Comparison of Mid-Level Feature Coding Approaches And Pooling Strategies in Visual Concept Detection.

P. Koniusz, F. Yan, K. Mikolajczyk

Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH, Guildford, UK

Abstract

Bag-of-Words lies at a heart of modern object category recognition systems. After descriptors are extracted from images, they are expressed as vectors representing visual word content, referred to as mid-level features. In this paper, we review a number of techniques for generating mid-level features, including two variants of Soft Assignment, Locality-constrained Linear Coding, and Sparse Coding. We also isolate the underlying properties that affect their performance. Moreover, we investigate various pooling methods that aggregate mid-level features into vectors representing images. Average pooling, Max-pooling, and a family of likelihood inspired pooling strategies are scrutinised. We demonstrate how both coding schemes and pooling methods interact with each other. We generalise the investigated pooling methods to account for the descriptor interdependence and introduce an intuitive concept of improved pooling. We also propose a coding-related improvement to increase its speed. Lastly, state-of-the-art performance in classification is demonstrated on Caltech101, Flower17, and ImageCLEF11 datasets.

Keywords: Bag-of-Words, Mid-level features, Soft Assignment, Sparse Coding, Locality-constrained Linear Coding, Max-pooling, Analytical Pooling, Power Normalisation, Comparison

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 matching and classification. Its first implementations were associated with object retrieval and scene matching [1], as well as visual categorisation [2]. The BoW approach has undergone significant changes over recent years but it can be summarised by the following steps:

1) First, local image descriptors such as SIFT [3, 4, 5] or Gabor-based [4] vectors are extracted from images. Next, a dictionary, also known as a visual vocabulary, is learnt by finding a set of descriptive discrete appearance prototypes defined in the descriptor space, *e.g.* by k-means clustering of descriptors from a training dataset. These prototypes are often called as visual words, centres, atoms, and anchors.
2) Feature coding a.k.a. mid-level coding is then performed by embedding local descriptors into the visual vocabulary space. This results in so-called mid-level features which express each descriptor by a subset of visual words.
3) A pooling step is carried out to transform mid-level features from an image into a final representation in form of a vector called image signature. A basic pooling approach aggregates every local descriptor represented by a combination of visual words into a single signature vector. Finally, training and classification can be performed on the signatures by a classifier, *e.g.* SVM [6] or KDA [7].

Each step has a strong impact on the quality of image representation and can affect the classification performance and computational speed. The objective of this paper is to closely examine various techniques proposed for the coding and pooling steps and demonstrate their performance in a number of benchmarks.

A baseline BoW approach [1] employs k-means clustering of local descriptors from a training dataset and assigning each descriptor to the nearest cluster (mid-level coding). This is often referred to as Hard Quantisation or

Email address: p.koniusz@surrey.ac.uk (P. Koniusz)

Hard Assignment. A histogram representing the image is obtained by counting the number of assignments per cluster. Averaging such counts by the number of descriptors in the image results in Average pooling [2, 8, 9].

A number of mid-level coding methods proposed to date include Kernel Codebook [8, 9, 10, 11, 12] a.k.a. Soft Assignment and Visual Word Uncertainty, the family of Linear Coordinate Coding, entailing Sparse Coding (*e.g.* Lasso [13, 14] and greedy coders like Match Pursuit [15] and Orthogonal Match Pursuit [16]), Local Coordinate Coding [17], Locality-constrained Linear Coding [18], Laplacian Sparse Coding [19], and Over-Complete Sparse Coding [20]. Other robust approaches include Fisher Kernels [21, 22], Super Vector Coding [23], Vector of Locally Aggregated Descriptors [24], and Vector of Locally Aggregated Tensors [25].

Quantisation effects in Hard Assignment coding were found to be a source of ambiguity [10]; descriptor vectors lying on the border of two clusters can be assigned to one or the other merely due to low-level stochastic noise. It is argued in [26] that a small set of descriptors along cluster boundaries are the most discriminative ones and must be represented well, *e.g.* by hierarchical clustering. The quantisation effect is somewhat alleviated by assigning descriptors to their l-nearest clusters [10, 7] rather than to the nearest cluster only. However, descriptor vectors can be different and yet they may share the same l-nearest clusters. Soft Assignment (SA) is another approach to feature coding [8, 9] that yields cluster membership probabilities for every visual word given a descriptor. Such a strategy is beneficial as descriptors are assigned to every cluster centre with different probabilities thus improving the quantisation properties of the coding step. Lastly, there has been a significant progress in Linear Coordinate Coding (LCC) methods [13, 14, 17, 18, 19, 23] leading to state-of-the-art results with BoW [27]. These approaches seek a few weighting coefficients to linearly combine elements of the dictionary to approximate a given descriptor. Final image signatures are formed from the largest coefficients per visual word which is termed Max-pooling [14, 28, 29, 12].

Recent progress in mid-level feature coding has also provided an insight into the role played by pooling during the generation of image signatures. The theoretical relation between Average and Max-pooling was studied in [28]. A detailed likelihood-based analysis of feature pooling was conducted in [29] which led to a *theoretical expectation of Max-pooling*, improving overall classification results. Power Normalisation has been also applied to Average pooling by Fisher Kernels [22]. Lastly, Max-pooling has been recognised as a lower bound on the likelihood of *at least one particular visual word being present in an image* [12]. We show later that some of these methods are closely related.

A crucial component of the BoW approach, which has an impact on pooling, is Spatial Pyramid Matching [30]. It exploits spatial bias in images by expressing spatial relations at multiple levels of quantisation. Furthermore, clustering mid-level features and applying pooling in each cluster [31] limits the uncertainty of pooling. Exploiting other types of bias in images to partition the features is also effective, *e.g.* Dominant Angle and Colour Pyramid Matching [32].

A recent review of coding schemes [33] includes Hard Assignment, Soft Assignment, Approximate Locality-constrained Linear Coding, Super Vector Coding, and Fisher Kernels. Evaluations of BoW in [34] employ ideas from text analysis: term frequency, inverse document frequency and various normalisation schemes. The importance of mid-level coding versus dictionary training is studied in [35]. Various dictionary learning approaches are considered and described in [36]. Lastly, Hard Assignment, Soft Quantisation, and Sparse Coding are combined with Average and Max-pooling, and their characteristics are studied in depth in [28]. More pooling strategies are presented in [29].

Although there exist various comparisons of BoW, there is a lack of large scale evaluation of both mid-level coding and pooling strategies in a common testbed. The analysis of interaction between these two stages constitutes the main contribution of our work:

1) We evaluate various mid-level coding schemes such as Soft Assignment (**SA**) [8, 9, 10, 11], its extension Approximate Locality-constrained Soft Assignment (**LcSA**) [12], Sparse Coding (**SC**) [13, 14], and Approximate Locality-constrained Linear Coding [18] (**LLC**).
2) We compare various pooling schemes such as Average [2, 8, 9] (**Avg**), Max-pooling [14, 28, 29, 12] (**Max**), Power Normalisation a.k.a. Gamma Correction [22] (**Gamma**), *theoretical expectation of Max-pooling* [29] (**MaxExp**), the probability of *at least one particular visual word being present in an image* [12] (**ExaPro**), L_p-norm as a trade-off between Average and Max-pooling [29] (**lp-norm**), and Mix-order Max-pooling [12] (**MixOrd**).
3) We devise a simple approximation of MaxExp pooling (**AxMin**) and show that Gamma also approximates MaxExp. Before evaluating MaxExp, AxMin, and Gamma, we generalise them to account for the descriptor interdependence. A pooling extension is proposed that uses the top n largest mid-level feature coefficients (**@n**) per visual word. This reduces the noise and improves the performance. We show that Max-pooling is a special case of @n.

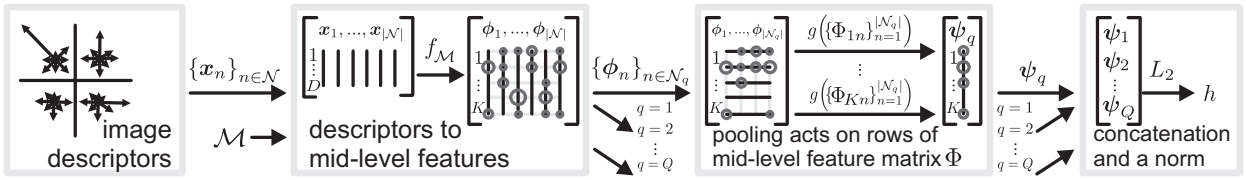


Figure 1: Overview of Bag-of-Words showing mid-level coding and pooling steps. (*a*) $|\mathcal{N}|$ local descriptors of dimension D are extracted from an image. (*b*) Mid-level coding embeds the descriptors into the visual vocabulary space using K visual words from dictionary $\mathcal{M}$. Circles of various sizes illustrate values of mid-level coefficients. (*c*) Mid-level features of partition q are stacked. Next, pooling aggregates the values along rows and forms a single vector per spatial partition. (*d*) Vectors from all partitions are concatenated and normalised to form signature h.

4) Spatial [30] and Dominant Angle Pyramid Matching [32] (**SPM** and **DoPM**) are employed to demonstrate their interaction with the pooling step. The early fusion of the spatial cues and descriptors called Spatial Coordinate Coding [32] (**SCC**) is used, as it leads to 36x faster kernel computations compared to SPM.
5) Finally, the role of the reconstruction error a.k.a. quantisation error in the coding schemes is illustrated. Furthermore, it is demonstrated empirically that minimising such an error over parameters of LcSA correlates well with its best classification performance. To increase the efficiency of coding, two coding methods are combined with Spill Trees [37] and compared to the baseline methods of various dictionary sizes.

Section 2 formally introduces Bag-of-Words and describes mid-level coding methods. Section 3 introduces pooling methods. Section 4 details the experimental framework. Various coding and pooling methods are then compared, followed by a detailed discussion. Section 5 draws conclusions on this work.

2. Overview of Mid-level Feature Coding Approaches

The goal of mid-level coding is to embed descriptors in a representative visual vocabulary space. This can be seen as a form of interpolation. Mid-level coding interpolates data on an irregular grid stretched across the surface of a hypersphere of L_2-norm normalised descriptor space. Due to the high dimensionality of the descriptor space, it is not practical to partition it evenly [38]. Thus, density estimation is usually employed to find the densely occupied regions.

Figure 1 illustrates the role of each step employed in Bag-of-Words. Formulations for mid-level coding and pooling will now be described. Let us assume descriptor vectors $\boldsymbol{x}_n \in \mathbb{R}^D$ 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. Further, $\mathcal{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ can be viewed as a descriptor set or a matrix $\mathcal{X} \in \mathbb{R}^{D \times N}$ with the descriptors as column vectors. 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 $\mathcal{N}$. Next, let us assume we have $k = 1, ..., K$ visual appearance prototypes $\boldsymbol{m}_k \in \mathbb{R}^D$ a.k.a. visual vocabulary, words, centres, atoms, and anchors. We form a dictionary $\mathcal{M} = \{\boldsymbol{m}_k\}_{k=1}^K$ such that $\mathcal{M} \in \mathbb{R}^{D \times K}$. Additionally, if applied, $q = 1, ..., Q$ denotes partitions of a chosen Pyramid Matching, *e.g.* SPM [30, 14], DoPM, or CoPM [32]. It follows $\mathcal{N}_q^i \subseteq \mathcal{N}^i$ (we write $\mathcal{N}_q$ for simplicity) is a subset of the descriptor indices that fall into a given pyramid partition q of image i. Following the formalism of [28], we express the mid-level coding and pooling steps in BoW as:

$$\boldsymbol{\phi}_n = [\Phi_{1n}, ..., \Phi_{Kn}]^T = f(\boldsymbol{x}_n, \mathcal{M}), \quad \forall n \in \mathcal{N} \tag{1}$$

$$\boldsymbol{\psi}_q = \left[\Psi_{1q}, ..., \Psi_{Kq}\right]^T, \quad \Psi_{kq} = g\left(\{\Phi_{kn}\}_{n \in \mathcal{N}_q}\right), \quad \forall q = 1, ..., Q \tag{2}$$

$$\boldsymbol{h} = \hat{\boldsymbol{h}} / \|\hat{\boldsymbol{h}}\|_2 \ , \quad \hat{\boldsymbol{h}} = \left[{\boldsymbol{\psi}_1}^T, ..., {\boldsymbol{\psi}_Q}^T\right]^T \tag{3}$$

Equation (1) represents a chosen mid-level feature mapping $f : \mathbb{R}^D \rightarrow \mathbb{R}^K$, *e.g.* Soft Assignment or Sparse Coding. It quantifies the image content in terms of the visual prototypes given in $\mathcal{M}$. Each descriptor $\boldsymbol{x}_n$ is embedded into the visual vocabulary space resulting in mid-level features $\boldsymbol{\phi}_n \in \mathbb{R}^K$. We also define a set/matrix $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_n\}_{n \in \mathcal{N}}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{K \times |\mathcal{N}|}$. It follows Φ_{kn} are element-wise entries of $\boldsymbol{\Phi}$. Intuitively, figure 1 (a) illustrates descriptors $\{\boldsymbol{x}_n\}_{n \in \mathcal{N}}$ of image i, later used by the coding step in figure 1 (b). Next, coding operates on each descriptor and produces corresponding mid-level features $\{\boldsymbol{\phi}_n\}_{n \in \mathcal{N}}$. Note that $\mathcal{M}$ is formed from k-means cluster centres, later used by all mid-level coding approaches. Hence, equation (1) does not include the dictionary learning step.

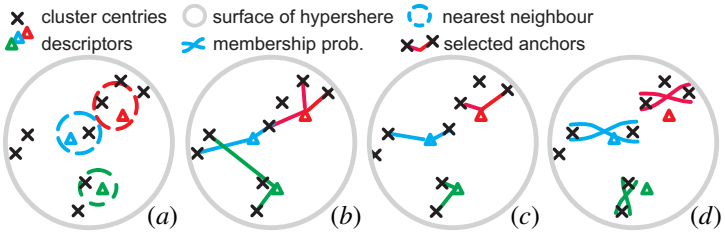


Figure 2: Illustration of (*a*) Hard Assignment, (*b*) Sparse Coding, (*c*) Locality-constrained Linear Coding, (*d*) Approximate Locality-constrained Soft Assignment. Descriptor vectors (triangles) are scattered on a surface of a hypersphere amongst the anchors (crosses). Note the difference between SC and LLC.

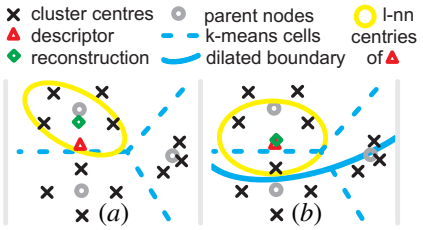


Figure 3: (*a*) Hierarchical NN: *l*-nearest anchors of a descriptor found in its nearest k-means cluster. (*b*) Dilating cluster boundaries improves quantisation: a descriptor and its reconstruction are brought closer.

Equation (2) represents the pooling operation, *e.g.* Average or Max-pooling. The role of g is to aggregate occurrences of visual words 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 partition q of image i to produce a k^{th} coefficient in vector $\boldsymbol{\psi}_q \in \mathbb{R}^K$. Set/matrix $\boldsymbol{\Psi}$ is defined as $\boldsymbol{\Psi} = \left\{\boldsymbol{\psi}_q\right\}_{q=1}^{Q}$ with Ψ_{kq} being element-wise entries of $\boldsymbol{\Psi}$. Figure 1 (c) depicts mid-level feature coefficients $\{\Phi_{kn}\}_{n \in \mathcal{N}_q}$ which are used by the pooling step given $k = 1, ..., K$. Note that g acts on a given k^{th} row of mid-level features by aggregating occurrences of $\boldsymbol{m}_k$ into a k^{th} coefficient in $\boldsymbol{\psi}_q$.

Equation (3) concatenates $\boldsymbol{\psi}_q$ for all partitions $q = 1, ..., Q$ into $\hat{\boldsymbol{h}} \in \mathbb{R}^{KQ}$. It also normalises signature $\hat{\boldsymbol{h}}$ to preserve only relative statistics of visual word occurrences in an image, irrespective of the number of descriptors contained within it. This yields the final signature $\boldsymbol{h} \in \mathbb{R}^{KQ}$ of unit length as illustrated in figure 1 (d). The resulting signatures $\boldsymbol{h}_i, \boldsymbol{h}_j \in \mathbb{R}^{KQ}$ for $i, j \in \mathcal{I}$ can be directly fed to a primary-formulated SVM classifier or used to form a linear kernel $ker_{ij} = (\boldsymbol{h}_i)^T \cdot \boldsymbol{h}_j$. This defines the similarity between images for kernel based classifiers. The latter is used in this work, with a dual-form KDA classifier [7].

The HA, SA, SC, LLC, and LcSA coding methods will now be described using the terms introduced above. For simplicity, $\boldsymbol{x}_n$ is referred to as $\boldsymbol{x}$, $\boldsymbol{\phi}_n$ as $\boldsymbol{\phi}$, and $\boldsymbol{\psi}_q$ as $\boldsymbol{\psi}$ where possible. Thus, $[\phi_1, ..., \phi_K]^T = \boldsymbol{\phi}$ and $[\psi_1, ..., \psi_K]^T = \boldsymbol{\psi}$. Further, we define the activation of anchor $\boldsymbol{m}_k$ given $\boldsymbol{x}$ as a response $\phi_k \neq 0$ and the local activation as $\phi_k \neq 0$ such that $r^2 = \|\boldsymbol{m}_k - \boldsymbol{x}\|_2^2$ and $r^2 < \kappa$ for an arbitrarily chosen constant $\kappa > 0$, where k defines a neighbourhood such that any two descriptors chosen from it have close visual appearances. Intuitively, $\phi_k \neq 0$ and $r^2 \geq \kappa$ define a non-local activation.

*2.1. Hard Quantisation a.k.a. Hard Assignment (**HA**)*

Bag-of-Words in its simplest form employs HA that solves the following optimisation problem:

$$\begin{aligned} \boldsymbol{\phi} &= \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \mathcal{M}\bar{\boldsymbol{\phi}} \right\|_2^2 \\ s.\,t.\ \ &\|\bar{\boldsymbol{\phi}}\|_1 = 1, \bar{\boldsymbol{\phi}} \in \{0, 1\}^K \end{aligned} \tag{4}$$

In practice, equation (4) means that having formed a dictionary $\mathcal{M}$ by k-means clustering (or any other method), every descriptor $\boldsymbol{x} \in \mathcal{X}$ is assigned to its nearest cluster with activation equal 1. This is illustrated in figure 2 (a). The L_1-norm constraint $\|\boldsymbol{\phi}\|_1 = 1$ ensures that $\boldsymbol{\phi}$ are histograms. Since $\boldsymbol{\phi}$ can take only binary values, the L_1-norm also ensures a single non-zero entry per $\boldsymbol{\phi}$. Recently, it was shown that HA with appropriate pooling can achieve improved results [29, 33] despite its inherently high quantisation error. However, methods like Sparse Coding were shown to consistently perform significantly better. Therefore, we omit HA in the following evaluations.

*2.2. Soft Assignment (**SA**)*

Consider a Mixture of K Gaussian functions [39] with the following parameters to estimate $\theta = (\theta_1, ..., \theta_K) = ((p_1, \boldsymbol{m}_1, \boldsymbol{\sigma}_1), ..., (p_K, \boldsymbol{m}_K, \boldsymbol{\sigma}_K))$. K denotes the number of Gaussian components G, p_k are the component mixing probabilities $k = 1, ..., K$, $\boldsymbol{m}_k$ are the Gaussian means, $\boldsymbol{\sigma}_k$ are the component standard deviations, and $\mathcal{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ are descriptors of a dataset. The density estimation problem can be addressed by optimising $\Lambda(\mathcal{X}; \theta)$:

$$\Lambda(\mathcal{X}; \theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k G(\boldsymbol{x}_n; \boldsymbol{m}_k, \boldsymbol{\sigma}_k) \tag{5}$$

The parameters of the model in equation (5) have a vast number of degrees of freedom and therefore are further reduced to $\theta = (\theta_1, ..., \theta_K) = ((\boldsymbol{m}_1, \sigma), ..., (\boldsymbol{m}_K, \sigma))$ by fixing all mixing probabilities $p_1 = p_2 = ... = p_K \neq 0$ to be equal and having a single σ parameter such that $\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_2 = ... = \boldsymbol{\sigma}_K = \sigma \neq 0$. This leads to the expression for the membership probability of component $\boldsymbol{m}_k$ being selected given descriptor $\boldsymbol{x}$:

$$\phi_k = p(\boldsymbol{m}_k|\boldsymbol{x}, \sigma) = \frac{G(\boldsymbol{x}; \boldsymbol{m}_k, \sigma)}{\sum_{k'=1}^{K} G(\boldsymbol{x}; \boldsymbol{m}_{k'}, \sigma)} \tag{6}$$

Defining $\psi_k = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn}$, where $\Phi_{kn} = p(\boldsymbol{m}_k|\boldsymbol{x}_n, \sigma)$, turns such a formulation into Soft Assignment [9]. Hence, $\{\boldsymbol{m}_k\}_{k=1}^{K}$ denotes the visual codewords formed with k-means and σ is the smoothing parameter of kernel G [9].

*2.3. Sparse Coding (**SC**)*

The goal of Sparse Coding [13, 14] is to express each descriptor vector $\boldsymbol{x}$ as a sparse linear combination of the visual words given by $\mathcal{M}$. This can be achieved by optimising the following with respect to $\boldsymbol{\phi}$:

$$\begin{aligned} &\boldsymbol{\phi} = \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \mathcal{M}\bar{\boldsymbol{\phi}} \right\|_2^2 + \alpha \|\bar{\boldsymbol{\phi}}\|_1 \\ &s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0 \end{aligned} \tag{7}$$

The L_1-norm over $\boldsymbol{\phi}$ induces a low number of activations per descriptor, referred to as sparsity, which can be adjusted with α. SC was found to perform well if combined with Max-pooling and Spatial Pyramid Matching [14]. Defining $\psi_k = \max(\{\Phi_{kn}\}_{n \in \mathcal{N}})$ in equation (2) renders this model equivalent to Sparse Coding [14] except for: i) a skipped dictionary learning step, ii) a non-negative constraint [1] on $\boldsymbol{\phi}$. The image signatures in [14] are twice as long due to pooling over positive and negative Φ_{kn} respectively. It is shown later that neglecting negative activations has no detrimental impact on the classification performance. Figure 2 (b) shows that SC can activate non-local anchors.

*2.4. Approximate Locality-constrained Linear Coding (**LLC**)*

Locality-constrained Linear Coding [18] addresses the non-locality that can occur in Sparse Coding. It prevents activations of visual words that are far from descriptors. See figures 2 (b) and (c) for intuitive differences. The problem is formulated as:

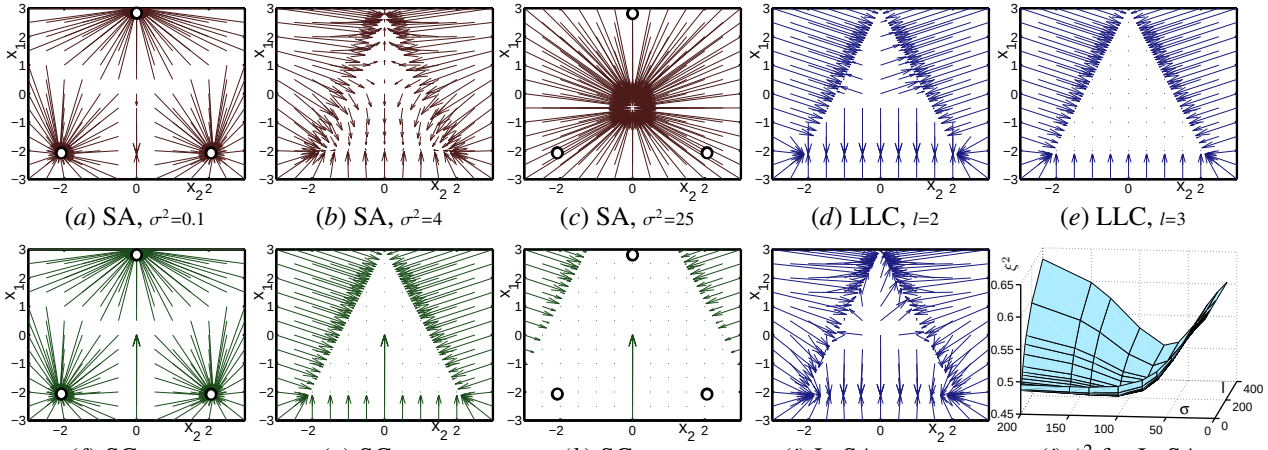
$$\begin{aligned} &\boldsymbol{\phi} = \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \mathcal{M}\bar{\boldsymbol{\phi}} \right\|_2^2 + \alpha \sum_{k=1}^{K} \left(\bar{\phi}_k \cdot e^{\frac{\|\boldsymbol{x} - \boldsymbol{m}_k\|_2}{\sigma}} \right)^2 \\ &s.\,t.\ \ \mathbf{1}^T \bar{\boldsymbol{\phi}} = 1 \end{aligned} \tag{8}$$

The squared L_2-norm, expressed as a summation on the right side of equation (8), penalises large ϕ_k if the corresponding $\boldsymbol{m}_k$ is far from a given descriptor $\boldsymbol{x}$. The penalty can be adjusted by α and σ. This problem is equivalent to the problem in [18], except for the dictionary learning step. In practice, we solve an alternative fast approximate formulation:

$$\begin{aligned} &\boldsymbol{\phi}^* = \arg\min_{\bar{\boldsymbol{\phi}}} \left\| \boldsymbol{x} - \mathcal{M}(\boldsymbol{x}, l)\, \bar{\boldsymbol{\phi}} \right\|_2^2 \\ &s.\,t.\ \ \bar{\boldsymbol{\phi}} \geq 0,\ \ \mathbf{1}^T \bar{\boldsymbol{\phi}} = 1 \end{aligned} \tag{9}$$

Descriptor $\boldsymbol{x}$ is coded with its l-nearest neighbour anchors found in dictionary $\mathcal{M}$ by NN search, a new compact dictionary is formed and used: $\mathcal{M}(\boldsymbol{x}, l) = NN_{\mathcal{M}}(\boldsymbol{x}, l) \in \mathbb{R}^{D \times l}$, where $l \ll K$. Hence, one has to adjust l instead of α and σ. Note, the resulting $\boldsymbol{\phi}^* \in \mathbb{R}^l$ has length l. In practice, we re-project its elements into the full length vector $\boldsymbol{\phi} \in \mathbb{R}^K$ as, for each atom in $\mathcal{M}(\boldsymbol{x}, l)$, we know its position in $\mathcal{M}$. A non-negativity constraint[1] is applied to $\boldsymbol{\phi}$ as no classification improvement is observed if $\boldsymbol{\phi} < 0$ is allowed. Figure 2 (c) depicts a local selection of anchors for LLC.

[1] To impose $\boldsymbol{\phi} \geq 0$ on SC and LLC, we used LAR [40] solver implemented in SPAMS [41] and Quadratic Programming [42], respectively. However, ignoring constraint $\boldsymbol{\phi} \geq 0$ and correcting SC and LLC codes by $\phi_k := max(0, \phi_k)$ for $k = 1, ..., K$ yielded equally good results.



(*a*) SA, σ^2=0.1 (*b*) SA, σ^2=4 (*c*) SA, σ^2=25 (*d*) LLC, l=2 (*e*) LLC, l=3

(*f*) SC, α=100 (*g*) SC, α=1 (*h*) SC, α=0.5 (*i*) LcSA, σ^2=4, l=2 (*j*) ξ^2 for LcSA

Figure 4: Simulation of the quantisation error: flow of the descriptors from their original positions $\boldsymbol{x}$ denoted by the grid points to the corresponding reconstructed positions pointed to by the arrows. (*a*) SA: the descriptors are moved to their nearest anchors 'o' like in HA. (*b*) SA: a near-optimal smoothing factor case yielding low ξ^2. (*c*) SA: a full blur of the data for large σ. The reconstructed positions overlap in the centre. (*d*) LLC: limited reconstruction due to low $l = 2$. (*e*) LLC: optimal reconstruction within the triangular region given $l = 3$. (*f*) SC: the descriptors are moved to their nearest anchors 'o' like in HA. Note, $\|\boldsymbol{\phi}\|_1 = 1/\alpha$ had to be rescaled to $\|\boldsymbol{\phi}\|_1 = 1$ to prepare this plot. (*g*) SC: optimal reconstruction within the triangular region. (*h*) SC: area of the optimal reconstruction is increased for small α at a price of non-sparsity. (*i*) LcSA: reconstruction capabilities of LcSA resemble closely LLC case (d). (*j*) LcSA: cost ξ^2 resulting from combining equations (10) and (11), shown as a function of (σ, l).

*2.5. Approximate Locality-constrained Soft Assignment (**LcSA**)*

Sparse Coding [13, 14] and Locality-constrained Linear Coding [18] are robust approaches that can learn a data manifold by approximating it with sparse and local linear combinations of anchors, respectively. This is achieved by constraining activations to a relevant subset of anchors. Thus, we constrain Soft Assignment to activate only the l-nearest anchors of the descriptors as in [18, 12] when computing the membership probabilities. This is illustrated in figure 2 (d). This method is referred to as Approximate Locality-constrained Soft Assignment. Recall that $\mathcal{M}(\boldsymbol{x}, l) = NN_{\mathcal{M}}(\boldsymbol{x}, l) \in \mathbb{R}^{D\times l}$ is a set of the l-nearest anchors of descriptor $\boldsymbol{x}$ given dictionary $\mathcal{M}$ such that $l \ll K$. Limiting the membership probability in equation (6) to be spanned with only l-local anchors $\mathcal{M}(\boldsymbol{x}, l)$ yields:

$$\phi_k = p(\boldsymbol{m}_k|\boldsymbol{x}, \sigma, l) = \begin{cases} \frac{G(\boldsymbol{x};\boldsymbol{m}_k,\sigma)}{\sum_{\boldsymbol{m}'\in\mathcal{M}(\boldsymbol{x},l)} G(\boldsymbol{x};\boldsymbol{m}',\sigma)} & \text{if } \boldsymbol{m}_k \in \mathcal{M}(\boldsymbol{x}, l) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

2.6. Mid-level Coding Parameters

To achieve good performance, SC and LLC optimise a trade-off between a quantisation loss (defined below) and an explicitly chosen regularisation penalty, *e.g.* sparsity as in equation (7) or locality as in equation (8). Such a trade-off can be subjected to additional constraints, *e.g.* non-negativity and an upper limit on the solution. The quality of quantisation in these mappings is measured in accordance with the theory of Linear Coordinate Coding [17]. Coordinate Coding is a pair $(f, \mathcal{M})$, where $\mathcal{M} \in \mathbb{R}^{D\times K}$ is a visual dictionary and f is a mapping a.k.a. coder of a descriptor $\boldsymbol{x} \in \mathbb{R}^D$ to a mid-level feature $[f_{\boldsymbol{m}}(\boldsymbol{x})]_{\boldsymbol{m}\in\mathcal{M}} \in \mathbb{R}^K$ as in section 2. One further constraint that may be imposed is $\sum_{\boldsymbol{m}} f_{\boldsymbol{m}}(\boldsymbol{x}) = 1$ and $f_{\boldsymbol{m}}(\boldsymbol{x}) \geq 0$ if histograms are required. The linear approximation of $\boldsymbol{x}$ can be expressed as: $\hat{\boldsymbol{x}} = \sum_{\boldsymbol{m}\in\mathcal{M}} f_{\boldsymbol{m}}(\boldsymbol{x})\boldsymbol{m}$. Thus, the residual error of approximation of a descriptor vector $\boldsymbol{x}$ is:

$$\xi^2(\boldsymbol{x}) = \left\| \boldsymbol{x} - \sum_{\boldsymbol{m}\in\mathcal{M}} f_{\boldsymbol{m}}(\boldsymbol{x}_n) \cdot \boldsymbol{m} \right\|_2^2 \tag{11}$$

Equation (11) shows that transforming descriptor $\boldsymbol{x}$ into mid-level feature $\boldsymbol{\phi} = f(\boldsymbol{x})$ results in a quantisation loss $\xi^2(\boldsymbol{x})$ a.k.a. the residual error which depends on the choice of mapping f. Transforming the mid-level feature back into the descriptor yields $\xi^2(\boldsymbol{x})$. The approximation error of N descriptors is $\xi^2 = \frac{1}{N}\sum_n \xi^2(\boldsymbol{x}_n)$. We assume ξ^2 is synonymous with the quantisation error, which is a source of ambiguity in coding, *e.g.* Hard Assignment. Moreover, regularisation terms must be imposed to ensure that each descriptor is coded by a representative fraction of atoms. For instance, we observed that given the optimal regularisation, mid-level features from various classes of textures exhibit high

intra-class and low inter-class similarity. However, removing regularisation leads to a sharp increase of inter-class similarity. Such mid-level features are not distinctive enough for a pooling step to produce informative signatures.

Figure 4 presents how mid-level features are affected by the quantisation error. Having coded descriptors $\boldsymbol{x} = [x_1, x_2]^T \in \langle -3; 3\rangle^2$ with $k = 1, 2, 3$ atoms $\boldsymbol{m}_k$ by various methods, the obtained codes $\boldsymbol{\phi}$ are projected back to the descriptor space: $\hat{\boldsymbol{x}} = \mathcal{M}\boldsymbol{\phi}$. The resulting quantisation effects are visualised as displacements between each descriptor $\boldsymbol{x}$ and its approximation $\hat{\boldsymbol{x}}$. Plots (a-c) present SA with low σ (HA equivalent), optimal, and large σ (data blur: if $\sigma \to +\infty$, then $\phi_k \to 1/K$). Plot (d) shows LLC, which modifies the descriptor space for $l = 2$. Plot (e) shows LLC yielding a good reconstruction for $l = 3$, however, this causes non-locality. Plots (f-h) show SC with high α (HA equivalent, $\|\boldsymbol{\phi}\|_1 = 1/\alpha$ was rescaled to $\|\boldsymbol{\phi}\|_1 = 1$), medium α (good trade-off), and low α at a price of non-sparsity. Plot (i) shows LcSA approximating LLC in plot (d). Lastly, plot 4 (j) shows the ξ^2 cost for LcSA coder f in equation (10) as a function of (σ, l) yielded by equation (11). Note, $\xi^2 > 0$ has a unique minimum and it varies smoothly with changes of (σ, l). Many variants of descriptors and datasets were consistently found to have a unique minimum.

Typically, the optimal coding parameters are determined during the cross-validation process. We found empirically that minimising $\xi^2 > 0$ w.r.t. (σ, l) in the LcSA model led to good classification results. This can be explained by two trade-off factors: i) Extreme σ results in either HA or the data blur as shown in plots 4 (a-c). Thus, measuring ξ^2 can be used to penalise selection of such extremes. ii) Usually, given the L_2-norm normalised data, descriptor $\boldsymbol{x}$ coded with the distant anchors yields approximation $\hat{\boldsymbol{x}}_1$ such that $\|\hat{\boldsymbol{x}}_1\|_2 < \|\boldsymbol{x}\|_2$ due to various implicit constraints of LcSA, *e.g.* $\boldsymbol{\phi} \geq 0$, $\|\boldsymbol{\phi}\|_1 = 1$. However, coding $\boldsymbol{x}$ with both distant and nearby anchors yields $\hat{\boldsymbol{x}}_2$ such that $\|\hat{\boldsymbol{x}}_1\|_2 < \|\hat{\boldsymbol{x}}_2\|_2 < \|\boldsymbol{x}\|_2$. Lastly, coding $\boldsymbol{x}$ with its nearby anchors only yields $\hat{\boldsymbol{x}}_3$ such that $\|\hat{\boldsymbol{x}}_1\|_2 < \|\hat{\boldsymbol{x}}_2\|_2 < \|\hat{\boldsymbol{x}}_3\|_2 < \|\boldsymbol{x}\|_2$. This suggests ξ^2 shown in plot 4 (j) favours local coding in LcSA. Thus, we combine equations (10) and (11) to find the initial σ and l-nearest anchors:

$$(\sigma,\ l) = \arg\min_{(\bar{\sigma},\ \bar{l})} \sum_{n=1}^{N} \left\| \boldsymbol{x}_n - \sum_{\boldsymbol{m} \in \mathcal{M}(\boldsymbol{x}_n, \bar{l})} \frac{G(\boldsymbol{x}_n; \boldsymbol{m}, \bar{\sigma})}{\sum_{\boldsymbol{m}' \in \mathcal{M}(\boldsymbol{x}_n, \bar{l},)} G(\boldsymbol{x}_n; \boldsymbol{m}', \bar{\sigma})} \cdot \boldsymbol{m} \right\|_2^2 \tag{12}$$

Such evaluated parameters were found to provide good initial estimates. Next, (σ, l) can be adjusted by cross-validation for optimal classification performance. Similar heuristics demonstrated good empirical results for SA [11].

2.7. Computational efficiency

When embedding descriptors (*e.g.* 6K per image) of a medium scale dataset to a vocabulary space (*e.g.* 16K atoms), the computational cost of coding becomes a major factor in experiments. Thus, this section details the computational complexity of HA, SA, LcSA, SC, and LLC and proposes an approach which increases the speed of coding.

HA. Hard Assignment requires a nearest neighbour search which scales linearly with the number of descriptors N and the number of visual words K. This results in a complexity $O(N \times K)$.

SA. Soft Assignment computes: i) Gaussian-based distances from a descriptor to each visual word, ii) the sum of such distances, iii) the ratio of (i) to the total distance (ii) as in equation (6). Therefore, $O(N \times 3K) = O(N \times K)$.

SC. The complexity of Sparse Coding based on the Feature Sign [13] solver is expressed as $O(N \times K \times S)$, where S is the average number of non-zero elements in the mid-level features. The complexity of the Least Angle Regression [40] based solver proposed in [41] is $O(N \times S^3 \ + \ N \times K \times S^2 \ + \ N \times K \times S) = O(N \times K \times S^2)$ for $S \ll K$.

LLC. Because Locality-constrained Linear Coding is $O(N \times K^2)$ complex, Approximate LLC was also introduced in [18]. It has a complexity $O(N \times K \times log\ l \ + \ N \times l^2) = O(N \times K \times log\ l)$ for $l \ll K$ nearest anchors.

LcSA. The speed of Approximate Locality-constrained Soft Assignment is restricted by the nearest-neighbour search based on the partial sort algorithm with typical complexity $O(N \times K \times log\ l)$, where l is the number of nearest anchors in the search. Summing distances and computing the ratio of Gaussians in equation (10) becomes an efficient task with complexity $O(N \times 2l)$. Therefore, the total complexity is $O(N \times K \times log\ l \ + \ N \times 2l) = O(N \times K \times log\ l)$. Note that LcSA becomes noticeably faster than SA for $log\ l \ll 3$ since $N \times K \times log\ l \ll N \times 3K$.

FHNNS. To increase coding speed, we propose a Fast Hierarchical Nearest Neighbour Search that uses an approximate dictionary search for the l-nearest neighbours of a to-be-coded descriptor $\boldsymbol{x}$ to form a compact dictionary $\mathcal{M}(\boldsymbol{x}, l)$. Figure 3 (a) shows a hierarchical k-means vocabulary with two levels of depth. The parent node which is closest to $\boldsymbol{x}$ is found and then the l-nearest children. However, such a process results in a high quantisation jitter and a poor selection of anchors. Thus, we propose to share k-means children nodes located along boundaries between their parent nodes. The dilation of k-means boundaries is shown in figure 3 (b). A similar approach to NN search is used by Spill Trees [37]. To measure the corresponding quantisation noise the formula (11) is used over a set of descriptors.

In detail, for every k-means parent node $\boldsymbol{m'} \in \mathcal{M}'$ its dilated set of children $\hat{\mathcal{M}}(\boldsymbol{m'}, \ell)$ is defined as $\hat{\mathcal{M}}(\boldsymbol{m'}, \ell) = NN_{\mathcal{M}}(\boldsymbol{m'}, \ell)$: the ℓ-nearest neighbours of each $\boldsymbol{m'}$ are chosen from the dictionary $\mathcal{M}$ representing the original child nodes of k-means. To increase the speed of LcSA and LLC, we combine two search operations such that $\boldsymbol{m'} = NN_{\mathcal{M}'}(\boldsymbol{x}, 1)$ indicates the nearest parent node $\boldsymbol{m'}$ of $\boldsymbol{x}$ and $\mathcal{M}(\boldsymbol{x}, l) = NN_{\hat{\mathcal{M}}(\boldsymbol{m'}, \ell)}(\boldsymbol{x}, l)$ forms a compact dictionary for $\boldsymbol{x}$. For SC, we take the nearest parent node $\boldsymbol{m'}$ of $\boldsymbol{x}$ and code $\boldsymbol{x}$ using the dilated dictionary $\hat{\mathcal{M}}(\boldsymbol{m'}, \ell)$. Varying $\ell = 1, ..., K$ affects a trade-off between speed and accuracy. In all cases, mid-level features remain of length K, rather than ℓ, as we re-project them for each atom in $\hat{\mathcal{M}}(\boldsymbol{m'}, \ell)$ to its corresponding position in $\mathcal{M}$. The complexity of LcSA and LLC becomes $O(N \times \ell_p + N \times \ell \times log\ l) = O(N \times \ell \times log\ l)$, for $\ell_p \ll \ell \ll K$ and $l \ll \ell$, where ℓ_p and ℓ are a number of parent nodes and children per node, respectively. The complexity of SC is thus $O(N \times \ell \times S^2)$.

Timing. Table 1 shows the computation times on a single 2.3GHz AMD Opteron core that are required to code 1K SIFT descriptors of 128 and 192 dimensions to mid-level features for 4K and 16K dictionaries, respectively. LcSA can run 4 times faster without a loss in its classification performance, as shown in section 4.3. SC also gains on speed.

3. Overview of Pooling Approaches

Pooling converts mid-level features into final image signatures by aggregating occurrences of visual words in each image. Formally, equation (2) expresses its place in the context of Bag-of-Words. Pooling is performed in each pyramid partition q of image i, $\mathcal{N}_q^i$ denotes a subset of descriptor indices to be processed. We abbreviate $\mathcal{N}_q^i$ to $\mathcal{N}$ and $\boldsymbol{\psi}_q$ to $\boldsymbol{\psi}$ for clarity, thus $[\psi_1, ..., \psi_K]^T = \boldsymbol{\psi}$.

*3.1. Average (**Avg**), Maximum Pooling (**Max**), Mix-order Max-pooling (**MixOrd**), and an L_p-norm based trade-off (**lp-norm**)*

Average and Max-pooling are intuitively introduced in section 1 and referred to in sections 2.2 and 2.3. To summarise, Average pooling is expressed as the average over responses to visual word $\boldsymbol{m}_k$:

$$\psi_k = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn} \tag{13}$$

Maximum pooling intuitively selects the largest value between mid-level features responding to visual word $\boldsymbol{m}_k$:

$$\psi_k = \max\left(\{\Phi_{kn}\}_{n \in \mathcal{N}}\right) \tag{14}$$

Therefore, the fundamental difference is that Average pooling counts all occurrences of visual word $\boldsymbol{m}_k$ in the image while Max-pooling only registers a presence of $\boldsymbol{m}_k$. Max-pooling has been shown to be a lower bound of the likelihood of *at least one visual word $\boldsymbol{m}_k$ being present in image i* [12]. This however does not clarify whether the lower bound formulation is more suited for classification than the exact analytical solution.

Further, Mix-order Max-pooling is proposed in [12] as a lower bound of *at least s visual words $\boldsymbol{m}_k$ being present in image i*. This is achieved by sorting all mid-level feature entries corresponding to a visual word $\boldsymbol{m}_k$ and selecting exactly the s^{th} largest value. This process is performed for $k = 1, ..., K$ and it results in an image signature. Furthermore, selecting t different values of s (*e.g.* $s_1 > s_2 > ... > s_t$) yields t different image signatures per image. They form separate kernels that can be combined using kernel methods [12].

Lastly, a trade-off between Average and Max-pooling was proposed in [29]. It employs an L_p-norm with parameter p which varies the solution between Average and Max-pooling for $p = 1$ and $p \to \infty$, respectively:

$$\psi_k = \left(\frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} |\Phi_{kn}|^p \right)^{1/p} \tag{15}$$

	SA	LcSA	LLC	SC
4K/128D	2.26	0.24	0.44	3.61
	LcSA ℓ=256	LcSA ℓ=512	LcSA ℓ=1024	LcSA ℓ=2048
	0.036	0.046	0.074	0.136

	SA	LcSA	LLC	SC
16K/192D	13.8	1.06	1.55	32.5
	SC ℓ=1024	SC ℓ=2048	SC ℓ=3072	SC ℓ=4096
	3.69	8.74	14.7	21.8

Table 1: Computational times (in seconds) required to code 1K SIFT descriptors to mid-level features.

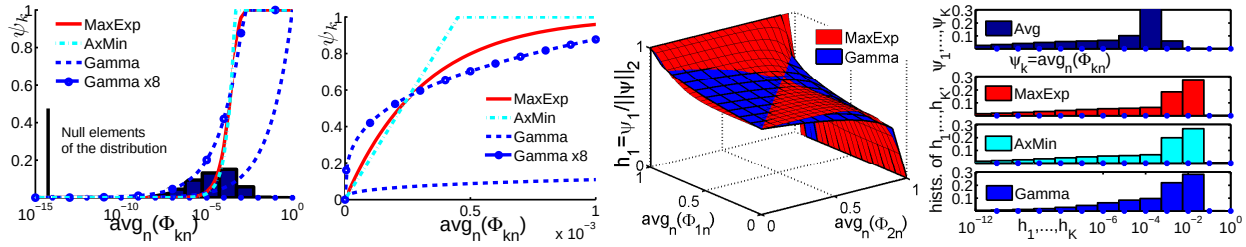


Figure 5: Illustration of pooling correction functions: MaxExp, AxMin, and Gamma. (a) Bar plot is a histogram of Average pooling $avg_n(\Phi_{kn})$ over $n = 1, ..., N$ for $k = 1, ..., K$ on Caltech101. AxMin and Gamma (if magnified x8) curves are approximations of MaxExp. Note the logarithmic scale. (*b*) Pooling methods as functions of Average pooling (linear scale). (*c*) L_2-norm normalised MaxExp and Gamma as functions of Average pooling on a dictionary $K = 2$ atoms (response h_1 for $\boldsymbol{m}_1$ is showed while we skip h_2 for clarity). (*d*) Histogram of Average pooling for $k = 1, ..., K$ on Flower17 is rearranged by MaxExp, AxMin, and Gamma, then L_2-norm normalised. This results in similar distributions (null entries not shown).

*3.2. Theoretical expectation of Max-pooling (**MaxExp**) and at least one visual word $\boldsymbol{m}_k$ present in image i (**ExaPro**)*

Likelihood based pooling methods have recently shed new light on the role of the pooling step in Bag-of-Words. It was shown in [29] that Max-pooling can be predicted analytically by drawing mid-level features (for a chosen $\boldsymbol{m}_k$) from Bernoulli distribution under the i.i.d. assumption. We assume the probability p for an event $(\Phi_{kn} = 1)$ and $1 - p$ for $(\Phi_{kn} = 0)$. Probability of all $\bar{N} = |\mathcal{N}|$ mid-level features to be $\{(\Phi_{k1} = 0), ..., (\Phi_{k\bar{N}} = 0)\}$ amounts to $(1 - p)^{\bar{N}}$. Similarly, the probability of at least one mid-level feature event $(\Phi_{kn} = 1)$ can be thought of as applying a logical 'or' operation $\left\{(\Phi_{k1} = 1) \,\middle|\, ... \,\middle|\, (\Phi_{k\bar{N}} = 1)\right\}$ and is defined as:

$$\sum_{n=1}^{\bar{N}} \binom{\bar{N}}{n} p^n (1-p)^{\bar{N}-n} = 1 - (1-p)^{\bar{N}} \tag{16}$$

Estimating p as the average of mid-level feature activations for a given $\boldsymbol{m}_k$ results in the final **MaxExp** formulation:

$$\psi_k = 1 - \left(1 - \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn}\right)^{\bar{N}}, \quad \bar{N} = |\mathcal{N}| \tag{17}$$

Next, similar assumptions to **MaxExp** were taken in [12]: mid-level features represent random variables drawn from a feature distribution under the i.i.d. assumption. Therefore, the probability of *at least one visual word $\boldsymbol{m}_k$ present in image i* (**ExaPro**) is defined as:

$$\psi_k = 1 - \prod_{n \in \mathcal{N}} (1 - \Phi_{kn}) \tag{18}$$

Note that the probabilistic interpretation of ExaPro also holds for MaxExp due to the way it acts on Average pooling. The next section shows that Power Normalisation used for Fisher Kernels [22] acts similarly on Average pooling.

*3.3. Power Normalisation a.k.a. Gamma Correction (**Gamma**)*

Power Normalisation has been successfully applied to Intersection Kernels [43], Fisher Kernels [22], and in image retrieval [44]. This is also known as Gamma Correction. Such a correction is shown to tackle *burstiness*: a phenomenon that a given visual word appears in an image more often than is statistically expected [44]. Gamma acts on Average pooling to improve the similarity of the image signatures belonging to each class of objects and it is expressed as:

$$\psi_k = \left(\frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn}\right)^{\gamma} \tag{19}$$

The correction factor $0 < \gamma \leq 1$ is usually found by cross-validation. Note, setting $\gamma = 0.5$ changes a dot product between such formed vectors $\boldsymbol{\psi}$ into Bhattacharyya coefficient [45]. As the nature of Gamma is not explored in previous studies [43, 22, 44], our study found it closely related to MaxExp. According to equations (17) and (19), these two corrections are functions of Average pooling. Thus, the best performing correction curves were plotted on Caltech101 in figure 5 (a, b). Both MaxExp and Gamma x8 (magnified x8) have a similar appearance. They rapidly

expand input intervals $\langle 0; 0.0005\rangle$ and $\langle 0.0005; 0.001\rangle$ having equal lengths to output intervals $\langle 0; 0.8\rangle$ and $\langle 0.8; 0.98\rangle$ of two different lengths 0.8 and 0.18. Hence, the importance of low averages of activations increases when compared to the strong cases. The similarity of MaxExp and Gamma (not to be confused with Gamma x8) becomes clear in figure 5 (c) due to L_2-norm normalisation as in equation (3). Averages of 2D mid-level features are taken to be the inputs for MaxExp and Gamma. Only γ is adjusted for the best fit between two curves. Resulting L_2-norm normalised histogram bins $h_1 = \psi_1/\|\boldsymbol{\psi}\|_2$ are shown. With L_2-norm handling the scaling, MaxExp and Gamma prove to be similar.

To validate whether Gamma and MaxExp act similarly in practice, a registration experiment was conducted. Assume $\hat{\boldsymbol{h}}_i^{\boldsymbol{exp}}$ are known image signatures generated with MaxExp pooling for its known optimal $\bar{N}$, while $\hat{\boldsymbol{h}}_i^{\gamma}$ are corresponding signatures generated with Gamma pooling for various candidates $\bar{\gamma}$. An unknown parameter γ of $\hat{\boldsymbol{h}}_i^{\gamma}$ is sought that minimises the least squares error between image signatures of MaxExp and Gamma for images $i \in \mathcal{I}$:

$$\gamma = \arg\min_{\bar{\gamma}} \sum_{i \in \mathcal{I}} \left\| \frac{\hat{\boldsymbol{h}}_i^{\boldsymbol{exp}}}{\|\hat{\boldsymbol{h}}_i^{\boldsymbol{exp}}\|_2} - \frac{\hat{\boldsymbol{h}}_i^{\bar{\gamma}}}{\|\hat{\boldsymbol{h}}_i^{\bar{\gamma}}\|_2} \right\|_2^2 \tag{20}$$

Indeed, section 4.2 later shows that the best performing γ determined by cross-validation matches closely γ found by optimising the target in equation (20).

3.4. Modelling the Impact of Descriptor Interdependency on Analytical Pooling

The standard approach to Bag-of-Words typically assumes the descriptor extraction on a dense grid [8, 9, 10, 11, 12, 14, 18, 19]. Thus, neighbouring descriptors largely overlap with each other. MaxExp and ExaPro pooling assume that activations ϕ_k of anchor $\boldsymbol{m}_k$ are independent in each image. However, if descriptor $\boldsymbol{x}$ results in activation ϕ_k of $\boldsymbol{m}_k$, descriptors significantly overlapping with $\boldsymbol{x}$ should also result in activations ϕ_k of $\boldsymbol{m}_k$. The same holds for repeatable visual patterns. Thus, we expect the average activation p (Average pooling) in equation (16) to be overestimated and p should be decreased by some factor μ, *e.g.* $p_{new} := (1-\mu)\,p$, where $0 \leq \mu < 1$. To correct MaxExp, the parameter $\bar{N}$ in equation (17) is adjusted such that $1 \leq \bar{N} \leq |\mathcal{N}|$; this has the same effect as decreasing p. Gamma pooling can be corrected by varying γ or predicting it by equation (20) from the optimal $\bar{N}$ of MaxExp. In the next section, the descriptor interdependence is shown in a simulation, with an approach to take further advantage of it.

First, let us introduce a close approximation of MaxExp that has a parameter β accounting for the interdependence of descriptors. Approximate Pooling (**AxMin**) is expressed as:

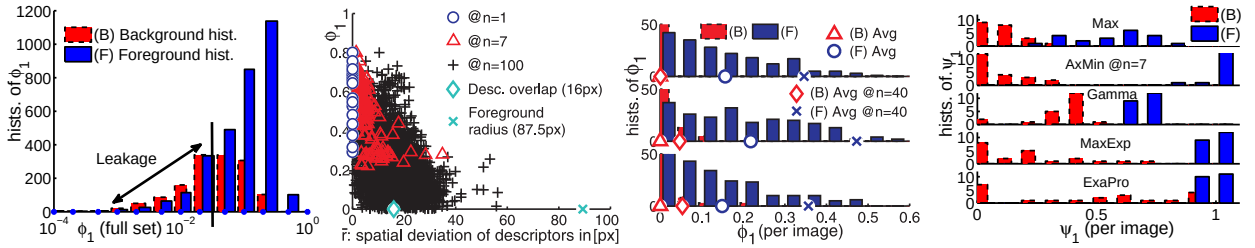
$$\psi_k = \min(1,\, \beta p) = \min\left(1,\, \beta \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn}\right), \quad 1 \leq \beta \leq |\mathcal{N}| \tag{21}$$

The AxMin curve, shown in figure 5 (a, b), follows closely MaxExp and represents a linear magnifying function with a saturation threshold. It can be shown that the steepness β of AxMin and $\bar{N}$ of MaxExp are related such that $\beta \approx \bar{N}$. Parameters β and μ are related by $\beta = |\mathcal{N}|\,(1-\mu)$, hence adjusting β accounts for the interdependence of descriptors. AxMin pooling implies that the confidence in the visual word $\boldsymbol{m}_k$ being present in image i can increase until it reaches the saturation threshold (full confidence). Once reached, any strong variations have no effect which discards the noise. This also prevents the counting of any further occurrences of $\boldsymbol{m}_k$. Such a behaviour increases intra-class similarity of the image signatures and therefore resembles MaxExp and Gamma methods.

To summarise MaxExp, AxMin, and Gamma, figure 5 (d) presents a distribution of coefficients of Average pooling on Flower17 by binning all ψ_k for $k = 1, ..., K$ for all images. Next, Average pooling is corrected with MaxExp, AxMin, and Gamma. The L_2-norm normalisation is applied per image and all signature coefficients h_k are binned. The similar distributions of MaxExp, AxMin, and Gamma highlight their closeness as shown in sections 3.3 and 3.4.

*3.5. Cross Vocabulary Leakage, Descriptor Interdependence, and Improved Pooling (@**n**)*

To understand why Max-pooling is a solid performer despite it being merely a lower bound of *at least one visual word $\boldsymbol{m}_k$ present in image i*, the primary factors that can affect pooling are discussed: i) cross vocabulary leakage, ii) propagated measurement error, iii) descriptor interdependence. These factors are addressed by an improved pooling strategy called @n. Note, terms such as activation and local/non-local activation have been defined in section 2.



(a) (b) (c) (d)

Figure 6: Toy experiment with 21/21 bounding boxes of faces/backgrounds. (a) Histograms of SC activations ϕ_1 for both foreground and background descriptors given visual word $\boldsymbol{m}_1$ that represents a nose. (b) Top 1, 7, and 100 largest activations ϕ_1 given $\boldsymbol{m}_1$ per foreground bounding box as functions of spatial deviation $\bar{r}$ between the descriptors inducing these activations. (c) 6 histograms of activations ϕ_1 given $\boldsymbol{m}_1$ for arbitrarily chosen 3 foreground and 3 background bounding boxes denoted as (F) and (B). Values of Average pooling are marked with circles and triangles, respectively, while Average pooling@$n = 40$ with crosses and diamonds. Note small separation distances between circles and triangles and large between crosses and diamonds. (d) Pooling methods are used to separate 21 faces from 21 backgrounds. Histograms of pooling responses ψ_1 (one ψ_1 per bounding box) given $\boldsymbol{m}_1$ are shown. Foreground and background are labelled as (F) and (B). Refer text for details.

Leakage. Cross vocabulary leakage can be defined as activation $\phi_k \neq 0$ of visual word $\boldsymbol{m}_k$ given descriptor $\boldsymbol{x}$ that should not occur but it does due to: a) the inherent nature of a particular mid-level coding to trigger non-local activations, b) features not representing $\boldsymbol{m}_k$ but having visual appearances similar to $\boldsymbol{m}_k$, hence triggering ϕ_k. Leakage activation $\phi_k \neq 0$ may have an associated correct activation $\phi_{k'} \neq 0$ for $k \neq k'$, hence *cross vocabulary* terminology.

Soft Assignment is used to illustrate case (a). Let us assume descriptor $\boldsymbol{x}$ such that $\boldsymbol{x} = \boldsymbol{m}_k$. This results in activations not related to $\boldsymbol{m}_k$ because $p(\boldsymbol{m}|\boldsymbol{x}, \sigma) > 0$ for any $\boldsymbol{m} \in \mathcal{M} \setminus \{\boldsymbol{m}_k\}$. Similar observations hold for $\boldsymbol{x} \neq \boldsymbol{m}_k$. SA results in large amounts of such a leakage, while LLC and LcSA circumvent this problem by suppressing most non-local activations explicitly in equations (9) and (10). Sparse Coding, however, allows non-local activations.

To illustrate leakage in SC, a toy experiment is introduced. 21 images of a subject's face were captured at similar scales and rotations, backgrounds varied. We applied SIFT [3] (4px grid interval, 16px radii). Next, a descriptor from the first image centred at the tip of the subject's nose was selected. With 32x32 pixel area, it does not cover eyes, lips, or cheeks. It was added as the first element $\boldsymbol{m}_1$ to a dictionary of 4K k-means atoms trained on background images. Descriptors within manually annotated bounding boxes (160x190 pixel) of faces are deemed foreground samples. Further, 21 bounding boxes (160x190 pixel) were selected at random from backgrounds. Figure 6 (a) shows histograms of SC activations ϕ_1 for both foreground and background descriptors. Foregrounds tend to yield the majority of the large responses. Note that below a certain value of ϕ_1, indicated with a vertical bar, background descriptors respond to $\boldsymbol{m}_1$ more often than foreground descriptors. This shows the leakage case (a, b) in practice.

Propagation Error. Having formulated the leakage, the propagation error of MaxExp is computed w.r.t. the average activation $\phi_k = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \Phi_{kn}$ on its input. Applying the first derivative to eq. (17) w.r.t. ϕ_k and assuming a measurement uncertainty $\Delta\phi_k$ representing the leakage error leads to: $\Delta\psi_k = \Delta\phi_k \cdot \bar{N}(1 - \phi_k)^{\bar{N}-1}$. Let us assume $\bar{N}$ to be equal to the average count of descriptors per image, *e.g.* $\bar{N} = 6000$, and the leakage error $\Delta\phi_k = 10^{-5}$. For the sample means $\phi_k = 10^{-5}$ and $\phi_k = 10^{-4}$ the absolute propagation errors are $\Delta\psi_k = 0.056$ and $\Delta\psi_k = 0.032$ respectively. Larger $\Delta\psi_k$ given smaller ϕ_k suggests that MaxExp is sensitive to variations $\Delta\phi_k$ for small ϕ_k and can magnify small perturbations, *e.g.* the leakage. Equivalent findings apply to Gamma and ExaPro. Note that Max-pooling selects only the largest Φ_{kn} over all $n \in \mathcal{N}$. Thus, it can suppress the leakage but it may be less robust to abrupt changes of large Φ_{kn} when compared to analytical pooling. Hence, a compromise between Max-pooling and analytical methods is desired.

Descriptor Interdependence. Section 3.4 discussed the descriptor interdependence and explained how pooling can account for it. Prior knowledge that neighbouring descriptors tend to activate similar visual words can be clearly visualised with our toy example. Let us assume that any two neighbouring descriptors located no more than 16px apart are similar as they overlap heavily. Otherwise, if located more than 16px apart, they have little or no overlap because the descriptor radius is 16px. Thus, descriptors can appear similar only if they describe repeatable image content. Figure 6 (b) illustrates three cases of the top 1, 7, and 100 largest activations ϕ_1 per foreground bounding box responding to our first visual word (the subject's nose). Spatial deviation of the descriptor locations (also per bounding box) given 1, 7, and 100 largest ϕ_1 is indicated along the $\bar{r}$ axis. Interestingly, responses for the top 1 and 7 largest activations are induced by descriptors that are mostly up to 16px apart from each other. Allowing the top 100 largest

Dataset	Splits no.	Train+Val. samples	Test samples	Total images	Dict. size	Descr. type/ dimensions
Caltech101	10x	12+3=15/24+6=30	rest	9144	4K	SIFT/128D
Flowers17	3x	680+340=1020	340	1680	4K	Opp. SIFT/192D
ImageCLEF11	1x	6K+2K=8K	10K	18K	16K	

	Descr. interval	Radii (px)	Descr. per img.	Spatial/other schemes	Kernel types	Classifier used
Caltech101	4,6,8,10px	16,24, 32,40px	5200	none/SCC/SPM	linear	multiclass
Flowers17	8,14,20,26		7900	SCC		
ImageCLEF11	8,12,16,20		4400	SCC/SPM/ DoPM	linear/χ^2_{RBF}	multilabel

Table 2: Summary of the datasets, descriptor parameters, and various experimental details.

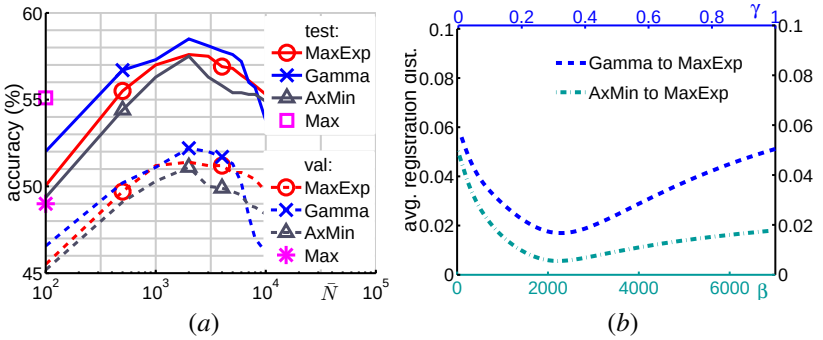
activations reveals that descriptors inducing them are located up to 60px apart. The majority of such descriptors do not cover the subject's nose. This suggests that rejecting low value activations could reduce false positives.

Improved pooling (@n). Reducing the leakage, abrupt changes in large Φ_{kn}, and utilisation of the descriptor interdependency are addressed by simply pooling over the most significant activations given a visual word and the descriptors. This can be easily incorporated into MaxExp, ExaPro, Gamma, and AxMin pooling schemes given in equations (17), (18), (19), and (21) by using the partial sort that selects only the top @n largest values Φ_{kn} over all $n \in \mathcal{N}$ to process, where @n is a parameter. It follows that Max-pooling is a special case, such that @$n = 1$, and a lower bound of ExaPro that can reject the leakage. Hence, @n can be seen as a trade-off between Max-pooling (@n=1) and a chosen analytical approach, where $1 \leq @n \leq |\mathcal{N}|$. The next section shows that mid-level approaches benefit from pooling the top @n most likely activations.

Between-class separation. The overview of the pooling approaches concludes with the toy example introduced in section 3.5 by showing that the @n scheme increases the separation between positive and negative classes compared to other approaches. Foreground bounding boxes of faces are represented by the first atom in the dictionary. This was extracted from the subject's nose as previously outlined. Figure 6 (c) presents 6 histograms of activations ϕ_1 for the first atom given three arbitrarily chosen foreground and background bounding boxes. The resulting values of Average pooling are indicated in the figure with circles and triangles corresponding to the foreground and background distributions respectively. The values of Average pooling@$n = 40$ are marked with crosses and diamonds. Note that Avg@$n = 40$ achieves a superior separation of foreground and background markers compared to Avg. With well adjusted @n, Avg@n (diamonds) penetrates the background distributions far to the left rejecting noise (unlike *e.g.* Max-pooling). Foreground distributions (crosses) are penetrated only marginally to the left. Thus, exploiting the shapes of these distributions improves separability. Figure 6 (d) illustrates pooling methods employed to separate the 21 foreground faces from 21 backgrounds using only pooling responses ψ_1 (one per bounding box) corresponding to the first visual word. The best separation (non-overlapping histograms) is achieved by AxMin@$n = 7$ and the worst separation by Max-pooling (histograms overlap).

4. Experimental Section

The coding and pooling methods are evaluated on the Caltech101 [46], Flower17 [47], and ImageCLEF11 [48] datasets. Approximate Locality-constrained Soft Assignment (LcSA), Approximate Locality-constrained Linear Coding (LLC), Sparse Coding (SC), and Soft Assignment (SA) are compared. Specifically, the baseline performance of selected pooling methods is shown in section 4.2 and their similarity is determined using the registration from section 3.3. Next, the coding and pooling methods are evaluated in section 4.3. LcSA, LLC, and SC mid-level features are processed by Max-pooling (Max), Gamma Correction (Gamma), *theoretical expectation of Max-pooling* (MaxExp), its approximation AxMin, and *at least one visual word* $\boldsymbol{m}_k$ *being present in image* i (ExaPro). Mix-order Max-pooling (MixOrd) and lp-norm are also briefly investigated. The @n scheme from section 3.5 is applied to AxMin and ExaPro to demonstrate it can improve classification performance. The impact of the dictionary size and performance of the coding optimisations from section 2.7 are also measured.



(a) (b)

Figure 7: Baseline LcSA mid-level coding (Caltech101, 15 training images/class, no spatial information, linear kernels). (*a*) LcSA with Max, MaxExp, Gamma, and AxMin pooling. Gamma and AxMin are brought to the MaxExp parameter space $\bar{N}$ by registration (eq. (20), sec. 3.3). Dashed and solid curves show the validation and test results. (*b*) Corresponding average registration distance between Gamma/AxMin and MaxExp signatures highlights their closeness.

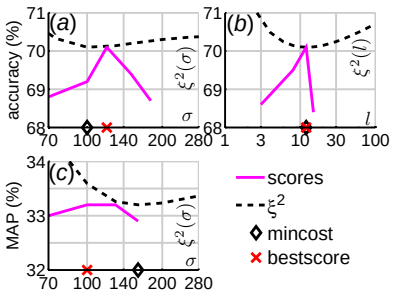


Figure 8: ξ^2 quantisation loss (dashed curves) and classification results (solid curves) as functions of σ and l. We varied (*a*) σ, (*b*) l on Caltech101, (*c*) σ on ImageCLEF11. Diamonds and crosses indicate the minima of ξ^2 and the best results. See text for details.

4.1. Experimental Arrangements and Datasets

The Caltech101 [46] set consists of 101 classes represented by objects which are aligned to the centres of images as well as a separate background class. The majority of evaluations are performed with 15 training images per class (unless otherwise stated). The Flower17 [47] set of 17 flower classes was used for further evaluations (data splits are supplied for this corpus). ImageCLEF11 Photo Annotation [48] 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. ImageCLEF11 is a subset of MIRFLICKR with vastly improved annotations which enables better classification [49, 50]. To evaluate the mid-level coding and pooling methods in a simple framework, only Opponent SIFT on a dense grid was used for this set. Only the visual annotation was used in this study. To best use the evaluated coding methods on ImageCLEF11, the training set was doubled by left-right flipping training images [33]. Table 2 presents the experimental parameters[2].

Dictionary. K-means was used throughout the experiments. However, Fast Hierarchical Nearest Neighbour Search, described in section 2.7, employs 64x64 and 128x128 hierarchical k-means on Caltech101 and ImageCLEF11.

Dataset bias. Spatial relations in images were exploited by either Spatial Coordinate Coding (SCC) [32] or Spatial Pyramid Matching [30]. SPM used 4 levels of coarseness with 1x1, 2x2, 3x3, and 4x4 grids. Dominant Angle Pyramid Matching (DoPM) [32] was used to exploit dominant edge bias in ImageCLEF11. DoPM used 5 levels of coarseness with 1, 3, 6, 9, and 12 grids. SCC and DoPM are introduced below.

Kernels. Linear kernels $ker_{ij} = (\boldsymbol{h}_i)^T \cdot \boldsymbol{h}_j$ were used, where $\boldsymbol{h}_i, \boldsymbol{h}_j \in \mathbb{R}^{KQ}$ 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 also used, $1/\rho$ is the RBF radius.

Classifier. Multi-class KDA [7] was applied to both Clatech101 and Flower17 to process kernels formed from different mid-level feature and pooling variants. Mean Accuracy is the reported performance measure. Multi-label KDA [7] was applied to ImageCLEF11, as it was previously found to be a robust performer on this set [51]. Due to the multi-label nature of ImageCLEF11, Mean Average Precision [7] (MAP) is used to report the performance.

SCC and DoPM. Spatial Coordinate Coding is proposed as a computationally efficient alternative to Spatial Pyramid Matching [32]. It is performed on the descriptor level by augmenting descriptor vectors $\boldsymbol{x}_n$ with their spatial positions $\boldsymbol{x}_n^{'}$ normalised with respect to image width and height: $\boldsymbol{x}_n^{aug} = [\sqrt{1-\omega}\boldsymbol{x}_n^T, \sqrt{\omega}(\boldsymbol{x}_n^{'})^T]^T$. The trade-off between the visual appearance and spatial bias is balanced by ω, which is determined experimentally by cross-validation. A single training kernel for Caltech101 (30 images/class), given the parameters specified in table 2, can be computed in 37s and 1340s with SSC and SPM respectively. If SCC is used, SPM is disabled by setting the number of its partitions $Q = 1$ (see equations (2) and (3), section 2). Dominant Angle Pyramid Matching (DoPM), proposed in [32], exploits orientations of dominant edges. Such orientations are specific for some objects, *e.g.* trunks of trees are likely to maintain vertical positions $\theta \in O_{tree}$. Thus, confidence in observing a tree increases $p(o = tree|\theta) \geq p(o = tree)$ if $\theta \in O_{tree}$. In practice, dominant angles from SIFT are used to split mid-level features built from rotation-invariant SIFT into Q sets and pooling is performed in each set.

[2]We plan to release an evaluation on additional datasets, *e.g.* PASCAL VOC 2007. See http://claret.wikidot.com for more results.

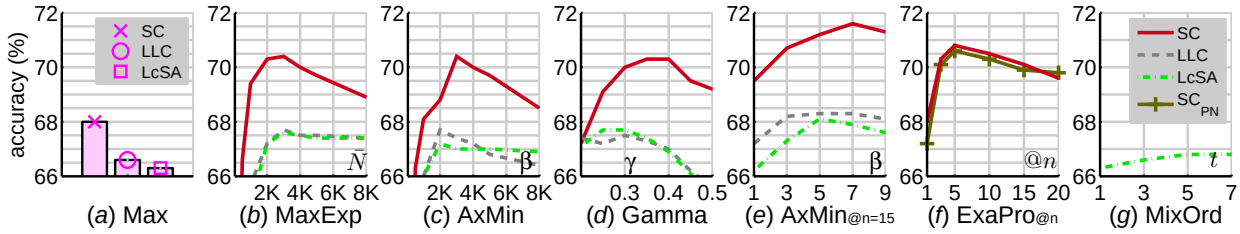


Figure 9: Performance of mid-level coding methods LcSA, LLC, and SC given pooling methods (Caltech101, 15 images/class, Spatial Coordinate Coding, linear kernels). The following are (*a*) baseline Max-pooling, (*b*) MaxExp pooling as a function of $\bar{N}$, (*c*) its close approximation AxMin pooling as a function of β, (*d*) Gamma pooling given γ, (*e*) AxMin@n = 15 as a function of β, (*f*) ExaPro@n for positive (in solid) and positive-negative activations (SC_{PN}) of SC as discussed in section 2.3, and (g) MixOrd pooling.

4.2. Baseline Performance and Registration between Gamma/AxMin and MaxExp.

The baseline performance of LcSA mid-level coding paired with various pooling methods is determined for Caltech101 (15 training images/class, no spatial information). Several sets of image signatures are computed on the training data for Gamma, AxMin, and MaxExp pooling given several values of their parameters γ, β, and $\bar{N}$. Next, registration between the signatures of Gamma/AxMin and MaxExp is performed by minimising equation (20) from section 3.3. For each $\bar{N}$, a corresponding γ and β is found. Figure 7 (a) shows the classification results on both validation and test sets. Results for MaxExp, Gamma, and AxMin pooling are shown as functions of the common parameter $\bar{N}$ due to the registration. The three curves shown have peak performance for the same value of $\bar{N}$, indicating that Gamma and AxMin act on mid-level features similar to MaxExp. This supports our discussion in sections 3.3 and 3.4 regarding the common theoretical basis of these methods. Figure 7 (b) shows the average Euclidian registration distance between Gamma/AxMin and MaxExp signatures as a function of parameters γ and β. Parameters $\gamma = 0.32$ and $\beta = 2200$ indicate the attained minima and correspond to the optimal $\bar{N} = 2000$ selected from plot 7 (a).

Further, figure 7 (a) shows the baseline Max-pooling accuracy of 55.1% on the test set. Gamma improved on this score by 3.4%, reaching 58.5% accuracy. The Average pooling is not reported in the following sections as it scored only 42.6% accuracy and consistently underperformed. Note that peaks in accuracy on the validation and test sets match each other closely. Thus, only performance achieved on test sets is reported in further sections. However, various parameters of the classification pipeline were determined during cross-validation on validation sets.

Lastly, figure 8 shows the classification results for the baseline Max-pooling as a function of LcSA coding parameters σ and l, respectively. Caltech101 (15 images/class, Spatial Pyramid Matching) and ImageCLEF11 (Spatial Coordinate Coding) were evaluated both on linear kernels. The best coding parameters, indicated by crosses, seem to correlate well with the minima of ξ^2, as indicated by diamonds. The above parameters were found by evaluating equation (12) given 156K descriptors per dataset that were drawn at random.

4.3. Evaluations of Mid-level Coding and Pooling Methods

This section describes how the coding and pooling methods performed in a practical classification scenario. The impact of pooling parameters on the classification is shown first. Next, the best scores of each coding and pooling pair are reported to facilitate comparisons. Additional components and kernel choices are described for each experiment.

Caltech101. Figure 9 introduces results for the coding and pooling methods as functions of the pooling parameters (15 training images/class, Spatial Coordinate Coding). Note that there are no erratic variations in plots. The best performance for each method corresponds to the peak of each curve (peaks on the validation and test set also matched each other). Plot 9 (a) shows that the baseline Max-pooling yields 68.0 ± 0.5%, 66.6 ± 0.4%, and 66.3 ± 0.3% accuracy for SC, LLC, and LcSA, respectively. Plots 9 (b-d) show the accuracy for MaxExp, AxMin, and Gamma. SC yields 70.4 ± 0.4% accuracy for all three schemes. LLC and LcSA achieve 67.7 ± 0.5% accuracy with AxMin and Gamma, respectively. Improvements over Max-pooling given SC, LLC, and LcSA amount to 2.4%, 1.1%, and 1.4%, respectively. Note that MaxExp scored best for $\bar{N} \approx 3000 < 5200$ (mean descriptor count). Figure 9 (e) shows that AxMin@n = 15 with SC yields 71.6 ± 0.4% giving a 3.4% improvement over Max-pooling due to the @n scheme. LLC and LcSA score 68.3 ± 0.4% and 68.1 ± 0.5%. Figure 9 (f) shows scores for ExaPro@n and SC that amount to 70.8 ± 0.3% and 70.6 ± 0.3% given the positive and positive-negative activations respectively. As suggested in section 2.3, no benefits of allowing $\phi_k < 0$ were observed. Next, plot (g) shows MixOrd given LcSA ($t = 1, 3, 5, 7$ signatures

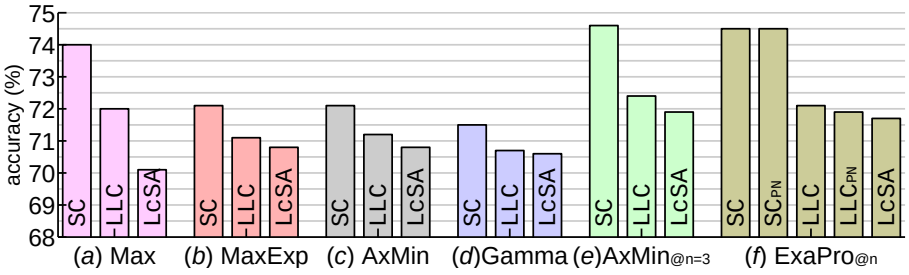


Figure 10: Performance of mid-level coding methods LcSA, LLC, and SC given pooling methods (Caltech101, 15 images/class, Spatial Pyramid Matching, linear kernels). SC, LLC, and LcSA are paired with (*a*) baseline Max-pooling, (*b*) MaxExp, (*c*) AxMin, (*d*) Gamma, (*e*) AxMin@n = 3, and (*f*) ExaPro@n. SC_{PN} and LLC_{PN} show results for SC and LLC given the positive-negative activations.

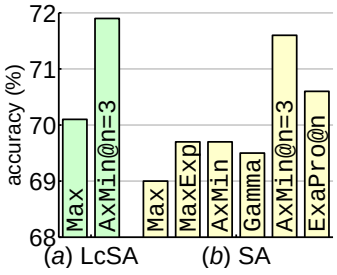


Figure 11: SA scores low given Max-pooling, MaxExp, AxMin, and Gamma. Note, SA and LcSA perform similar for AxMin@n = 3.

per image were combined as described in section 3.1). This resulted in an 0.8% increase over Max-pooling. Not included in the plots, lp-norm and LcSA yields 66.4 ± 0.5% at best, ExaPro and LLC yields 68.2 ± 0.5%.

Figure 10 shows additional performance results of coding and pooling (15 training images/class, Spatial Pyramid Matching). Plot 10 (a) shows that the baseline Max-pooling scores 74.0 ± 0.3%, 72.0 ± 0.5% and 70.1 ± 0.4% given SC, LLC, and LcSA. Plots 10 (b-d) show scores for MaxExp, AxMin, and Gamma. Performance of SC and LLC deteriorated for these three schemes. LcSA scores 70.8 ± 0.5%, yielding a small improvement. Plot 10 (e) shows the positive impact of AxMin@n = 3 on the coding methods. SC and LLC improve marginally from 74.0 ± 0.3% and 72.0 ± 0.5% given Max-pooling to 74.6 ± 0.4% and 72.4 ± 0.5% accuracy. LcSA yields 71.9 ± 0.4% giving a 1.8% improvement over Max-pooling. Plot 10 (f) shows ExaPro@n with SC reaching 74.5 ± 0.4% and LLC achieving 72.1 ± 0.3%. Note that alowing positive-negative activations does not improve the performance. Not in the plots, lp-norm and MixOrd yield 70.3 ± 0.3% and 70.1 ± 0.4% at best. Table 3 summarises the best scores achieved by this study on Caltech101 (15 and 30 training images/class). This is compared to various results achieved by others in table 4. The best results reported in the literature are Group-Sensitive Multiple Kernel Learning (GS-MKL) [52] with performance of 84.3%, Discriminative Affine Sparse Codes (ASIFT) [53] with 83.3%, Multi-way SVM on appearance and shape features (M-SVM) [54] with 81.3%, and Graph-matching Kernel (GMK) [55] with 80.3% accuracy.

Soft Assignment and Leakage. Section 3.5 discussed Soft Assignment and the problem of the inherent leakage in this method. The experimental findings are shown in figure 11 (Caltech101, 15 training images/class, Spatial Pyramid Matching) and present SA given a variety of pooling methods. SA scores only 69.0 ± 0.6% accuracy given

	SA AxMin@n	LcSA AxMin@n	LcSA Max
SCC (15)	67.8 ± 0.6	68.1 ± 0.5	66.3 ± 0.3
SPM (15)	71.6 ± 0.4	71.9 ± 0.4	70.1 ± 0.4
SPM (30)	78.6 ± 0.5	78.8 ± 0.4	77.8 ± 0.3
	LLC AxMin@n	SC AxMin@n	SC Max
SCC (15)	68.3 ± 0.4	**71.6 ± 0.4**	68.0 ± 0.5
SPM (15)	72.4 ± 0.5	**74.6 ± 0.4**	74.0 ± 0.3
SPM (30)	79.5 ± 0.5	**81.3 ± 0.6**	80.4 ± 0.6

Table 3: Summary of our best results on Caltech101. The first column indicates how the spatial information was injected. Numbers of training images per class are indicated in brackets.

Boureau et al. [29]	HA, 1K, MaxExp	71.8 ± 0.8
Chatfield et al. [33]	HA, 8K, Avg+χ^2	74.2 ± 0.6
Chatfield et al. [33]	SA, 8K, Avg+χ^2	75.9 ± 0.6
Liu et al. [12]	LcSA, 1K, Max	76.5 ± 0.7
Wang et al. [14]	LLC, 1K, Max	73.4
Chatfield et al. [33]	LLC, 8K, Max	76.9 ± 0.4
Yang et al. [14]	SC, 1K, Max	73.2 ± 0.5
Boureau et al. [29]	SC, 1K, Max, MF	75.1 ± 0.9
Boureau et al. [31]	SC, 1K x64, CSP	77.1 ± 0.7
Chatfield et al. [33]	Fisher, 256x256, Gamma	77.8 ± 0.6
Duchenne et al. [55]	GMK	80.3 ± 1.2
Bosch et al. [54]	M-SVM	81.3 ± 0.8
Kulkarni and Li [53]	ASIFT	83.3
Yang et al. [52]	GS-MKL	84.3

Table 4: Results on Caltech101 (30 training images/class) reported in the literature. Mid-column: coding type, signature length, and pooling. MF are Macrofeatures [29], CSP is Pooling in Configuration Space [31]. The last four rows show the highest results (acronyms explained in text).

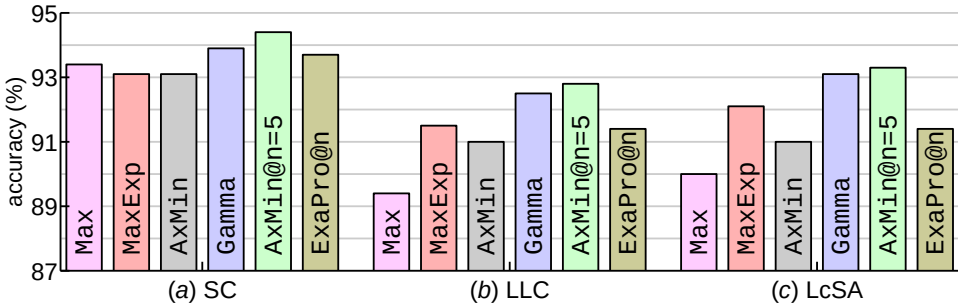


Figure 12: Performance of mid-level coding methods for various pooling schemes (Flower17, Spatial Coordinate Coding, linear kernels). Plots (*a-c*) show results for SC, LLC, and LcSA, respectively. Note that the majority of pooling schemes outperform Max-pooling.

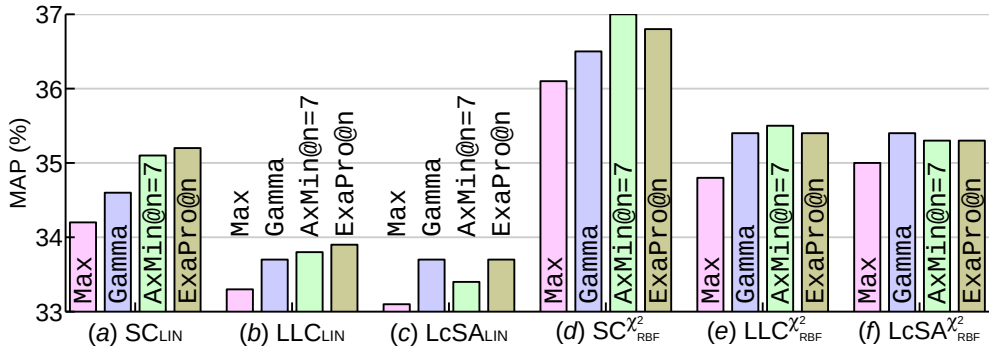


Figure 13: Performance of mid-level coding and pooling (ImageCLEF11, Spatial Coordinate Coding). SC, LLC, and LcSA are paired with Max-pooling, Gamma, AxMin@n = 7, and ExaPro@n. We used (*a-c*) linear and (*d-f*) χ^2_{RBF} kernels.

Max-pooling. MaxExp, AxMin, and Gamma yield small improvements. However, applying AxMin@n = 3 to SA yields a 2.6% improvement over Max-pooling leading to 71.6 ± 0.4% accuracy. For comparison, LcSA with AxMin@n = 3 scores 71.9 ± 0.4%. Note that Max-pooling scores poorly despite being a special case of @n pooling, *e.g.* AxMin@n = 1. We suspect that exploiting the descriptor interdependency (@n > 1), as outlined in section 3.5, is important in tackling the leakage.

Flower17. Plots 12 (a-c) show results for SC, LLC, and LcSA for various pooling schemes (Spatial Coordinate Coding, linear kernels). Plot 12 (a) shows that SC combined with either MaxExp or AxMin has a performance below the baseline Max-pooling which yields 93.4 ± 0.3%. However, SC with Gamma gives 93.9 ± 1.6% accuracy. SC with AxMin@n = 5 scores 94.4 ± 0.4%. LLC in plot 12 (b) also improves over its baseline of 89.4 ± 1.6% accuracy reaching 92.6 ± 1.8% and 92.8 ± 0.5% for Gamma and AxMin@n = 5, which is a 3.4% improvement. LcSA in plot 12 (c) scores 93.1 ± 1.1% and 93.3 ± 0.5% accuracy for Gamma and AxMin@n = 5. This is a 3.3% improvement over the Max-pooling baseline of 90.0 ± 0.2%. Table 5 summarises our results. The best results in previous studies are [32] with 91.4%, [47] with 88.3%, [56] with 88.2%, and [57] with 86.7% accuracy.

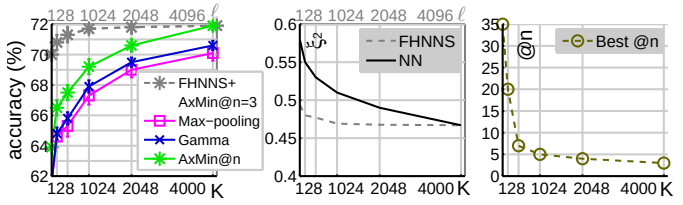
ImageCLEF11. To conclude the coding and pooling experiments on a challenging set, SC, LLC, and LcSA are paired with Max-pooling, Gamma, AxMin@n = 7, and ExaPro@n. MaxExp and AxMin are not reported as they perform similar to Gamma. Spatial Coordinate Coding was used in these tests. Plots 13 (a-c) show results on linear kernels.

	LcSA	LLC	SC
Max	90.0 ± 0.2	89.4 ± 1.6	**93.4 ± 0.3**
Gamma	93.1 ± 1.1	92.5 ± 1.1	**93.9 ± 1.6**
AxMin@n	93.3 ± 0.5	92.8 ± 0.8	**94.4 ± 0.4**

Table 5: Summary of the best results attained by us on Flower17 (Spatial Coordinate Coding and linear kernels were used). The first column indicates the pooling type: Max, Gamma, and AxMin@n = 5.

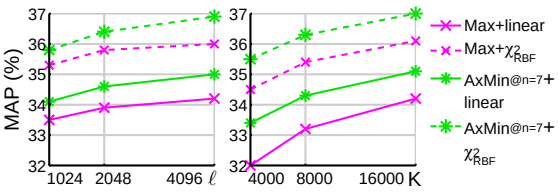
	SCC	SPM	DoPM	Comb.
linear	35.1	35.2	35.3	36.6
χ^2_{RBF}	**37.0**	36.7	36.8	**38.4**

Table 6: Our best results on ImageCLEF11 (Sparse Coding and AxMin@n = 7). First column: kernel type. First row: bias type. Comb. denotes SPM and DoPM combined.



(a) (b) (c)

Figure 14: Performance of LcSA given Fast Hierarchical Nearest Neighbour Search (section 2.7) and ordinary NN (Caltech101, 15 training images/class, Spatial Pyramid Matching). (*a*) LcSA with FHNNS as a function of ℓ (cluster dilation). Also, LcSA with NN as a function of K (dictionary size) for Max, Gamma, and AxMin@n. (*b*) Corresponding quantisation errors ξ^2. (*c*) The optimal value @n for AxMin@n as a function of the dictionary size K.



(a) SC+FHNNS (b) SC

Figure 15: Performance of SC given FHNNS and ordinary NN (ImageCLEF11, Spatial Coordinate Coding). We applied linear and χ^2_{RBF} kernels to Max-pooling and AxMin@n = 7 based signatures. (*a*) SC with FHNNS as a function of ℓ (cluster dilation). (*b*) SC with NN as a function of K (dictionary size).

Max-pooling scores 34.2%, 33.3%, and 33.0% MAP given SC, LLC, and LcSA. Figure 13 (a) shows AxMin@n = 7 and ExaPro@n yield 35.1% and 35.2% for SC. This gives a 1% improvement over Max-pooling (the best result on linear kernels). LLC and LcSA yield 33.9% and 33.8% for ExaPro@n and Gamma, respectively.

Plots 13 (d-f) show results on χ^2_{RBF} kernels that improve performance further. Plots 13 (b) show that Max-pooling yields 36.1%, 34.9%, and 35.0% MAP given SC, LLC, and LcSA. Next, AxMin@n = 7 scores 37.0% (the best result on χ^2_{RBF} kernels). This is 0.9% improvement over Max-pooling. Lastly, LLC and LcSA yield 35.5% and 35.4% given AxMin@n = 7 and Gamma. The evaluated pooling schemes again improved results over the baseline on both kernel types. We note a trend that LcSA works well with Gamma (also MaxExp and AxMin in previous sections). SC and LLC tend to benefit more from AxMin@n and ExaPro@n. Also, LLC and LcSA yield very similar results.

ImageCLEF11 and Bias in Images. Given the complexity of ImageCLEF11, Spatial Pyramid Matching (SPM) and Dominant Angle Pyramid Matching (DoPM, section 4.1) were emplyed for the final experiments (Sparse Coding, AxMin @n = 7, linear and χ^2_{RBF} kernels used). Table 6 shows results for SPM and DoPM. Given linear kernels, they have a performance of 35.2% and 35.3% MAP. For χ^2_{RBF}, they yield 36.7% and 36.8%. Furthermore, combining either SCC (scored 37.0%) or SPM with DoPM yields 38.4% MAP. Only Opponent SIFT on a dense grid is used. The best results in previous studies for the visual configuration are 38.8% [58] (multiple interest points, descriptors, and kernels combined) and 38.2% [59] (multiple semantic contexts, SPM channels, sematic features, and kernels combined).

Dictionary Size and Fast Hierarchical Nearest Neighbour Search. To conclude these evaluations, there follows a brief investigation into: i) the impact of the dictionary size on LcSA and SC, ii) Fast Hierarchical Nearest Neighbour Search (FHNNS), outlined in section 2.7, paired with LcSA and SC.

Dictionary Size. Figure 14 (a) shows the performance on Caltech101 (15 training images/class, Spatial Pyramid Matching, linear kernels used) for LcSA given Max-pooling, Gamma, and AxMin@n. The dictionary size K was varied. Max-pooling and Gamma perform similar for $K \in \langle 128; 512 \rangle$. Gamma scores marginally better than Max-pooling for larger K. AxMin@n appears a strong performer even for small K. Plot 14 (c) shows how the best performing parameter @n of AxMin@n varies as a function of K. Figure 15 (b) shows that ImageCLEF11 (SC, Spatial Coordinate Coding, χ^2_{RBF} kernels used) benefits from a larger dictionary.

FHNNS. Figure 14 (a) also presents the results for LcSA with FHNNS and AxMin@n = 3 using K' = 4096 atoms. Given $\ell \ll K'$ (ℓ impacts the cluster dilation), LcSA and FHNNS had a higher performance than LcSA and Nearest Neighbour. The first approach searches through only ℓ anchors to code a descriptor. However, it still produces K' long mid-level features. The latter method searches through $K = \ell$ anchors and produces only K long features in a comparable coding time. Hence, its performance drops for small values of K. Plot 14 (b) shows the corresponding quantisation error for LcSA with FHNNS is smaller when compared to LcSA with NN (assuming $K = \ell \ll K'$). Lastly, figure 15 (a) presents the classification results for SC with FHNNS on ImageCLEF11. Given ℓ = 4096 and K' = 16384, this method is as robust as ordinary SC in figure 15 (b) and saves on computational cost (see table 1).

4.4. Discussion on the Coding and Pooling Approaches

Mid-level coding methods differ both in their classification performance (section 4.3) and computational cost (table 1). SA, LcSA, LLC, and SC exhibited varied performance depending on the pooling variant. Further, a strong

relation is observed between Gamma and MaxExp pooling, as discussed in section 3.3, and empirically validated in figures 7 (a, b). Classification experiments also suggest these two methods are similar. In practice, using a carefully selected pooling methods led to significant improvements over the baseline Max-pooling approach. Specifically, LcSA and LLC benefited from MaxExp, AxMin, Gamma, and the @n pooling schemes. SC and SA demonstrated their best performance during the classification when paired with the @n scheme. This may be attributed to the leakage suppression discussed in section 3.5. Furthermore, carefully selected pooling parameters led to the best classification performance by accounting for the descriptor interdependence, as outlined in sections 3.4 and 3.5. AxMin@n and ExaPro@n are examples of extending AxMin and ExaPro pooling with the @n scheme. Note that SC consistently outperformed LcSA and LLC, but at the price of higher computational cost. Regarding computational efficiency, Fast Hierarchical Nearest Neighbour Search, from section 2.7, benefited the coding as shown in section 4.3. Combining LcSA and SC with FHNNS improved their computational speed 4x and 1.5x (table 1) with no observable decline in the classification results. Large overlap between the k-means dictionary clusters was required to limit the quantisation noise along the cluster boundaries. Lastly, the impact of Spatial Coordinate Coding, Spatial Pyramid Matching, and Dominant Angle Pyramid Matching on the classification quality was evaluated. Due to the compactness of mid-level features generated with SCC, it thrived on the discriminative properties of the @n scheme, as explained in section 3.5. Note that computing kernels from SCC based signatures was 36x faster than using SPM signatures (section 4.1). Moreover, SCC yielded better performance than SPM on ImageCLEF11. Combining SCC/SPM and DoPM gave the best final performance.

Pipeline Variants. For rapid classification, LcSA/LLC with FHNNS, MaxExp/Gamma pooling, Spatial Coordinate Coding, and a linear kernel is effective. For large complex datasets, SC, AxMin@n, SPM, DoPM, and χ^2_{RBF} kernels may be used. For small datasets, SC, AxMin@n, Spatial Coordinate Coding, and a linear kernel are a good choice.

5. Conclusions

This paper is an extensive comparison of four widely used mid-level coding schemes on three popular datasets. Various pooling strategies were evaluated to asses their impact on classification. We demonstrated that the performance of SA, LcSA, LLC, and SC schemes depends on the choice of pooling. Evaluated MaxExp, Gamma, AxMin, and ExaPro improved the performance over the baseline Max-pooling scheme. Furthermore, we proposed a simple extension termed @n which is applicable to these pooling schemes. Its positive impact on performance with AxMin@n and ExaPro@n pooling is observed. SC outperformed SA, LcSA, and LLC on the evaluated datasets leading to 81.3% accuracy on Caltech101, 94.4% accuracy on Flower17, and 38.4% MAP on ImageCLEF11 (visual configuration, Opponent SIFT used only). LLC and LcSA were close competitors. In the future, we plan to experiment with Fisher encoding and pooling schemes. We also plan to merge optimisation of the pooling parameters with the classifier.

Acknowledgements. This work was supported by the EU CHIST-ERA, UK EPSRC EP/K01904X/1, and BBC R&D grants. We would like to thank Mark Barnard, David Windridge, and Tim Sheerman-Chase for their help in improving readability of the manuscript. Furthermore, we would like to thank Y-Lan Boureau and Teo De Campos for several insightful discussions, Julien Mairal for providing us with an updated Lasso solver, and Bevis King for his frequent IT support outside of regular working hours.

References

[1] J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, ICCV 2 (2003) 1470–1477.
[2] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual Categorization with Bags of Keypoints, ECCV Workshop on Statistical Learning in Computer Vision (2004) 1–22.
[3] D. G. Lowe, Object Recognition from Local Scale-Invariant Features, CVPR 2 (1999) 1150–1157.
[4] K. Mikolajczyk, C. Schmid, A Performance Evaluation of Local Descriptors, PAMI 27 (2005) 1615–1630.
[5] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, A Comparison of Color Features for Visual Concept Classification, CIVR (2008) 141–149.
[6] C. Cortes, V. Vapnik, Support-Vector Networks, ML 20 (1995) 273–297.
[7] M. Tahir, J. Kittler, K. Mikolajczyk, F. Yan, K. van de Sande, T. Gevers, Visual Category Recognition using Spectral Regression and Kernel Discriminant Analysis, ICCV Workshop on Subspace Methods (2009).
[8] J. van Gemert, J.-M. Geusebroek, C. Veenman, A. Smeulders, Kernel Codebooks for Scene Categorization, ECCV 5304 (2008) 696–709.
[9] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, J. M. Geusebroek, Visual Word Ambiguity, PAMI (2010).

[10] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases., CVPR (2008).
[11] P. Koniusz, K. Mikolajczyk, Soft Assignment of Visual Words as Linear Coordinate Coding and Optimisation of its Reconstruction Error, ICIP (2011).
[12] L. Lingqiao, L. Wang, X. Liu, In Defence of Soft-assignment Coding, ICCV (2011).
[13] H. Lee, A. Battle, R. Raina, A. Y. Ng, Efficient Sparse Coding Algorithms, NIPS (2007) 801–808.
[14] J. Yang, K. Yu, Y. Gong, T. S. Huang, Linear Spatial Pyramid Matching using Sparse Coding for Image Classification, CVPR (2009) 1794–1801.
[15] S. Mallat, Z. Zhang, Matching Pursuit with Time-Frequency Dictionaries, TSP 41 (1993) 3397–3415.
[16] J. A. Tropp, Greed is Good: Algorithmic Results for Sparse Approximation, TIT 50 (2004) 2231–2242.
[17] K. Yu, T. Zhang, Y. Gong, Nonlinear Learning using Local Coordinate Coding, NIPS (2009).
[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained Linear Coding for Image Classification, CVPR (2010).
[19] S. Gao, I. W. Tsang, L. Chia, P. Zhao, Local Features Are Not Lonely - Laplacian Sparse Coding for Image Classification, CVPR (2010).
[20] J. Yang, K. Yu, T. Huang, Efficient Highly Over-Complete Sparse Coding using a Mixture Model, ECCV (2010) 113–126.
[21] F. Perronnin, C. Dance, Fisher Kernels on Visual Vocabularies for Image Categorization, CVPR 0 (2007) 1–8.
[22] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher Kernel for Large-Scale Image Classification, ECCV (2010) 143–156.
[23] X. Zhou, K. Yu, T. Zhang, T. S. Huang, Image Classification using Super-Vector Coding of Local Image Descriptors, ECCV (2010) 141–154.
[24] H. Jegou, M. Douze, C. Schmid, P. Pérez, Aggregating Local Descriptors into a Compact Image Representation, CVPR (2010) 3304–3311.
[25] R. Negrel, D. Picard, P.-H. Gosselin, Compact Tensor Based Image Representation for Similarity Search, ICIP (2012).
[26] L. Wang, Toward A Discriminative Codebook: Codeword Selection across Multi-resolution, CVPR 0 (2007) 1–8.
[27] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results, http://pascallin.ecs.soton.ac.uk/challenges/VOC, 2010.
[28] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning Mid-Level Features for Recognition, CVPR (2010).
[29] Y. Boureau, J. Ponce, Y. LeCun, A Theoretical Analysis of Feature Pooling in Vision Algorithms, ICML (2010).
[30] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, CVPR 2 (2006) 2169–2178.
[31] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, Y. LeCun, Ask the Locals: Multi-way Local Pooling for Image Recognition, ICCV (2011).
[32] P. Koniusz, K. Mikolajczyk, Spatial Coordinate Coding to Reduce Histogram Representations, Dominant Angle and Colour Pyramid Match, ICIP (2011).
[33] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods, BMVC (2011).
[34] J. Yang, Y.-G. Jiang, A. G. Hauptmann, C.-W. Ngo, Evaluating Bag-of-Visual-Words Representations in Scene Classification, MIR (2007) 197–206.
[35] A. Coates, A. Ng, The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization, ICML (2011) 921–928.
[36] I. Tosic, P. Frossard, Dictionary Learning, SPM 28 (2011) 27–38.
[37] T. Liu, A. W. Moore, A. Gray, K. Yang, An Investigation of Practical Approximate Nearest Neighbor Algorithms, NIPS (2004) 825–832.
[38] T. Tuytelaars, C. Schmid, Vector Quantizing Feature Space with a Regular Lattice, ICCV (2007).
[39] J. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Technical Report, 1998.
[40] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least Angle Regression, Annals of Statistics 32 (2004) 407–499.
[41] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online Learning for Matrix Factorization and Sparse Coding, JMLR (2010).
[42] The MOSEK Optimization Software, http://www.mosek.com, 2012.
[43] S. Boughorbel, J.-P. Tarel, N. Boujemaa, Generalized Histogram Intersection Kernel for Image Recognition, ICIP (2005) 161–164.
[44] H. Jégou, M. Douze, C. Schmid, On the Burstiness of Visual Elements, CVPR (2009) 1169–1176.
[45] T. Jebara, R. Kondor, A. Howard, Probability Product Kernels, JMLR 5 (2004) 819–844.
[46] L. Fei-fei, R. Fergus, 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).
[47] M. E. Nilsback, A. Zisserman, Automated Flower Classification over a Large Number of Classes, ICVGIP (2008) 722–729.
[48] S. Nowak, K. Nagel, J. Liebetra, The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks, CLEF (2011).
[49] M. J. Huiskes, M. S. Lew, The MIR Flickr Retrieval Evaluation, MIR (2008) 39–43.
[50] M. J. Huiskes, B. Thomee, M. S. Lew, New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative, MIR (2010) 527–536.
[51] M. A. Tahir, F. Yan, M. Barnard, M. Awais, K. Mikolajczyk, J. Kittler, The University of Surrey Visual Concept Detection System at ImageCLEF 2010: Working Notes, ICPR (2010).
[52] J. Yang, Y. Tian, L.-Y. Duan, T. Huang, W. Gao, Group-Sensitive Multiple Kernel Learning for Object Recognition, TIP 21 (2012) 2838–2852.
[53] N. Kulkarni, B. Li, Discriminative Affine Sparse Codes for Image Classification, CVPR (2011) 1609–1616.
[54] A. Bosch, A. Zisserman, X. Munoz, Image Classification using Random Forests and Ferns, ICCV (2007).
[55] O. Duchenne, A. Joulin, J. Ponce, A Graph-Matching Kernel for Object Categorization, ICCV (2011).
[56] J. Liu, C. Zhang, Q. Tian, C. Xu, H. Lu, S. Ma, One Step Beyond Bags of Features: Visual Categorization using Components, ICIP (2011).
[57] F. Yan, K. Mikolajczyk, M. Barnard, H. Cai, J. Kittler, Lp Norm Multiple Kernel Fisher Discriminant Analysis for Object and Image Categorisation, CVPR (2010).
[58] A. Binder, W. Samek, M. Kawanabe, The joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF 2011 Photo Annotation Task: Working Notes, CLEF (2011).
[59] Y. Su, F. Jurie, Semantic Contexts and Fisher Vectors for the ImageCLEF 2011 Photo Annotation Task, CLEF (2011).