [7], [11], which assumes no explicit locality constraint as opposed to LcSA [10], [11], also attains 61.3% MAP when combined with

the geodesic distance operators. As the $@n$ operator scores less than the geodesic distance counterparts on LcSA and SA, we conclude that the locality constraint addresses the decorrelation process only marginally.

The above conclusions are supported by the simulations in plot 18. SIFT descriptors from 100 randomly selected images (PascalVOC07) were coded with the SC, LLC, LcSA, and SA coder, respectively. The coding parameters and dictionaries from evaluations in plots 16 and 17 were used. Then, histograms of the absolute values of Pearson's correlation coefficients were computed. Plot 18 (a) shows that most of $K = 600$ coefficients of mid-level features are weakly correlated given SC and LLC (large peaks to the left). LcSA decorrelates the data only partially while SA results in high correlation. Plot 18 (b) reveals that the ePN operator decorrelates the data irrespectively of coder. SC and LLC also perform decorrelation well themselves and hence the $@n$ operator can further whiten the mid-level representations without corrupting them. Plot 18 (c) shows that coefficients of the low-level descriptors can be decorrelated by ePN (no coder needed). Lastly, plot 18 (d) suggests that the regularisation terms in SC, LLC, LcSA, and SA act as surrogates of the decorrelation process.

6 Conclusions

This paper proposes a theoretically derived framework that extends Bag-of-Words with the second- or higher-order statistics computed on the mid-level features. We term these approaches as Second- and Higher-order Occurrence Pooling. According to our evaluations, Uni-modal Second-order Occurrence Pooling offers the best trade-off between the tractability of coding, the length of signatures, and the classification quality for the grey scale descriptors. It outperformed the standard BoW with various Pyramid Matching schemes, Fisher Vector Encoding, and Vector of Locally Aggregated Tensors. Evaluations were conducted in a common testbed on the PascalVOC07, Caltech101, ImageCLEF11, 15 Scenes, and PascalVOC10 AR sets. Comparison of various coding and pooling techniques highlights Sparse Coding and the $@n$ pooling operator as the best performers. Also, the role of decorrelation in BoW is explained.

Moreover, a coder-free Third-order Occurrence approach with a novel two stage pooling are proposed. This method can challenge coder-based BoW on simple datasets but it results in large signatures. We emphasise that the coder-based models perform better if their quantisation loss stays low and sparsity is maintained. This was evident in BoW comparisons on Second- and Third-order Occurrence Pooling. The latter model used a very small dictionary and performed worse. In contrast, coder-free Third-order Occurrence Pooling outperformed its second-order counterpart.

To benefit from the multi-modal nature of visual concepts, a bi-modal approach is formulated. We call it Bi-modal Second-order Occurrence Pooling. Numerous extensions to the multi-modal and higher-order variants are suggested. The proposed bi-modal approach highlights the need for cross-modal statistics. Their importance is demonstrated with extended Pyramid Matching schemes and Residual Descriptor exploiting the quantisation effects in coding.

Such a bi-modal variant is also demonstrated to outperform the outlined early and late fusions of the grey and colour features on standard BoW, Second-order Occurrence Pooling, Fisher Vector Encoding, and Vector of Locally Aggregated Tensors given the PascalVOC07, Flower102, and ImageCLEF11 sets. Table 2 lists the best results from our study.

Grey SIFT only	VOC 07	CLEF11	Calt. 101 (15 img.)	Calt. 101 (30 img.)	VOC 10AR	15 Scenes
Coder, $r=2$	Uni-modal					Bi-modal RD
	66.2	**40.1**	76.6±.5	83.6±.4	**65.0**	**90.1±.6**
No coder r=3	Multi-modal SPM+DoPM		Bi-modally fused SPM			
	62.7	38.1	**77.2±.2**	**83.9±.8**	63.5	88.4±.6
FV	64.3	38.8	75.7±.5	82.2±.4	-	-
VLAT	63.7	-	74.2±.6	81.1±.7	-	-
SCC ($r=1$)	62.4	-	72.0±.3	77.7±.7	-	-
SPM ($r=1$)	62.8	-	74.9±.4	81.5±.5	-	-
DoPM ($r=1$)	63.6	-	-	-	-	-

Grey + Opp. SIFT	VOC07	CLEF11	Flower102	VOC10AR
Bi-modal (coder, $r=2$)	**69.2**	**41.2**	**90.2**	-
Early (coder, $r=2$)	67.3	-	89.4	-
Late (coder, $r=2$)	68.6	40.8	89.3	**66.5**
Late (no coder, $r=3$)	-	-	-	66.0
FV	65.6	40.4	89.3	-
VLAT	64.8	-	88.7	-
DoPM ($r=1$)	-	-	89.3	-

TABLE 2
Summary of the best results from this study. See figures 9-15 for fair and exact comparisons.

References

[1] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *ICCV*, vol. 2, pp. 1470–1477, 2003.

[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," *ECCV Workshop on Statistical Learning in Computer Vision*, pp. 1–22, 2004.

[3] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *CVPR*, vol. 2, pp. 1150–1157, 1999.

[4] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[5] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "A Comparison of Color Features for Visual Concept Classification," *CIVR*, pp. 141–149, July 2008.

[6] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel Codebooks for Scene Categorization," *ECCV*, vol. 5304, pp. 696–709, 2008.

[7] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual Word Ambiguity," *PAMI*, 2010.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases." *CVPR*, 2008.

[9] P. Koniusz and K. Mikolajczyk, "Soft Assignment of Visual Words as Linear Coordinate Coding and Optimisation of its Reconstruction Error," *ICIP*, 2011.

[10] L. Lingqiao, L. Wang, and X. Liu, "In Defence of Soft-assignment Coding," *ICCV*, 2011.

[11] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of Mid-Level Feature Coding Approaches And Pooling Strategies in Visual Concept Detection," *CVIU*, 2012.

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient Sparse Coding Algorithms," *NIPS*, pp. 801–808, 2007.

[13] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear Spatial Pyramid Matching using Sparse Coding for Image Classification," *CVPR*, pp. 1794–1801, 2009.

[14] K. Yu, T. Zhang, and Y. Gong, "Nonlinear Learning using Local Coordinate Coding," *NIPS*, 2009.

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained Linear Coding for Image Classification," *CVPR*, 2010.

[16] S. Gao, I. W. Tsang, L. Chia, and P. Zhao, "Local Features Are Not Lonely - Laplacian Sparse Coding for Image Classification," *CVPR*, 2010.

[17] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image Classification using Super-Vector Coding of Local Image Descriptors," *ECCV*, pp. 141–154, 2010.

[18] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating Local Descriptors into a Compact Image Representation," *CVPR*, pp. 3304–3311, 2010.

[19] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," *CVPR*, vol. 0, pp. 1–8, 2007.

[20] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," *ECCV*, pp. 143–156, 2010.

[21] R. Negrel, D. Picard, and P.-H. Gosselin, "Compact Tensor Based Image Representation for Similarity Search," *ICIP*, 2012.

[22] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Generalized Histogram Intersection Kernel for Image Recognition," *ICIP*, pp. 161–164, 2005.

[23] H. Jégou, M. Douze, and C. Schmid, "On the Burstiness of Visual Elements," *CVPR*, pp. 1169–1176, 2009.

[24] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating Bag-of-Visual-Words Representations in Scene Classification," *MIR*, pp. 197–206, 2007.

[25] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods," *BMVC*, 2011.

[26] A. Coates and A. Ng, "The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization," *ICML*, pp. 921–928, June 2011.

[27] I. Tosic and P. Frossard, "Dictionary Learning," *SPM*, vol. 28, no. 2, pp. 27–38, 2011.

[28] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning Mid-Level Features for Recognition," *CVPR*, 2010.

[29] Y. Boureau, J. Ponce, and Y. LeCun, "A Theoretical Analysis of Feature Pooling in Vision Algorithms," *ICML*, 2010.

[30] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *CVPR*, vol. 2, pp. 2169–2178, 2006.

[31] P. Koniusz and K. Mikolajczyk, "Spatial Coordinate Coding to Reduce Histogram Representations, Dominant Angle and Colour Pyramid Match," *ICIP*, 2011.

[32] J. Sánchez, F. Perronnin, and T. E. de Campos, "Modeling the Spatial Layout of Images Beyond Spatial Pyramids." *PRL*, 2012.

[33] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Analysis and Applications*, vol. 21, pp. 1253–1278, 2000.

[34] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[35] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic Segmentation with Second-Order Pooling." *ECCV*, 2012.

[36] X. YU and Y.-J. ZHANG, "A 2-D Histogram Representation of Images for Pooling." *SPIE*, 2011.

[37] M. Tahir, J. Kittler, K. Mikolajczyk, F. Yan, K. van de Sande, and T. Gevers, "Visual Category Recognition using Spectral Regression and Kernel Discriminant Analysis," *ICCV Workshop on Subspace Methods*, 2009.

[38] J. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," ICSI, Tech. Rep. TR-97-021, 1997.

[39] D. Picard and P.-H. Gosselin, "Improving Image Similarity with Vectors of Locally Aggregated Tensors," *ICIP*, 2011.

[40] P. Koniusz and K. Mikolajczyk, "On a Quest for Image Descriptors Based on Unsupervised Segmentation Maps," *ICPR*, vol. 0, pp. 762–765, 2010.

[41] M. E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," *ICVGIP*, pp. 722–729, 2008.

[42] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," *ICVGIP*, Dec 2008.

[43] X.-T. Yuan and S. Yan, "Visual Classification with Multi-Task Joint Sparse Representation," *CVPR*, 2010.

[44] J. Yang, Y. Tian, L.-Y. Duan, T. Huang, and W. Gao, "Group-Sensitive Multiple Kernel Learning for Object Recognition," *TIP*, vol. 21, no. 5, pp. 2838–2852, 2012.

[45] F. Yan, K. Mikolajczyk, M. Barnard, H. Cai, and J. Kittler, "Lp Norm Multiple Kernel Fisher Discriminant Analysis for Object and Image Categorisation," *CVPR*, 2010.

[46] S. Nowak, K. Nagel, and J. Liebetra, "The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks," *CLEF*, 2011.

[47] M. J. Huiskes and M. S. Lew, "The MIR Flickr Retrieval Evaluation," *MIR*, pp. 39–43, 2008.

[48] M. A. Tahir, F. Yan, M. Barnard, M. Awais, K. Mikolajczyk, and J. Kittler, "The University of Surrey Visual Concept Detection System at ImageCLEF 2010: Working Notes," *ICPR*, 2010.

[49] A. Binder, W. Samek, and M. Kawanabe, "The joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF 2011 Photo Annotation Task: Working Notes," *CLEF*, 2011.

[50] A. Hegerath, T. Deselaers, and H. Ney, "Patch-based Object Recognition Using Discriminatively Trained Gaussian Mixtures," *BMVC*, vol. 2, pp. 519–528, 2006.

[51] J. Yang, Z. Wang, Z. Lin, X. Shu, and T. Huang, "Bilevel Sparse Coding for Coupled Feature Spaces," *CVPR*, 2012.

[52] S. Avila, N. Thome, M. Cord, E. Valle, and A. de Arajo, "Pooling in Image Representation: The Visual Codeword Point of View," *CVIU*, 2012.

[53] M. Awais, F. Yan, K. Mikolajczyk, and J. Kittler, "Novel Fusion Methods for Pattern Recognition," *ECML*, 2011.

[54] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei, "Action recognition by learning bases of action attributes and parts," *ICCV*, 2011.

[55] N. Kulkarni and B. Li, "Discriminative Affine Sparse Codes for Image Classification," *CVPR*, pp. 1609–1616, 2011.

[56] C. Zhang, Q. Huang, J. Liu, Q. Tian, C. Liang, and X. Zhu, "Image Classification Using Haar-like Transformation of Local Features with Coding Residuals," *SP*, 2012.

[57] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," *CVPR*, 2011.

[58] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007-2012 Results," http://pascallin.ecs.soton.ac.uk/challenges/VOC, 2012.

[59] L. Fei-fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *CVPR Workshop on Generative-Model Based Vision*, 2004.

[60] M. J. Huiskes, B. Thomee, and M. S. Lew, "New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative," *MIR*, pp. 527–536, 2010.

[61] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *JMLR*, 2010.

[62] P. Koniusz, "Novel Image Representations for Visual Categorisation with Bag-of-Words," Ph.D. dissertation, Centre for Vision, Speech and Signal Processing, University of Surrey, March 2013.