Chapter 2
Bag-of-Words Image Representation: Key Ideas and Further Insight

Marc T. Law, Nicolas Thome and Matthieu Cord

Abstract In the context of object and scene recognition, state-of-the-art performances are obtained with visual Bag-of-Words (BoW) models of mid-level representations computed from dense sampled local descriptors (e.g., Scale-Invariant Feature Transform (SIFT)). Several methods to combine low-level features and to set mid-level parameters have been evaluated recently for image classification. In this chapter, we study in detail the different components of the BoW model in the context of image classification. Particularly, we focus on the coding and pooling steps and investigate the impact of the main parameters of the BoW pipeline. We show that an adequate combination of several low (sampling rate, multiscale) and mid-level (codebook size, normalization) parameters is decisive to reach good performances. Based on this analysis, we propose a merging scheme that exploits the specificities of edge-based descriptors. Low and high contrast regions are pooled separately and combined to provide a powerful representation of images. We study the impact on classification performance of the contrast threshold that determines whether a SIFT descriptor corresponds to a low contrast region or a high contrast region. Successful experiments are provided on the Caltech-101 and Scene-15 datasets.

M. T. Law (✉) · N. Thome · M. Cord
LIP6, UPMC—Sorbonne University, Paris, France
e-mail: Marc.Law@lip6.fr

N. Thome
e-mail: Nicolas.Thome@lip6.fr

M. Cord
e-mail: Matthieu.Cord@lip6.fr

B. Ionescu et al. (eds.), *Fusion in Computer Vision*, Advances in Computer Vision and Pattern Recognition, DOI: 10.1007/978-3-319-05696-8_2,

2.1 Introduction

Image classification refers to the ability of predicting a semantic concept based on the visual content of the image. This topic is extensively studied due to its large number of applications in areas as diverse as Image Processing, Information Retrieval, Computer Vision, and Artificial Intelligence [10]. This is one of the most challenging problems in these domains. For instance, in the context of Computer Vision, the ability to predict complex semantic categories, such as scenes or objects, from the pixel level, is still a very hard task. How to properly represent images for successfully categorizing images, i.e., filling the semantic gap, remains a major issue for computer vision researchers.

Different methodologies have been explored in the last decade to fulfill this goal. Biologically inspired models [37, 42] try to mimic the mammalian visual system, and show interesting performances for classification and detection. Recently, deep learning has attracted lots of attention due to the large success of deep convolutional nets in the Large Scale Visual Recognition Challenge 2012 (ILSVRC2012).[1] Using pixels as input, the network automatically learns useful image representations for the classification task. The results reveal that deep learning significantly outperforms state-of-the-art computer vision representations competitors [29]. However, although this trend is unquestionable for this large-scale context (1 million training examples), the feasibility of reaching state-of-the-art performances in other complex datasets with fewer training examples remains unclear.

Therefore, the Bag-of-words (BoW) model [39] that proved to be the leading strategy in the last decade, remains a very competitive representation. Two main breakthroughs have been reached that explain the BoW model strength. The first one is the design of discriminative low-level local features, such as Scale-Invariant Feature Transform (SIFT) [32] and Histograms of oriented Gradients (HoG) [12]. The second one is the emergence of mid-level representations inspired from the text retrieval community. Indeed, coding local features and aggregating the codes (pooling) make it possible to output a vectorial representation for each image. Subsequently, this representation can be used to train powerful statistical learning models, e.g., Support Vector Machine (SVM) [11], or to model visual attention maps by feature weighting and selection using biologically inspired methods [23, 38, 47].

Extensive studies have been carried out for adapting to images, the initial method [39] inspired by text information retrieval. In particular, many attempts for improving the coding and pooling steps have been done. In this chapter, we first investigate the BoW pipeline in terms of parameter setting and feature combination for classification. We do believe that such an analysis should help to clarify the real difference between mid-level representations for a classification purpose. Based on this study, we also introduce an early fusion [41] method that takes into account and distinguishes low contrast regions from high contrast regions in images. Low contrast regions are usually either completely removed and ignored from the mid-level representation of images, or processed as any common feature. The idea is to

[1] http://www.image-net.org/challenges/LSVRC/2012/

exploit occurrence statistics of low contrast regions and combine them with classical recognition methods applied on high contrast regions. The fusion we propose does not exploit low-level features of different natures (such as combining edge-based, color, metadata descriptors, etc.) but processes low-level features differently with regard to their gradient magnitude. We focus our experiments on the Caltech-101 [18] and Scene-15 [30] datasets, where most of state-of-the-art methods improving over the BoW model have been evaluated.

The remainder of the chapter is organized as follows: Section 2.2 gives an overview of the most significant mid-level BoW improvements, and clarifies the impact of other low-level and mid-level parameters on classification performances. Section 2.3 presents the classification pipeline evaluated in this chapter. In Sect. 2.4, we specifically study the pooling fusion of low and high contrast regions. With local edge-based descriptors (e.g., SIFT), the feature normalization process is likely to produce noisy features: we analyze the use of a thresholding procedure used in VLFEAT [44] to overcome this problem. In addition, we propose novel coding and pooling methods that are well adapted for handling low contrast regions. Section 2.5 provides a systematic evaluation of the impact on classification performances of the different parameters studied in the chapter.

2.2 Bag-of-Words Literature

In the BoW model, converting the set of local descriptors into the final image representation is performed by a succession of two steps: coding and pooling. In the original BoW model, coding consists of hard assigning each local descriptor to the closest visual word, while pooling averages the local descriptor projections. The final BoW vector can thus be regarded as a histogram counting the occurrences of each visual word in the image. Since the notion of 'word' is not as clear for image classification as for text retrieval, many efforts have been recently devoted to improve coding and pooling [4]. It is worth mentioning that extensive works have also been proposed to combine multiple low-level features. For example, Vedaldi et al. [45] and Gehler et al. [20] report that the performances can be significantly improved, using Multiple Kernel Learning (MKL) [2] or LP-boosting, respectively. Combining multiple low-level features is a complementary approach to the improvement of mid-level representation, and we do not give more details on such methods. Since this low-level combination is known to have a large impact on performances (for example in [20] the best reported results are above 82 % while the best performing low-level feature is below 70 %), the remainder of the methods studied by now focus on mono feature methods, where mid-level representations are extracted from a single low-level visual modality.

When dealing with images and visual codebooks, the hard assignment strategy induces an approximation of the local feature. To attenuate this quantization loss, soft-assignment attempts to smoothly distribute features to the codewords [21, 31]. In sparse coding approaches [5, 48, 49], there is an explicit minimization of the

feature reconstruction error, along with a prior regularization that encourages sparse solutions. However, one main drawback in sparse coding is that the code optimization needs to be solved for each descriptor. This makes inference very slow, especially when there are many descriptors or when the dictionary is large. As sparse coding are decoder networks, some approaches propose to learn encoding-decoder networks [22, 28], in which an encoder is concurrently learned to avoid performing the heavy sparse coding minimization. Another way to make coding more accurate is to have a vectorial coding scheme. In aggregate methods, such as Fisher vectors [34, 35], VLAD [26] or super-vectors [51], the difference in the feature space between the local descriptor and each codeword is stored. Despite their very good performances, these aggregate methods cause a huge inflation in the representation size, where a dimensionality of 1 million is common.

Regarding pooling, alternative strategies to averaging the codes have been studied. Max pooling is a promising alternative to sum pooling [5, 7, 31, 48, 49], especially when linear classifiers are used. Other works study the pooling beyond a scalar pooling. In [1, 16], the probability density function (pdf) of the distance between the local features and each codeword is estimated, providing a richer statistics of the codes than using average or max pooling (that output a scalar value). This vectorial pooling strategy is shown to improve performances in various image databases.

Finally, one important limitation of the visual BoW model is the lack of spatial information. The most popular extension to overcome this problem is the Spatial Pyramid Matching Scheme (SPM) [30]. SPM independently pools information from different images regions defined by a multi-resolution spatial grid and concatenates the histograms to form the final image vector. Despite its simplicity and rigidity, SPM generally brings a substantial gain in classification performances in most databases. Using more sophisticated models to incorporate spatial information in the vectorial representation, generally fails to improve performances over SPM. A noticeable exception is [13], where a graph-matching kernel strategy is used to model spatial alignment between images. The performance increases above 80 % which is outstanding for a mono feature method. However, the results were less impressive in a scene database such as Scene-15 [30]. Karaman et al. [27] propose a multi-layer structural approach for the task of object based image retrieval. The structural features are nested multi-layered local graphs built upon sets of SURF [3] feature points with Delaunay triangulation. A BoW framework is applied on these graphs. The multi-layer nature of the descriptors consists in scaling from trivial Delaunay graphs by increasing the number of nodes layer by layer up to graphs with maximal number of nodes. For each layer of graphs, its own visual dictionary is built. Finally, an interesting attempt including both spatial pooling and aggregation in the feature space is the work of Feng et al. [19]. They propose to learn both aspects of the pooling from data using a supervised criterion. Specifically, a per-class ℓ_p geometric pooling is introduced that learns the optimal pooling in between max and average pooling. A spatial weighting is also learned from data to maximize performances. They report the score of 82.6 %, which is, to the best of our knowledge, the state-of-the-art result using mono features in the Caltech-101 [18] database.

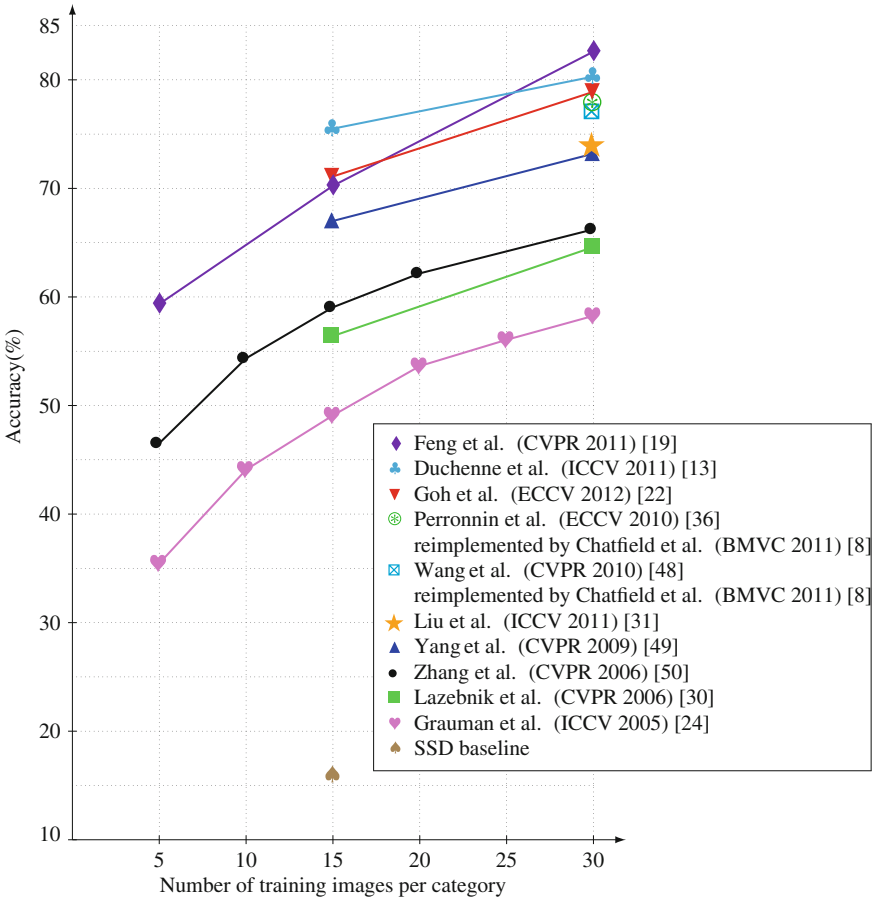

Fig. 2.1 State-of-the-art results since 2006 on the Caltech 101 for BoW pipeline methods in mono feature setup

Figure 2.1 shows the performance evolution of different mid-level representations, in the Caltech-101 database, using mono features. It is obvious that the mid-level steps improvement since 2006 significantly boosted performances: for example, using 30 training examples, there is a substantial gain of about 20 pt from the baseline SPM work of Lazebnik et al. [30] (∼64 % in 2006) to the pooling learning method of Feng et al. [19] (∼83 % in 2011). Nonetheless, the absolute numbers reported in the different publications to illustrate the improvement brought out by some mid-level representation sometimes also include differences in the feature computation or learning algorithms. These variations make the merits of a given mid-level representation confusing. This aspect has recently been studied by Chatfield et al. [8]. The authors re-implemented some of the most powerful mid-level representations (e.g., [35, 48, 51]), and provide an experimental comparison in

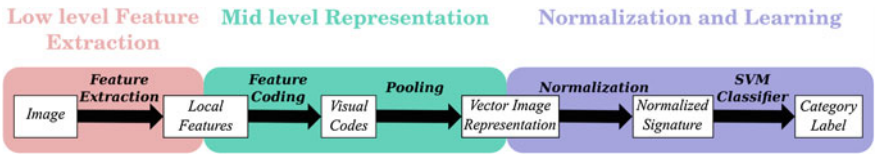

Fig. 2.2 BoW pipeline for classification

the PASCAL VOC [14] dataset. Apart from the fact that some methods are nonreproducible,[2] the main conclusion is that mid-level representation performances are strongly impacted by low-level feature extraction parameters. In particular, it is clear that mid-level representations benefit from a "heavy" low-level feature computation setup, mainly using a dense and multiscale extraction scheme.

In this chapter, we analyze the impact on classification performance of different parameters in the BoW pipeline. We extend here the analysis of Chatfield et al. [8] by including novel parameters, such as normalization of the image signature and a specific coding/pooling strategy for low contrast regions, that strongly impact performances. We focus our experiments on the Caltech-101 [18] and Scene-15 [30] databases, where most state-of-the-art methods improving over the BoW model have been evaluated.

2.3 Classification Pipeline

Figure 2.2 illustrates the whole classification pipeline studied in this chapter. Local features are first extracted in the input image, and encoded into an off-line trained dictionary. The codes are then pooled to generate the image signature. This mid-level representation is ultimately normalized before training the classifier. Each block of the figure is detailed in the following sections. In particular, we specify the main parameters of the BoW pipeline that have a strong impact on classification performances, and evaluate these parameters in Sect. 2.5.

2.3.1 Low-Level Feature Extraction

The first step of the BoW framework corresponds to local feature extraction. To extract local descriptors, one first issue is to detect relevant image regions. Many attempts have been done to achieve that goal, generally based on a saliency criterion:

[2] Chatfield et al. [8] report that their re-implementation of Zhou et al. [51] performs 6 % below the published results. From personal communication with the authors of Zhou et al. [51], the results reported in Chatfield et al. [8] are representative of the method performances, without including non trivial modifications not discussed in the chapter.

Harris detector [25] or its multiscale version [33], SIFT detector [32], etc. However, for classification tasks, most evaluations reveal that a regular grid-based sampling strategy leads to optimal performances [17]. Therefore, we follow this brute-force region selection scheme. In each patch, SIFT features [32] are computed because of their excellent performances attested in various datasets. SIFT features [32] have initially been designed for an image matching purpose. In this context, the ability to match image regions under various geometric and photometric deformations is crucial. For that reason, the SIFT descriptor is made rotationally invariant, by computing the gradient orientation relatively to the dominant orientation of the patch. However, in an image classification context, it is shown that ignoring the rotation invariance favorably impacts performances [43]. This is due to the fact that the orientation of the patch is actually informative for scene and object recognition, and we disable the orientation invariance in our experiments.

In the sampling process, two parameters have a strong impact on classification performances:

- **Sampling density** As we verify experimentally, the denser the sampling is, the better the performances get. The density is set through the spatial stride parameter and corresponds to the distance between the center of two closest extracted patches. In most recent published papers [5, 31, 48, 49], a commonly reported setup is to use a dense monoscale SIFT extraction scheme with a spatial stride set to 8 pixels. However, some authors provide publicly available code using different setups than those reported in their paper.[3]
- **Monoscale versus multiscale features** It is known [8] that using multiscale features increases the amount of low-level information for generating the mid-level signatures, and thus favorably impacts performances. Again, most approaches have been evaluated with a monoscale dense sampling strategy. Wang et al. [48] evaluate their method (LLC) in a multiscale setting, making the comparison with respect to other methods that use monoscale features somehow unfair.

2.3.2 Mid-Level Coding and Pooling Scheme

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_N)$ be the set of local descriptors in an image, where N is the number of local descriptors in the image. In the BoW model, the mid-level signature generation first requires a set of codewords $\left(\mathbf{b}_i \in \mathbb{R}^d\right)_{i=1}^{M}$ (d is the local descriptor's dimensionality, and M is the number of codewords). This set of codewords is called visual codebook or dictionary. Different strategies to compute the codebook exist. The codebook can be performed with a *static* clustering, e.g., Smith

[3] In the provided source codes for evaluation, the sampling is sometimes set to lower values: e.g., 6 pixels in http://www.ifp.illinois.edu/~jyang29/ScSPM.htm for Liu et al. [31] or http://users.cecs.anu.edu.au/~lingqiao/ for Liu et al. [31]. Compared to the value of 8 pixels, the performances decrease of about 1–2 %, making some reported results in published papers over-estimated.

Table 2.1 Coding and pooling strategies. The functions f and g are explicited below

	$\mathbf{x}_1$		$\mathbf{x}_j$		$\mathbf{x}_N$	
$\mathbf{b}_1$	$u_{1,1}$	$\cdots$	$u_{1,j}$	$\cdots$	$u_{1,N}$	
	$\vdots$		$\vdots$		$\vdots$	
$\mathbf{b}_i$	$u_{i,1}$	$\cdots$	$u_{i,j}$	$\cdots$	$u_{i,N}$	$\Rightarrow$ g : *pooling*
	$\vdots$		$\vdots$		$\vdots$	
$\mathbf{b}_M$	$u_{M,1}$	$\cdots$	$u_{M,j}$	$\cdots$	$u_{M,N}$	
			$\Downarrow$			
			f : *coding*			

and Chang [40] use a codebook of 166 regular colors defined a priori. These techniques are generally far from optimal, except in very specific applications. Usually, the codebook is learnt using an unsupervised clustering algorithm applied on local descriptors randomly selected from an image dataset, providing a set of M clusters with centers $\mathbf{b}_i$. K-means is widely used in the BoW pipeline, whereas Gaussian Mixture Models are preferred with Fisher Vectors [35]. Other approaches [5, 22] try to include supervision to improve the dictionary learning. However, Coates and Ng [9] report that dictionary elements learned with "naive" unsupervised methods (k-means or even random sampling) are sufficient to reach high performances on different image datasets. What is reported in [9] is that most of the recognition performance is a function of the choice of architecture, specifically a good encoding function (i.e., sparse or soft) is required. In our experiments, we then choose to perform the codebook with a k-means algorithm. Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_i, \ldots, \mathbf{b}_M)$ denote the resulting visual dictionary, where M is the number of visual codewords (clusters).

In Chatfield et al. [8], several mid-level representations including different coding and pooling methods are evaluated. In this chapter, we focus our re-implementation on one specific method: the Localized Soft Coding (LSC) approach [31]. Indeed, LSC proves to be a very competitive method, reaching very good results in Caltech-101 and Scene-15 databases.[4] Specifically, LSC is shown to be comparable or superior to sparse coding methods (e.g., [5, 48, 49]) while the encoding is significantly faster since no optimization is involved. Moreover, LSC is used with linear classifiers (see Sect. 2.3.3), making the representation adequate for dealing with large-scale problems.

Table 2.1 gives a matrix illustration of the mid-level representation extraction in the BoW pipeline, for scalar coding and pooling schemes. The set of local descriptors $\mathbf{X}$ is represented in columns, while the set of dictionary elements $\mathbf{B}$ occupies the rows.

[4] Note that from personal communication with the authors, we discover that the performances of 74 % in Liu et al. [31] in the Caltech-101 dataset have been obtained with a wrong evaluation metric. The level of performances that can be obtained with the setup depicted in Liu et al. [31] is about 70 % (see Sect. 2.5). However, the conclusion regarding the relative performances of LSC with respect to sparse coding remains valid.

One column of the matrix thus represents the encoding of a given local descriptor $\mathbf{x}_j$ into the codebook that we denote as $f(\mathbf{x}_j)$. In each row, aggregating the codes for a given dictionary elements $\mathbf{b}_i$ results in the pooling operation, denoted as $g(\mathbf{x}_j)$.

In LSC [31], the encoding $u_{i,j}$ of $\mathbf{x}_j$ to $\mathbf{b}_i$ is computed as follows using the k-nearest neighbors $\mathcal{N}_k(\mathbf{x}_j)$:

$$u_{i,j} = \frac{e^{-\beta \hat{d}(\mathbf{b}_i, \mathbf{x}_j)}}{\sum_{l=1}^{M} e^{-\beta \hat{d}(\mathbf{b}_l, \mathbf{x}_j)}} \tag{2.1}$$

where $\hat{d}(\mathbf{b}_i, \mathbf{x}_j)$ is the "localized" distance between $\mathbf{b}_i$ and $\mathbf{x}_j$, i.e., we encode a local descriptor $\mathbf{x}_j$ only on its k-nearest neighbors:

$$\hat{d}(\mathbf{b}_i, \mathbf{x}_j) = \begin{cases} d(\mathbf{b}_i, \mathbf{x}_j) & \text{if } \mathbf{b}_i \in \mathcal{N}_k(\mathbf{x}_j) \\ +\infty & \text{otherwise} \end{cases} \tag{2.2}$$

From the Localized Soft Coding strategy leading to $u_{i,j}$ codes, *max pooling* is used to generate the final image signature $\mathbf{Z} = \{z_i\}_{i \in \{1,\ldots,M\}}$ with:

$$z_i = \max_{j \in \{1,\ldots,N\}} u_{i,j} \tag{2.3}$$

The coding (function f) and pooling (function g) can be represented in this way:

$$f(\mathbf{x}_j, \mathbf{B}) = \mathbf{U}_j = \begin{pmatrix} u_{1,j} \\ \vdots \\ u_{i,j} \\ \vdots \\ u_{M,j} \end{pmatrix} \tag{2.4}$$

$$g(\mathbf{X}, \mathbf{B}) = \mathbf{Z} = \begin{pmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_M \end{pmatrix} \tag{2.5}$$

where z_i is defined in Eq. (2.3).

In addition, spatial information is incorporated using a linear version [49] of the Spatial Pyramid Matching (SPM) Scheme [30]: signatures are computed in a multi-resolution spatial grid with three levels 1×1, 2×2 and 4×4. At the mid-level representation stage, the main parameter impacting accuracy is definitely M, the dictionary size.

2.3.3 Normalization and Learning

Once spatial pyramids are computed, we use linear SVMs to solve the supervised learning problem. The signature normalization is questionable. In Chatfield et al. [8], ℓ_2-normalization is applied, because this processing is claimed to be optimal with linear SVMs [46]. On the other hand, normalizing the data may discard relevant information for the classification task. For that reason, some authors report that ℓ_2-normalization negatively impacts performances, and therefore choose not performing any normalization, as in LSC [31] or in the sparse coding work of Boureau et al. [6]. Therefore, we propose here to experimentally evaluate the impact of the normalization policy on classification performances.

We use for all experiments the ℓ_2-regularized ℓ_1-loss linear SVM classification solver of the LibLinear library [15]. The SVM model can be written:

$$\begin{aligned} &\min_{\mathbf{w},\xi,b} \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i(\mathbf{w}^\top \mathbf{p}_i + b) \geq 1 - \xi_i, \forall i = 1,\ldots,n \\ &\xi_i \geq 0, \forall i = 1,\ldots,n \end{aligned} \tag{2.6}$$

where $(\mathbf{p}_i, y_i)_{i=1}^n$ with $y_i \in \{-1, +1\}$ are training samples, and C is a regularization parameter, which provides a way to avoid overfitting.

The regularization parameter C of the SVM can be determined on a validation set. In our experiments, we simply set it to a default value (10^5) because we did not observe improvement nor decline of accuracy for large values of C.

2.4 Pooling Fusion of Low and High Contrast Regions

Originally, local descriptors like SIFT [32] have been used to describe the visual content around keypoints. The keypoints are generally detected as high saliency image areas, where the contrast in the considered region is large, making the extraction of edge-based descriptors relevant. However, when a dense sampling strategy is used, the feature extraction becomes problematic because edge-based feature extraction is prone to noise in low contrast areas. This drawback is worsened with SIFT descriptors that are ℓ_2-normalized in order to gain robustness to illuminate variations: in the dense sampling setup, this normalization might make (noisy) descriptors be close to descriptors with very large gradient magnitude.

Two different ways to deal with low contrast regions are proposed in different publicly available libraries. A first one[5] renormalizes all the features whose norm is superior to a given threshold γ and divides the other features by γ so that all the

[5] Available on Svetlana Lazebnik's professional homepage: http://www.cs.illinois.edu/homes/slazebni/.

resulting features have a norm between 0 and 1. In this way, relevant patches with low contrast are preserved. Another one[6] also renormalizes all the features whose norm is superior to a given threshold γ but sets the other features to zero. From our experience, the latter method outperforms the former one in multiscale dense sampling strategies. We then choose to focus on the VLFEAT implementation and study the impact of the parameter γ on recognition performance. To the best of our knowledge, the impact of the normalization of low contrast regions on classification performance has never been explored in any prior work.

To better deal with low contrast areas in the BoW classification pipeline, we propose the following improvements: defining visual stop features (Sect. 2.4.1), and specific coding and pooling methods for low contrast regions (Sect. 2.4.2).

2.4.1 Visual Stop Feature: Thresholding Low Contrast Patches

In the context of image retrieval, Sivic and Zisserman [39] define **visual stop words** as the most frequent visual words in images that need to be removed from the feature representation. With the SIFT computation in low contrast patches, we are concerned about a specific type of problematic features that we call **visual stop features** since they arise at the feature extraction step (before the BoW computation). To overcome the problem of noisy SIFT computation, we threshold the descriptor norm magnitude. Let us consider a given SIFT feature $\mathbf{x}$ extracted in some region of an image. We apply the following post-processing to $\mathbf{x}$ so that the output of the feature computation is $\mathbf{x_p}$:

$$\begin{cases} \mathbf{x_p} = \mathbf{0} & \text{if } \|\mathbf{x}\| < \gamma \\ \mathbf{x_p} = \frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{otherwise.} \end{cases} \tag{2.7}$$

As already mentioned, this post-processing for the SIFT computation is performed in some publicly available libraries, e.g., VLFEAT [44]. The idea is to set the descriptors corresponding to low contrast regions to a default value (e.g., $\mathbf{0}$), and not normalizing them in this case. This thresholding is dedicated to filter out the noisy feature computation by assigning a constant value to "roughly" homogeneous regions. The parameter γ defines the threshold up to which a region is considered homogeneous. In a given image $\mathcal{I}$, we denote as $\mathcal{X}_s$ the set of stop features: $\mathcal{X}_s = \{\mathbf{x} \in \mathcal{I} \,/\, \|\mathbf{x}\| < \gamma\}$. We also denote $\mathcal{X}_m$ the set of nonhomogeneous regions: $\mathcal{X}_m = \{\mathbf{x} \in \mathcal{I} \,/\, \|\mathbf{x}\| \geq \gamma\}$. Figures 2.3 and 2.4 illustrate some examples of visual stop features (illustrated with red circles) depending on γ, in Caltech-101 and Scene-15, respectively. We notice that patches with lowest magnitude mostly do not belong to the object to be recognized or do not belong to their discriminative parts (but may contain some relevant contextual information), supporting the relevance of the applied post-processing. We propose in Sect. 2.4.2 a specific modeling, in the BoW framework of stop features.

[6] Example VLFEAT [44] http://www.vlfeat.org/.

Fig. 2.3 Visualization of the visual stop features (*in circles*) depending on the threshold γ applied to the SIFT descriptor norm

2.4.2 Hybrid Image Representation

Figure 2.5 illustrates the proposed method to better deal with low contrast regions in the BoW pipeline. In particular, we adapt the coding and pooling scheme to the case of low contrast regions that are treated separately.

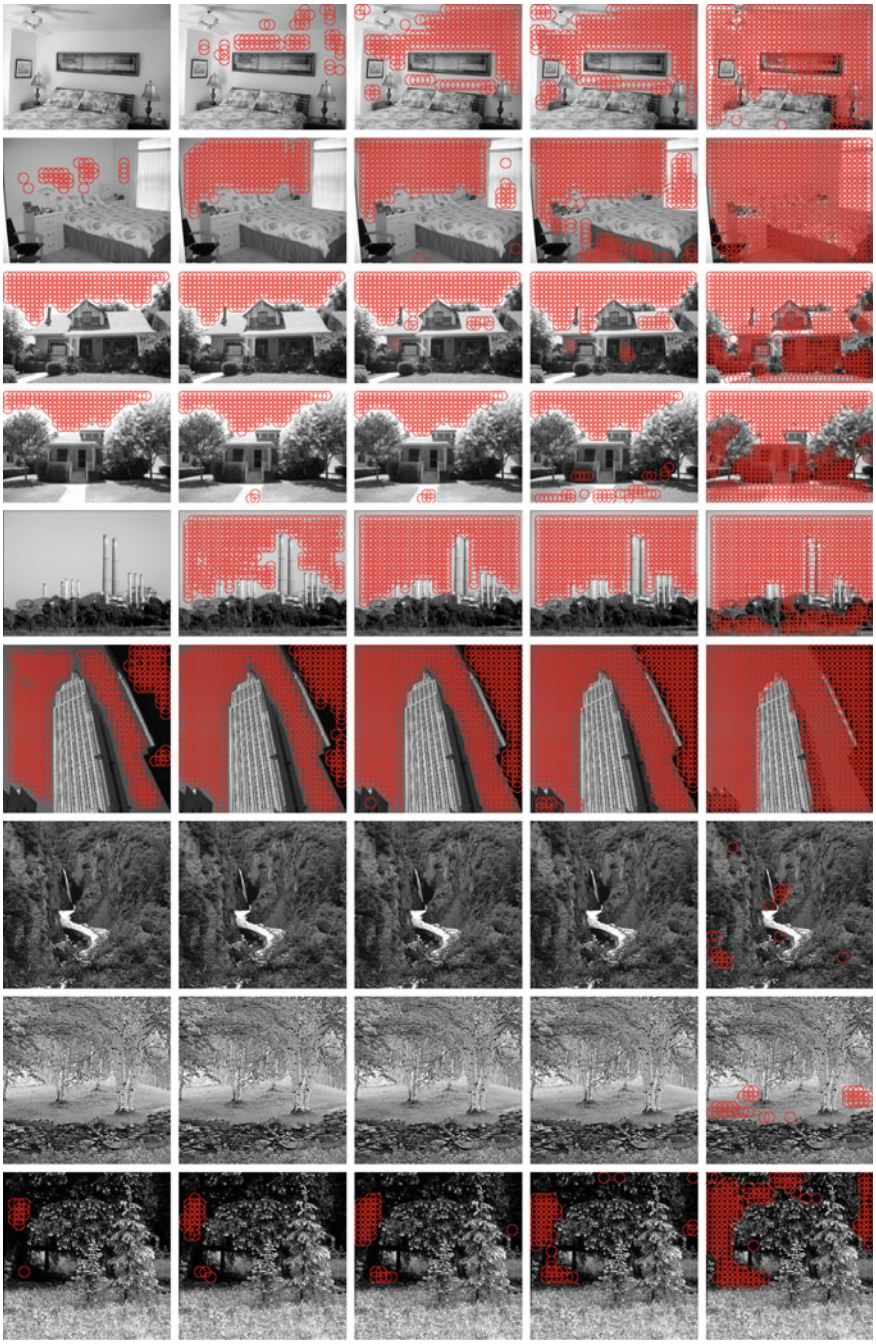

Fig. 2.4 Visualization of the visual stop features (*in circles*) depending on the threshold γ applied to the SIFT descriptor norm

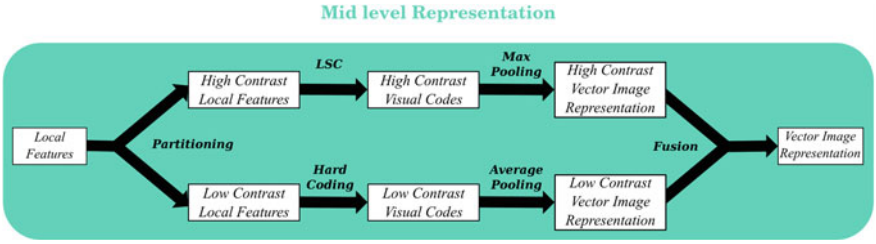

Fig. 2.5 Proposed mid-level representation in the BoW pipeline

2.4.2.1 New Dictionary Training and Feature Coding

We propose to identify a specific word in the dictionary ($\mathbf{b_0}$) to represent homogeneous regions. During codebook training, we learn the $M-1$ remaining codewords, ($\mathbf{b_1}$, ..., $\mathbf{b_{M-1}}$), thus excluding stop features when randomly sampling descriptors in the database. During feature encoding, we propose to hard assign each visual stop feature to the specific word corresponding to homogeneous regions ($\mathbf{b_0}$). This is sensible, since the thresholding consists of ignoring the small magnitude norm information.
For the other features, i.e., $\mathcal{X}_m$, we use the LSC method described in Sect. 2.3.2, encoding each feature on the $M-1$ "nonhomogeneous" codewords elements.

2.4.2.2 Early Fusion: Hybrid Pooling Aggregation

As described in Sect. 2.3.2, max pooling is used with LSC because it achieves better classification performances than average pooling. For visual stop features, however, since hard assignment is performed, the corresponding pooled value z_0 for the word representing homogeneous regions $\mathbf{b_0}$ using max pooling would be binary. Thus, it would only account for the presence/absence of homogeneous regions in the image. Using average pooling instead seems more appropriate: the pooled value then incorporates a statistic estimation of the ratio of low contrast regions in the image that is much more informative than the binary presence/absence value. We thus follow a hybrid pooling strategy, using average pooling for $\mathcal{X}_s$ and max pooling for $\mathcal{X}_m$. Both representations are then concatenated into a global descriptor before normalization and learning. This early fusion scheme is applied in each bin of the SPM pyramid independently.

Our hybrid pooling BoW pipeline has the following advantages: (1) The codebook can be learned only for features of $\mathcal{X}_m$, resulting in a richer representation of $\mathcal{F}_m$ for the same number of training samples; (2) The hard assignment to $\mathbf{b_0}$ for $\mathcal{X}_s$ is relevant since each homogeneous region should not be encoded in the "nonhomogeneous" codewords; (3) The encoding of $\mathcal{X}_s$ is substantially faster than using the standard LSC method, since the automatic assignment avoids the (approximate) nearest

neighbor search that dominates the computational time; (4) The average pooling strategy applied to the homogeneous codeword $\mathbf{b_0}$ incorporates a richer information about the ratio of homogeneous regions in the image. This feature that must vary among different classes, can therefore be capitalized on when training the classifier.

Figure 2.6 illustrates the computation of our BoW representation for a given input image.

2.5 Experiments

Before evaluating our hybrid method, we first report an exhaustive quality assessment of the BoW strategy.

2.5.1 Datasets and Experimental Setup

Experiments are proposed on two widely used datasets: Caltech-101 [18] and Scene-15 [30]. Caltech-101 is a dataset of 9,144 images containing 101 object classes and a background class. Scene-15 contains 4,485 images of 15 scene categories.

A fixed number of images per category (30 for Caltech-101 and 100 for Scene-15) is selected to train models and all the remaining images are used for test. The reported accuracy is measured as the average classification accuracy across all classes over 100 splits. For each class, the accuracy is measured as the percentage of images of the class that are correctly assigned to the class by the learned classifier. All the images are resized to have a maximum between width and height set to 300 pixels.

Like Chatfield et al. [8], we only extract SIFT descriptors. We use a spatial stride of between 3 and 8 pixels (corresponding to the sampling density), and at 4 scales for the multiscale, defined by setting the width of the SIFT spatial bins to 4, 6, 8 and 10 pixels respectively. The default spatial stride is 3 pixels. When referring to monoscale, we set the width of the spatial bins to 4 pixels, with a default spatial stride of 8 pixels. SIFT descriptors are computed with the vl_phow command included in the VLFEAT toolbox [44], version 0.9.14, for the following experiments (Sect. 2.5.2). Apart from the stride and scale parameters, the default options are used. In Sect. 2.5.4, monoscale patches are extracted with the default vl_dsift command designed for monoscale extraction.

For LSC implementation, Liu et al. [31] use $\beta = 1/(2\sigma^2) = 10$ (Eq. 2.1) with normalized features. Since the norms of VLFEAT features are equal to 512 (instead of 1 as the descriptors used in [31]), we set $\sigma \simeq 115$ and the number of nearest neighbors $k = 10$ (Eq. 2.1) to be consistent with [31].

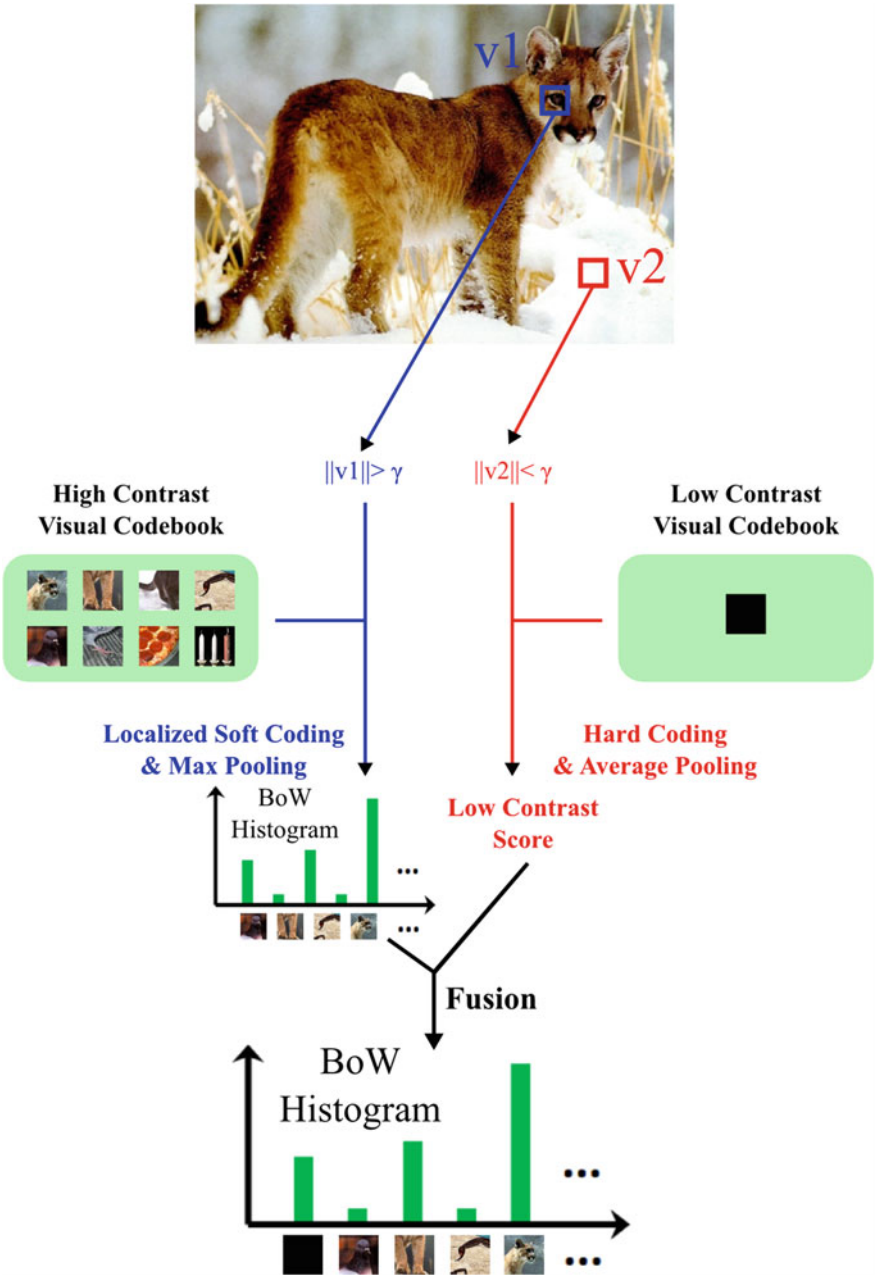

Fig. 2.6 Pooling fusion applied on an image: local descriptors are separated in two different sets depending on their norm. High contrast and low contrast regions are processed in two different coding and pooling schemes. Their resulting BoWs are concatenated in a single BoW

Table 2.2 Classification results on Caltech-101 dataset with 30 training images per class

Spatial stride	Scaling	Codebook size	Accuracy (no norm)	Accuracy (ℓ_2-norm)
8	Monoscale	800	70.07 ± 0.96	70.46 ± 1.04
6	Monoscale	800	71.64 ± 0.99	72.01 ± 0.96
3	Monoscale	800	72.45 ± 1.05	72.73 ± 0.99
8	Monoscale	1700	71.67 ± 0.93	71.95 ± 0.90
8	Monoscale	3300	72.13 ± 0.99	72.50 ± 0.97
8	Multiscale	800	73.35 ± 0.89	73.83 ± 0.96
8	Multiscale	1700	75.34 ± 0.92	75.97 ± 0.86
8	Multiscale	3300	76.91 ± 0.98	77.02 ± 0.94
3	Multiscale	800	73.81 ± 0.95	73.99 ± 0.86
3	Multiscale	1700	75.72 ± 1.13	76.00 ± 0.94
3	Multiscale	3300	77.23 ± 1.02	77.47 ± 0.99
3	Multiscale	6500	78.00 ± 1.05	78.46 ± 0.95

2.5.2 Bag-of-Words Pipeline Evaluation

We study in Table 2.2 the results of the BoW pipeline using the LSC coding method for Caltech-101 dataset. The main parameters studied are the codebook size, the spatial stride, the mono/multiscale strategy, and the normalization.

We selected the most important combinations between all the possibilities. First, one can notice that multiscale is always above monoscale results. In monoscale setup, we do not investigate too many combinations. The best results are 72.73 % for a small spatial stride with normalization. The codebook size of 3,300 also gives good results. Compared to the classical performance of 64 % of the BoW SPM [30], it is remarkable to see how a careful parametrization including normalization of a BoW soft pipeline may boost the performances up to 9 %.

These trends are fully confirmed in the multiscale setting. The best score of 78.46 % is obtained with a small spatial stride of 3, multiscale, and a dictionary of size 6,500 with ℓ_2-normalization. The soft BoW pipeline outperforms the advanced methods presented in Chatfield et al. [8], the Fisher Kernel method (reported at 77.78 %), and the LLC (reported at 76.95 %) with the same multiscale setup and a codebook of 8,000 words (for LLC). It is also above the score of Boureau [6], where the best result reported using sparse coding is 77.3 %. They use a very high dimensional image representation and a costly sparse coding optimization, with a monoscale scheme but a two-step aggregating SIFT features.

Table 2.3 reports the experimental results on Scene-15. They are all consistent with the experiments on Caltech-101. The best result of 83.44 % is also obtained for a multiscale scheme, a small spatial stride of 3, and a large dictionary of size 6,800 with normalization. This score is still slightly better than the Boureau one of 83.3 % [6], but remains below state-of-the-art results for that database.

These experiments confirm that the parameters mentioned in Sect. 2.3 may significantly improve the recognition. A small spatial stride with multiscale, a large

Table 2.3 Classification results on Scene-15 dataset with 100 training images per class

Spatial stride	Scaling	Codebook size	Accuracy (no norm)	Accuracy (ℓ_2-norm)
8	Monoscale	1000	78.72 ± 0.62	78.96 ± 0.60
6	Mnoscale	1000	79.53 ± 0.65	79.74 ± 0.65
3	Monoscale	1000	79.74 ± 0.61	80.05 ± 0.67
8	Monoscale	1700	79.98 ± 0.61	80.29 ± 0.58
8	Monoscale	3400	80.61 ± 0.61	81.16 ± 0.57
8	Multiscale	1000	79.59 ± 0.63	80.12 ± 0.56
8	Multiscale	1700	80.91 ± 0.56	81.25 ± 0.54
8	Multiscale	3400	82.01 ± 0.72	82.39 ± 0.60
3	Multiscale	1000	79.74 ± 0.60	80.14 ± 0.59
3	Multiscale	1700	81.03 ± 0.65	81.23 ± 0.60
3	Multiscale	3400	82.17 ± 0.73	82.42 ± 0.59
3	Multiscale	6800	82.66 ± 0.62	83.44 ± 0.55

codebook and a proper normalization of the spatial pyramid is the winning cocktail for the BoW pipeline. However, the accuracy improvement is more impressive for Caltech-101 (reaching very high performances) than for Scene-15.

2.5.3 Distribution of Gradient Magnitudes

A distribution in Caltech-101 and Scene-15 of the gradient magnitudes of patches in a monoscale setup is illustrated in Fig. 2.7. In Caltech-101, about 6 % of patches have a gradient magnitude smaller than 10^{-4} and 40 % of patches have a feature norm greater than 0.05. In Scene-15, less than 1 % of patches have a magnitude close to 0. This difference compared to Caltech-101 comes from the fact that almost no fully homogeneous region exists in Scene-15 (whereas some images in Caltech-101 contain uniform background).

We study in Sect. 2.5.4 the impact of the parameter γ (in Eq. 2.7) on classification performance with our proposed strategy.

2.5.4 Evaluation of Our Strategy

We evaluate here the classification performances of our early fusion detailed in Sect. 2.4. First, we study the impact of γ (Eq. 2.7). Figure 2.8a shows the evolution of the classification performances depending on γ on Caltech-101 database, in both monoscale and multiscale settings. The results are largely impacted when γ varies: the performances can be improved up to 3 % for the monoscale setup using $\gamma \simeq 10^{-2}$ compared to the default value. The same trend appears for the multiscale setting.

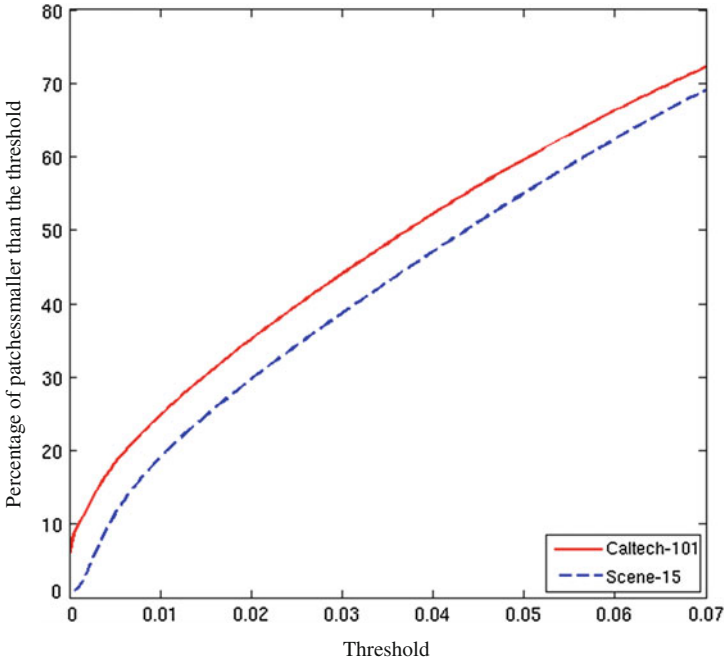

Fig. 2.7 Percentage of patches with gradient magnitude smaller than a given threshold in a monoscale setup

For Scene-15 dataset (Fig. 2.8b), the conclusion differs: in a multiscale setting the performances can be slightly improved, whereas the best result is obtained for $\gamma = 0$ with monoscale features. This may be explained by the fact that in object recognition (particularly on Caltech-101), the patches with lowest magnitude usually do not describe the object to be recognized and belong to the background (see Fig. 2.3).

Second, we evaluate the specific encoding and pooling method for low contrast regions described in Sect. 2.4.2. We provide two gradual evaluations (see Fig. 2.9). The proposed changes improve performances in Caltech-101 database, in both monoscale (Fig. 2.9a) and multiscale settings (Fig. 2.9b). For the multiscale setup, the performances are in addition more robust to γ variations. For the monoscale setup, the average pooling outperforms the max pooling method, validating the idea that enriching the homogeneous regions pooling with a nonbinary value can favorably impact performances. This is not the case in the multiscale experiments, probably because fewer homogeneous regions are extracted in such a setup (due to the increase of the region size), making the statistical estimate of the homogeneous regions ratio less reliable.

Finally, if we use the best setting of parameters with a codebook of 10^4 words, we obtain the score of $\mathbf{79.07 \pm 0.83}\,\%$ on Caltech-101 dataset and $\mathbf{83.83 \pm 0.59}\,\%$ on Scene-15 with our fusion scheme over low/high contrast regions. The reported

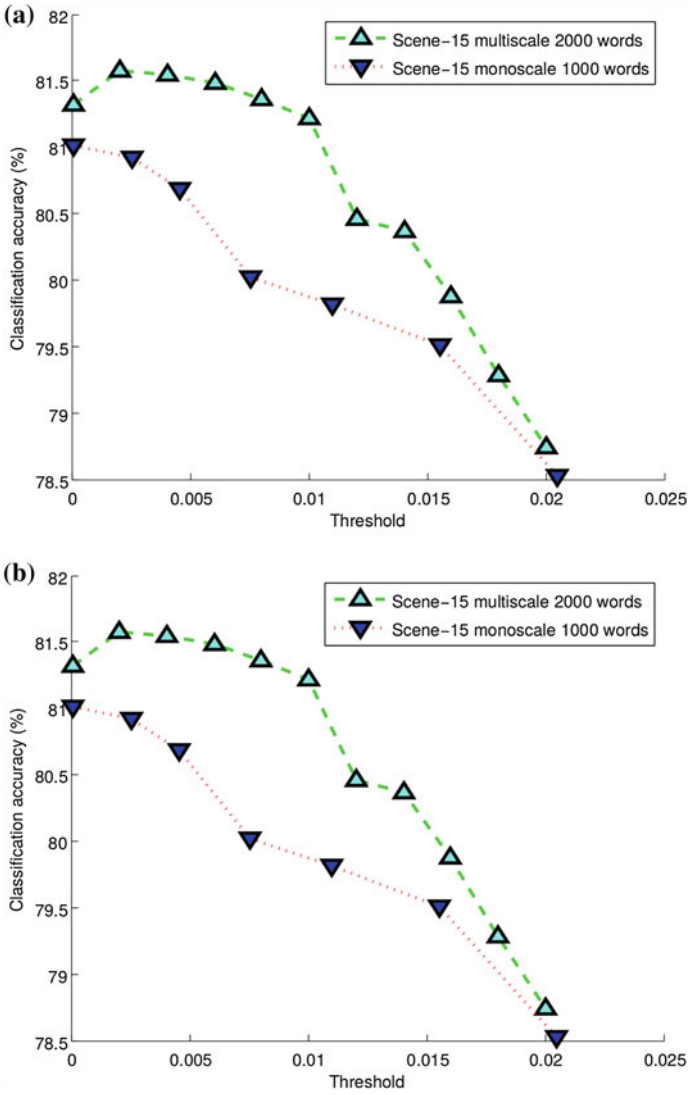

Fig. 2.8 Accuracy of the normalized LSC model as the threshold under which features are set to 0 varies **a** on Caltech-101, **b** on Scene-15

result on the Caltech-101 benchmark is comparable to the best published score [22] following the standard BoW pipeline (scalar coding and pooling + SPM), for a single descriptor type and linear classification. Note that the results obtained by Duchenne et al. [13] or Feng et al. [19] are obtained with methodological tools (resp. graph matching and pooling learning) that are complementary to our method. Therefore, a combination is expected to further boost performances.

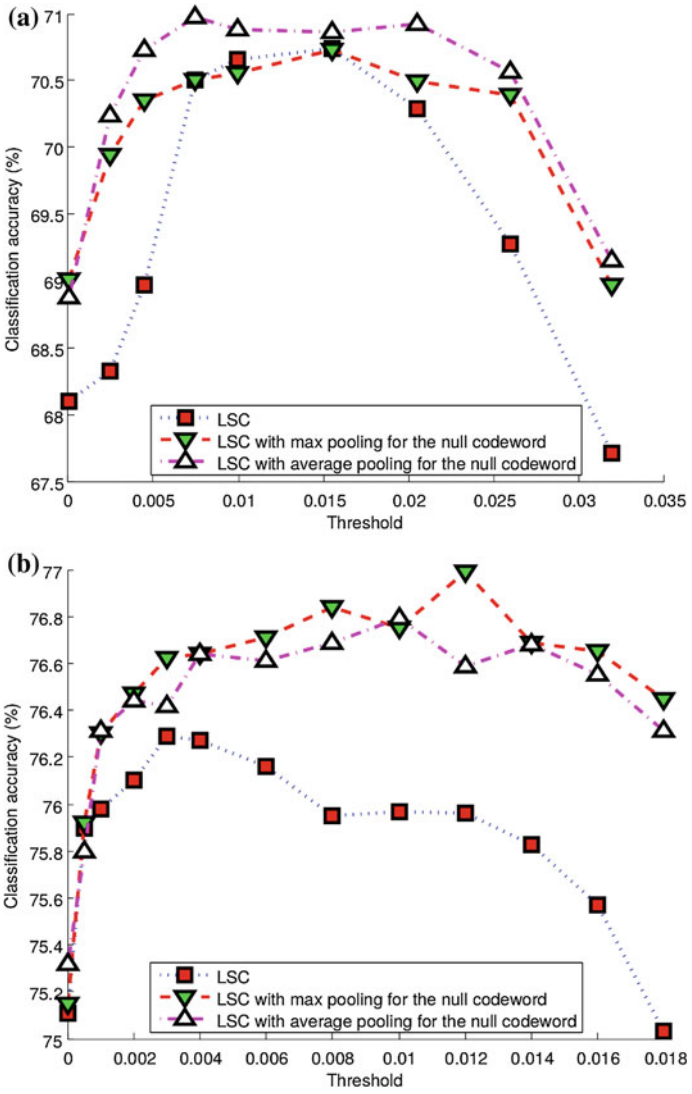

Fig. 2.9 Accuracy of the normalized LSC strategies on Caltech-101 **a** monoscale setup with a codebook of 1000 words, **b** multiscale setup with a codebook of 2000 words

2.6 Conclusions

In this chapter, we have studied in detail the different components of the BoW model in the context of image classification. Particularly, we have shown that several low (sampling rate, multiscale) and mid-level (codebook size, normalization) parameters have an impact on recognition. The codebook size and mono/multiscaling are

definitely the most significant parameters as they allow to describe images with richer or denser information. The sampling rate is more significant in monoscale setup as it allows to increase the number of descriptors to represent images; this number is small in monoscale setup.

We have also investigated some early fusion methods that process low and high contrast regions separately. We have proposed a novel scheme to efficiently embed low contrast information into the BoW pipeline. This scheme is more robust than classic methods to the choice of threshold under which SIFT descriptor are normalized. This results from the fact that meaningful high contrast regions are not mixed with noisy low contrast regions. Finally, our strategy obtains state-of-the-art performances on Caltech-101 and very good results on Scene-15 dataset.

References

1. Avila S, Thome N, Cord M, Valle E, de Araujo A (2011) Bossa: extended bow formalism for image classification. In: Proceedings of the IEEE international conference on image processing (ICIP)
2. Bach FR, Lanckriet GR, Jordan MI (2004) Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the twenty-first international conference on machine learning (ICML)
3. Bay H, Ess A, Tuytelaars T, van Gool L (2008) SURF: speeded Up robust features. Comput Vis Image Underst (CVIU) 110(3):346–359
4. Benois-Pineau J, Bugeau A, Karaman S, Mégret R (2012) Spatial and multi-resolution context in visual indexing. In: Visual Indexing and Retrieval, pp 41–63
5. Boureau Y-L, Bach, F, LeCun Y, Ponce J (2010) Learning mid-level features for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
6. Boureau Y-L, Le Roux N, Bach F, Ponce J, LeCun Y (2011) Ask the locals: multi-way local pooling for image recognition. In: Proceedings of the IEEE international conference on computer vision (ICCV)
7. Boureau Y-L, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in vision algorithms. In: Proceedings of the international conference on machine learning (ICML)
8. Chatfield K, Lempitsky V, Vedaldi A, Zisserman A (2011) The devil is in the details: an evaluation of recent feature encoding methods. In: Proceedings of the British machine vision conference (BMVC)
9. Coates A, Ng A (2011) The importance of encoding versus training with sparse coding and vector quantization. In: Proceedings of the 28th international conference on machine learning (ICML)
10. Cord M, Cunningham P (2008) Machine learning techniques for multimedia: case studies on organization and retrieval. Machine learning techniques for multimedia, cognitive technologies. Springer, Heidelberg
11. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
12. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
13. Duchenne O, Joulin A, Ponce J (2011) A graph-matching kernel for object categorization. In: Proceedings of the IEEE international conference on computer vision (ICCV)
14. Everingham M, Zisserman A, Williams C, Van Gool L (2007) The PASCAL visual obiect classes challenge 2007 (VOC2007) results. Technical Report, Pascal Challenge
15. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: a library for large linear classification. J Mach Learn Res (JMLR) 9:1871–1874

16. de Avila Fontes SE, Thome N, Cord M, Valle E, de Albuquerque Arajo A (2013) Pooling in image representation: The visual codeword point of view. Comp Vis Image Underst 117(5):453–465
17. Fei-fei L (2005) A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
18. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) workshop on GMBV
19. Feng J, Ni B, Tian Q, Yan S (2011) Geometric ℓ_p-norm feature pooling for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
20. Gehler P, Nowozin S (2009) On feature combination for multiclass object classification. In: Proceedings of the IEEE international conference on computer vision (ICCV)
21. van Gemert J, Veenman C, Smeulders A, Geusebroek JM (2010) Visual word ambiguity. IEEE Trans Pattern Anal Mach Intell (TPAMI) 32(7):1271–1283
22. Goh H, Thome N, Cord M, Lim J-H (2012) Unsupervised and supervised visual codes with restricted Boltzmann machines. In: Proceedings of the European conference on computer vision (ECCV)
23. González-Díaz I, Buso V, Benois-Pineau J, Bourmaud G, Megret R (2013) Modeling instrumental activities of daily livinf in egocentric vision as sequences of active objects and context for Alzheimer disease research. In: ACM multimedia workshop on multimedia information indexing and retrieval for healthcare
24. Grauman K, Darrell T (2005) The pyramid match kernel: discriminative classification with sets of image features. In: Proceedings of the IEEE international conference on computer vision (ICCV)
25. Harris S, Stephens M (1988) A combined corner and edge detector. In: Proceedings of the 4th Alvey vision conference, pp 147–151
26. Jégou H, Douze M, Schmid C, Pérez P (2010) Aggregating local descriptors into a compact image representation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
27. Karaman S, Benois-Pineau J, Mgret R, Bugeau A (2012) Multi-layer local graph words for object recognition. In: Proceedings of the international conference on multimedia modeling
28. Kavukcuoglu K, Sermanet P, Boureau Y-L, Gregor K, Mathieu M, LeCun Y (2010) Learning convolutional feature hierachies for visual recognition. In: Proceedings of advances in neural information processing systems (NIPS), pp 1090–1098
29. Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of advances in neural information processing systems (NIPS), pp. 1106–1114
30. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
31. Liu L, Wang L, Liu X (2011) In defense of soft-assignment coding. In: Proceedings of the IEEE international conference on computer vision (ICCV)
32. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis (IJCV) 60:91–110
33. Mikolajczyk K, Schmid C (2004) Scale and affine invariant interest point detectors. Int J Comput Vis (IJCV) 60(1):63–86
34. Mironica I, Uijlings J, Rostamzadeh N, Ionescu B, Sebe N (2013) Time matters! capturing variation in time in video using fisher kernels. In: Proceedings of the 21st ACM international conference on multimedia
35. Perronnin F, Dance CR (2007) Fisher kernels on visual vocabularies for image categorization. In: Proceedings of the IEEE Conference on computer vision and pattern recognition (CVPR)
36. Perronnin F, Sánchez J, Mensink T (2010) Improving the fisher kernel for large-scale image classification. In: Proceedings of the European conference on computer vision (ECCV)

37. Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T (2007) Robust object recognition with cortex-like mechanisms. IEEE Trans Pattern Anal Mach Intell (TPAMI) 29:411–426
38. Sharma G, Jurie F, Schmid C (2012) Discriminative spatial saliency for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
39. Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. In: Proceedings of the IEEE international conference on computer vision (ICCV)
40. Smith JR, Chang S-F (1997) VisualSEEk: a fully automated content-based image query system. In: Proceedings of the fourth ACM international conference on Multimedia, ACM, pp 87–98
41. Snoek C, Worring M, Hauptmann A (2006) Learning rich semantics from news video archives by style analysis. ACM Transa Multimedia Comput Commun Appl (TOMCCAP) 2(2):91–108
42. Thériault C, Thome N, Cord M (2013) Extended coding and pooling in the HMAX model. IEEE Trans Image Process 22(2):764–777
43. van de Sande KEA, Gevers T, Snoek CGM (2010) Evaluating color descriptors for object and scene recognition. IEEE Trans Pattern Anal Mach Intell (TPAMI) 32(9):1582–1596
44. Vedaldi A, Fulkerson B (2008) VLFeat: an open and portable library of computer vision algorithms. http://www.vlfeat.org/
45. Vedaldi A, Gulshan V, Varma M, Zisserman A (2009) Multiple kernels for object detection. In: Proceedings of the IEEE international conference on computer vision (ICCV)
46. Vedaldi A, Zisserman A (2011) Efficient additive kernels via explicit feature maps. IEEE Trans Pattern Anal Mach Intell (TPAMI) 34:480–492
47. Vig E, Dorr, M, Cox DD (2012) Space-variant descriptor sampling for action recognition based on saliency and eye movements. In: Proceedings of the European conference on computer vision (ECCV)
48. Wang J, Yang J, Yu K, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
49. Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
50. Zhang H, Berg AC, Maire M, Malik J (2006) SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
51. Zhou X, Yu K, Zhang T, Huang TS (2010) Image classification using super-vector coding of local image descriptors. In: Proceedings of the european conference on computer vision (ECCV)

http://www.springer.com/978-3-319-05695-1

Fusion in Computer Vision
Understanding Complex Visual Content
Ionescu, B.; Benois-Pineau, J.; Piatrik, T.; Quénot, G. (Eds.)
2014, XIV, 272 p. 74 illus., 65 illus. in color., Hardcover
ISBN: 978-3-319-05695-1