Hybrid Pooling Fusion in the BoW Pipeline

Marc Law, Nicolas Thome, and Matthieu Cord

LIP6, UPMC - Sorbonne University, Paris, France
{Marc.Law,Nicolas.Thome,Matthieu.Cord}@lip6.fr

Abstract. In the context of object and scene recognition, state-of-the-art performances are obtained with Bag of Words (BoW) models of mid-level representations computed from dense sampled local descriptors (*e.g.* SIFT). Several methods to combine low-level features and to set mid-level parameters have been evaluated recently for image classification.
In this paper, we further investigate the impact of the main parameters in 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 exploiting the specificities of edge-based descriptors. Low and high-contrast regions are pooled separately and combined to provide a powerful representation of images. Sucessful experiments are provided on the Caltech-101 and Scene-15 datasets.

1 Introduction & Related Work

Image classification is one of the most challenging problems in computer vision. Indeed, the prediction of complex semantic categories, such as scenes or objects, from the pixel level, is still a very hard task. Two main breakthroughs have been reached in the last decade to achieve this goal. The first one is the design of discriminative low-level local features, such as SIFT [1]. The second one is the emergence of mid-level representations inspired from the text retrieval community, based on the Bag of Words (BoW) model [2].

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 in hard assigning each local descriptor to the closest visual word, while pooling averages the local descriptor projections. 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 Scheme [3]. In addition, many efforts have been recently devoted to improve coding and pooling [4]. To attenuate the quantization loss, soft assignment attempts to smoothly distribute features to the codewords [5, 6]. In sparse coding approaches [7–9], there is an explicit minimization of the feature reconstruction error, along with a regularization prior that encourages sparse solutions. Different pooling strategies have also been studied. Max pooling is a promising alternative to sum pooling [6–10], especially when linear classifiers are used. Therefore, the combination of sparse coding, spatial pyramids and max-pooling is often regarded as the strategy leading to state-of-the-art performances.

In this paper, 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 clarify the real difference between mid-level representations for a classification purpose. Based on this study, we also introduce an early fusion [11] 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, either processed as any common feature. The idea is to 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...) but processes low-level features differently with regard to their gradient magnitude. We focus our experiments on the Caltech-101 [12] and Scene-15 [3] datasets, where most of state-of-the art methods improving over the BoW model have been evaluated. The remainder of the paper decomposes as follows. Section 2 presents the classification pipeline evaluated in the paper. In section 3, 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 [13] to overcome this problem. In addition, we propose novel coding and pooling methods that are well adapted for handling low-constrast regions. Section 4 provides a systematic evaluation of the impact on classification performances of the different parameters studied in the paper.

2 Classification Pipeline

Fig. 1 illustrates the whole classification pipeline studied in this paper. 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.

2.1 Low-level Feature Extraction

The first step of the BoW framework is the feature extraction. We follow a regular grid-based sampling strategy, that proves to be superior to other sparse or random samplings for classification tasks [14]. SIFT features are computed because

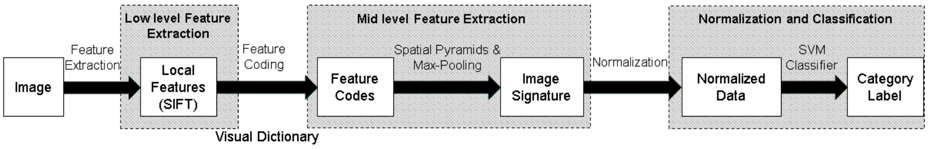


Fig. 1. BoW pipeline for classification

of their excellent performances attested in various datasets. In the sampling process, two parameters have a strong impact on classification performances:
– **Sampling density**. The denser the sampling is, the better the performances get. The density is set through the spatial stride parameter. In published papers [6–9], the stride is usually set to 8 pixels [1].
– **Monoscale versus multiscale features**. It is known [15] that using multiscale features increases the amount of low-level information for generating the mid-levels signatures, and thus favorably impacts performances. Wang *et al.* [8] evaluate their method (LLC) in a multiscale setting, making the comparison with respect to other methods that use monoscale features somehow unfair.

2.2 Mid-level Coding and Pooling Scheme

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N)$ be the set of local descriptors in an image. In the BoW model, the mid-level signature generation first requires a set of codewords $\mathbf{b}_i \in \mathbb{R}^d$ (d is the local descriptor's dimensionality). Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_j, \ldots, \mathbf{b}_M)$ denote the resulting visual dictionary. Usually, $\mathbf{B}$ is learned using an unsupervised clustering algorithm applied on local descriptors randomly selected from an image dataset, providing a set of M clusters with centers $\mathbf{b}_j$.

In [15], several mid-level representations including different coding and pooling methods are evaluated. In this paper, we focus our re-implementation on one specific method: the Localized Soft Coding (LSC) approach [6]. Indeed, LSC proves to be a very competitive method, reaching very good results in Caltech-101 and Scene-15 databases[2]. Specifically, LSC is shown to be comparable or superior to sparse coding methods, *e.g.* [7–9], while the encoding is significantly faster since no optimization is involved. Note that LSC is used with linear classifiers (see section 2.3), making the representation adequate for dealing with large-scale problems. In LSC [6], the encoding $u_{i,j}$ of $\mathbf{x}_i$ to $\mathbf{b}_j$ is computed as follows using the k-nearest neighbors $\mathcal{N}_k(\mathbf{x}_i)$:

$$u_{i,j} = \frac{e^{-\beta \hat{d}(\mathbf{x}_i, \mathbf{b}_j)}}{\sum\limits_{l=1}^{M} e^{-\beta \hat{d}(\mathbf{x}_i, \mathbf{b}_l)}} \qquad \hat{d}(\mathbf{x}_i, \mathbf{b}_j) = \begin{cases} d(\mathbf{x}_i, \mathbf{b}_j) & \text{if } \mathbf{b}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

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

[1] However, in the provided source codes for evaluation, the sampling is often set to lower values (*e.g.* 6 pixels)(http://www.ifp.illinois.edu/∼jyang29/ScSPM.htm or http://users.cecs.anu.edu.au/∼lingqiao/). Compared to the value of 8 pixels, the performances decrease of about $1 \sim 2\%$, making some reported results in published papers over-estimated.

[2] Note that from personal communication with the authors, we discover that the performances of 74% in [6] 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 [6] is about 70% (see section 4). However, the conclusion regarding the relative performances of LSC with respect to sparse coding remains valid.

strategy leading to $u_{i,j}$ codes, *max pooling* is used to generate the final image signature $\mathbf{Z} = \{z_j\}_{j \in \{1;M\}}$ and $z_j = \underset{i \in \{1;N\}}{max} u_{i,j}$. In addition, spatial information is incorporated using a linear version [7] of the Spatial Pyramid Matching (SPM) Scheme [3]: 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 definitively M, the dictionary size.

2.3 Normalization and Learning

Once spatial pyramids are computed, we use linear SVMs to solve the supervised learning problem. The signature normalization is questionable. In [15], ℓ_2-normalization is applied, because this processing is claimed to be optimal with linear SVMs [16]. 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 [6] or in the sparse coding work of [17].

We use for all experiments the ℓ_2-regularized ℓ_1-loss support vector classification solver of the LibLinear library [18]. The C parameter of the SVM can be determined on a validation set, we set it to 10^5 because we did not observe improvement nor decline of accuracy for large values of C.

3 Pooling Fusion of Low and High Contrast Regions

Originally, local descriptors like SIFT [1] 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 egde-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 worsen with SIFT descriptors that are ℓ_2-normalized in order to gain robustness to illumination variations: in the dense sampling setup, this normalization might make (noisy) descriptors be close to descriptors with very large gradient magnitude.

To better deal with low-contrast areas in the BoW classification pipeline, we propose the following improvements: defining visual stop features (section 3.1), and specific coding and pooling methods for low-contrast regions (section 3.2).

3.1 Visual Stop Feature: Thresholding Low Contrast Patches

In the context of image retrieval, Sivic and Zisserman [2] 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,

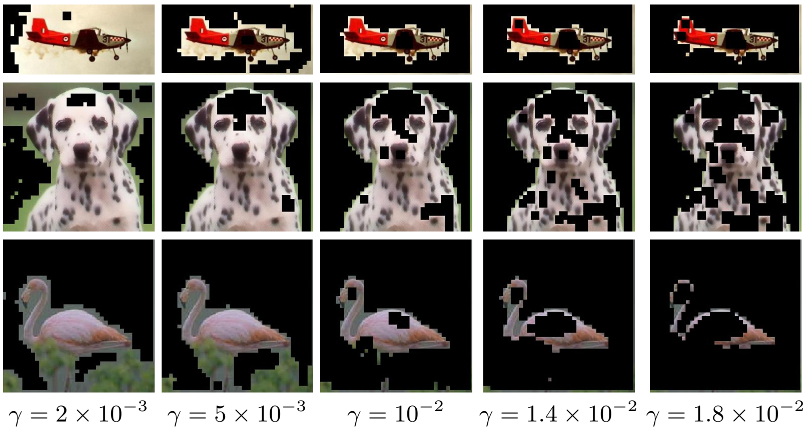

$\gamma = 2 \times 10^{-3}$
$\gamma = 5 \times 10^{-3}$
$\gamma = 10^{-2}$
$\gamma = 1.4 \times 10^{-2}$
$\gamma = 1.8 \times 10^{-2}$

Fig. 2. Visualization of the visual stop features (regions filled in black) depending on the threshold γ applied to the SIFT descriptor norm

we propose to 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 $\mathbf{x_p}$:

$$\mathbf{x_p} = 0 \text{ if } ||\mathbf{x}|| < \gamma \quad \text{and } \mathbf{x_p} = \mathbf{x}/||\mathbf{x}|| \text{ otherwise} \tag{2}$$

This post-processing for the SIFT computation is performed in some publicly available libraries, *e.g.* VLFEAT [13]. The idea is to set the descriptors corresponding to low contrast regions to a default value (*e.g.* 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: $\mathcal{X}_s = \{\mathbf{x} \in \mathcal{I} \ / \ ||\mathbf{x}|| < \gamma\}$. Contrarily, the set of non-homogeneous regions $\mathcal{X}_m$ fulfills: $\mathcal{X}_m = \{\mathbf{x} \in \mathcal{I} \ / \ ||\mathbf{x}|| \geq \gamma\}$. Fig. 2 illustrates some examples of visual stop features (filled in black) depending on γ, in Caltech-101. We notice that patches with lowest magnitude mostly do not belong to the object to be recognized, supporting the relevance of the applied post-processing. We propose in the next subsection a specific modeling, in the BoW framework, of stop features.

3.2 Hybrid Image Representation

New Dictionary Training & Feature Coding. First, 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. Second, during feature encoding, we propose to hard assign each visual stop feature to the specific word corresponding to homogeneous regions ($\mathbf{b_0}$).

For the other features, *i.e.* $\mathcal{X}_m$, we use the LSC method described in section 2.2, encoding each feature on the $M-1$ "non-homogeneous" codewords elements.
Early Fusion: Hybrid Pooling Aggregation. As described in section 2.2, 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 an 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 indepently.

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 the each homogeneous region should not be encoded in the "non-homogeneous" 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.

4 Experiments

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

4.1 Datasets & Experimental Setup

Experiments are proposed on two widely used datasets: Caltech-101 [12] and Scene-15 [3]. Caltech-101 is a dataset of 9144 images containing 101 object classes and a background class. Scene-15 contains 4485 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 performance is measured as the average classification accuracy across all classes over 100 splits. All the images are resized to have a maximum between width and height set to 300 pixels.

Like Chatfield *et al.* [15], 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 [13], version 0.9.14, for the following experiments (Subsection 4.2). Apart from the stride and scale parameters, the default options are used. In Subsection 4.3, monoscale patches are extracted with the default vl_dsift command designed for monoscale extraction.

For LSC implementation, Liu *et al.* [6] use $\beta = 1/(2\sigma^2) = 10$ (Eq 1) with normalized features. Since VLFEAT feature norms are 512, we set $\sigma \simeq 115$ and the number of nearest neighbors $k = 10$ (Eq 1) to be consistent with [6].

4.2 BoW Pipeline Evaluation

We study in Table 1 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.

Table 1. Classification results on Caltech-101 dataset with 30 training images per class

Spatial Stride	Scaling	Codebook size	Accuracy (no norm)	Acc. (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

Table 2. Classification results on Scene-15 dataset with 100 training images per class

Spatial Stride	Scaling	Codebook size	Acc. (no norm)	Acc. (ℓ_2-norm)
8	monoscale	1000	78.72 ± 0.62	78.96 ± 0.60
6	monoscale	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

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 3300 also gives good results. Compared to the classical performance of 64% of the BoW SPM [3], 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 6500 with ℓ_2-normalization. The soft BoW pipeline outperforms the advanced methods presented in [15], the Fisher Kernel method (reported at 77.78%), and the LLC (reported at 76.95%) with the same multiscale setup and a codebook of 8000 words (for LLC). It is also above the score of Boureau [17], 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.

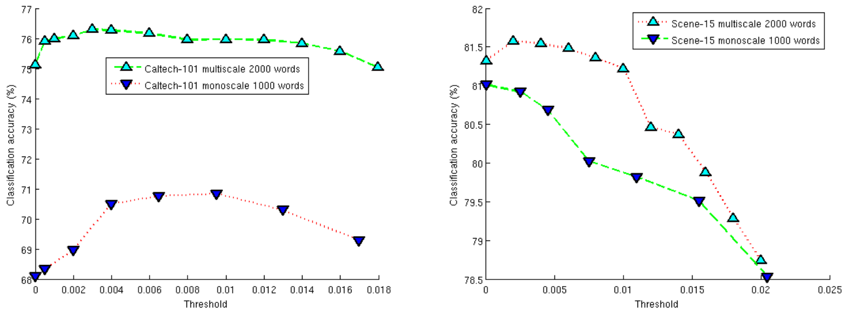
Table 2 reports the experimental results on Scene-15. They are all consistent with the experiments on Caltech-101. The best result of 83.44% is also obtained for a multiscale scheme, a small spatial stride of 3, and a large dictionary of size 6800 with normalization. This score is still slightly better than the Boureau one of 83.3% [17], but remains below state-of-the-art results for that database.

These experiments confirm that the parameters mentioned in section 2 may significantly improve the recognition. A small spatial stride with multiscale, a large 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.

4.3 Evaluation of our Strategy

We evaluate here the classification performances of our early fusion detailed in section 3. First, we study the impact of γ (Eq 2). Fig. 3(a) 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 $\gamma \approx 10^{-2}$ compared to the default value. The same trend appears for the multiscale setting. For Scene-15 dataset (Fig. 3(b)), 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).

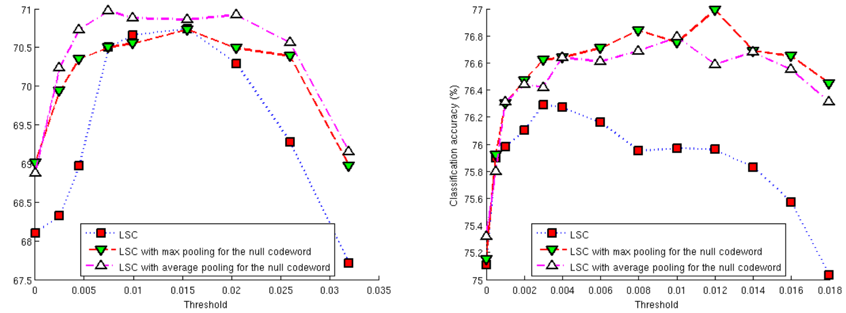
Second, we evaluate the specific encoding and pooling method for low contrast regions described in section 3.2. We provide two gradual evaluations (see Fig. 4). The proposed changes improve performances in Caltech-101 database, in both monoscale (Fig. 4(a)) and multiscale setting (Fig. 4(b)). For the multiscale setup, the performances are in addition more robust to γ variations. For the monoscale



(a) Caltech-101

(b) Scene-15

Fig. 3. 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



(a) monoscale Caltech-101

(b) multiscale Caltech-101

Fig. 4. 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

setup, the average pooling outperforms the max pooling method, validating the idea that enriching the homogeneous regions pooling with a non-binary 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 **79.07±0.83%** on Caltech-101 dataset and **83.71±0.52%** on Scene-15 with our fusion scheme over low/high contrast regions. To the best of our knowledge, this performance on the Caltech-101 benchmark is above all previously published results for a single descriptor type and linear classification.

5 Conclusions

The BoW strategy is still very competitive for image classification. In this paper, we have investigated some early fusion methods to deal with artifacts inherited from dense sampling methods on low contrast regions. We have proposed a novel scheme to efficiently embed this low contrast information into the BoW pipeline. Experiments are provided on Caltech and Scene-15 datasets. We have first shown the great impact of the setting of several low and mid level parameters (density and multiscale sampling, normalization) for object and scene recognition. We have achieved a gain around 20% on Caltech-101 from a monoscale setup with a small dictionary to the winner cocktail combining multiscale dense sampling, soft coding and normalization. Finally, our strategy obtains state-of-the-art performances on Caltech-101 and very good results on Scene-15 dataset.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
2. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. (2003)
3. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
4. Benois-Pineau, J., Bugeau, A., Karaman, S., Mégret, R.: Spatial and multi-resolution context in visual indexing. Visual Indexing and Retrieval (2012) 41–63
5. van Gemert, J., Veenman, C., Smeulders, A., Geusebroek, J.M.: Visual word ambiguity. PAMI (2010)
6. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV. (2011)
7. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR. (2009)
8. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR. (2010)
9. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR. (2010)
10. Boureau, Y., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in vision algorithms. In: ICML. (2010)
11. Snoek, C., Worring, M., Hauptmann, A.: Learning rich semantics from news video archives by style analysis. TOMCCAP **2** (2006)
12. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR Workshop on GMBV. (2004)
13. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/` (2008)
14. Fei-fei, L.: A bayesian hierarchical model for learning natural scene categories. In: CVPR. (2005)
15. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC. (2011)
16. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. PAMI **34** (2011)
17. Boureau, Y., Le Roux, N., Bach, F., Ponce, J., LeCun, Y.: Ask the locals: multi-way local pooling for image recognition. In: ICCV. (2011)
18. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR **9** (2008)