Efficient Bag of Scenes Analysis for Image Categorization

Sébastien PARIS\textsuperscript{1}, Xanadu HALKI\textsuperscript{AS2} and Hervé GLOTIN\textsuperscript{23}
\textsuperscript{1}DYNI team, LSIS CNRS UMR 7296, Aix-Marseille University
\textsuperscript{2}DYNI team, LSIS CNRS UMR 7296, Université Sud Toulon-Var
\textsuperscript{3}Institut Universitaire de France
sebastien.paris@lsis.org, \{halkias, glotin\}@univ-tln.fr

Keywords: Image Categorization, Scenes Categorization, Fine-Grained Visual Categorization, Non-parametric local patterns, Multi-Scale LBP/LTP, Dictionary Learning, Sparse Coding, LASSO, Max-Pooling, SPM, Linear SVM.

Abstract: In this paper, we address the general problem of image/object categorization with a novel approach referred to as Bag-of-Scenes (BoS). Our approach is efficient for low semantic applications such as texture classification as well as for higher semantic tasks such as natural scenes recognition or fine-grained visual categorization (FGVC). It is based on the widely used combination of i) Sparse coding (Sc), ii) Max-pooling and iii) Spatial Pyramid Matching (SPM) techniques applied to histograms of multi-scale Local Binary/Ternary Patterns (LBP/LTP) and its improved variants. This approach can be considered as a two-layer hierarchical architecture: the first layer encodes the local spatial patch structure via histograms of LBP/LTP while the second encodes the relationships between pre-analyzed LBP/LTP-scenes/objects. Our method outperforms SIFT-based approaches using Sc techniques and can be trained efficiently with a simple linear SVM.

1 INTRODUCTION

Image categorization\textsuperscript{1} consists of assigning a unique label with a generally high-level semantic value to an image while FGVC refers to the task of classifying objects that belong to the same basic-level class. Both have long been a challenging problem area in computer vision, biomonitoring and robotics and can mainly be viewed as belonging to the broader supervised classification framework. In scene categorization, the difficulty of the task can be partly explained by the high-dimensional input space of the images as well as the high-level semantic visual concepts that lead to large intra-class variation. For object recognition more specifically, the small aspect ratio (object size vs image size) can induce a high level of uninformative background pixels. A preliminary detection procedure is required to “home-in” the object in a Region of Interest (ROI) (Bosch et al., 2007; Larios et al., 2011).

The direct framework (see Fig.1) in vision systems consists of extracting directly from the images meaningful features (using shape/texture/similarity/color information) in order to achieve the maximum generalization capacity during the classification stage. Examples of such popular features in computer vision and human cognition inspired models include GIST (Oliva and Torralba, 2001), HOG (Dalal and Triggs, 2005), Self-Similarity (Deselaers and Ferrari, 2010) and WLD (Chen et al., 2010).

Widely used in face detection (Fröba and Ernst, 2004; Wu et al., 2011), face recognition (Marcel et al., 2007; Zhang et al., 2007), texture classification (Sadat et al., 2011; Bianconi et al., 2012) and scene categorization (Wu and Rehg, 2008; Gao et al., 2010; Paris and Glotin, 2010; Zhang et al., 2010), Local Binary Pattern (LBP) (Ojala et al., 2002) and recent derivatives such as Local Ternary Pattern (LTP) (Zheng et al., 2010), Gabor-LBP (Zhang et al., 2009; Lee et al., 2010), Local Gradient Pattern (LGP) (Jun and Kim, 2012) or Local Quantized Pattern (LQP) (Hussain and Triggs, 2012) are efficient local micro-patterns that define competitive features achieving state-of-the-art performances.

LBP can be considered as a non-parametric local visual micro-pattern texture, encoding mainly contours and differential excitation information of the 8 neighbors surrounding a central pixel (Heikkilä et al., 2006; Huang et al., 2011). This process represents

\textsuperscript{1}Granted by COGNILEGO ANR 2010-CORD-013 and PEPS RUPTURE Scale Swarm Vision
a contractive mapping from $\mathbb{R}^9 \mapsto N_{9^3} \subset \mathbb{N}^+$ for each local patch $p(x)$ centered in $x$ (Bianconi and Fernández, 2011) provide a theoretical study of LBP. The total number of different LBPs is relatively small and by construction is finite: from 256 up to 512 different patterns (if improved LBP is used).

LTP (Tan and Triggs, 2010) have been extended from LBP as a parametric approximation of a ternary pattern. Instead of mapping $\mathbb{R}^9 \mapsto N_{9^3} \subset \mathbb{N}^+$, they propose to split the ternary pattern into two binary patterns and concatenating the two associated histograms. In (Hussain and Triggs, 2012), they generalize local pattern using LQP by both increasing neighborhood range, number of neighbors and pattern cardinality leading to map $\mathbb{R}^9 \mapsto N_{pW} \subset \mathbb{N}^+$.

Histograms of LBP (HLBP) (respectively HLTP), which count the occurrence of each LBP (respectively LTP) in the scene, can easily capture general structures in the visual scene by integrating information in a ROI, while being less sensitive to local high frequency details. This property is important when the desire is to generalize visual concepts. As depicted in this work, it is advantageous to extend this analysis for several sizes of local ROIs using a spatial pyramid denoted by $\Lambda$.

Recently, the alternative scheme of Bag-of-Features (BoF) has been employed in several computer vision tasks with wide success. It offers a deeper extraction of visual concepts and improves accuracy of computer vision systems. BoF image representation (Willamowski et al., 2004) and its SPM extension (Lazebnik et al., 2006) share the same idea as HLBP: counting the presence (or combination) of visual patterns in the scene. BoF contains at least three modules prior to the classification stage: (i) region selection for patch extraction; (ii) codebook/dictionary generation and feature quantization; (iii) frequency histogram based image representation with SPM. In general, SIFT/HOG patches (Lowe, 2009; Dalal and Triggs, 2005) are employed in the first module. These visual descriptors are then encoded, in an unsupervised manner, into a moderate sized dictionary using Vector Quantization (VQ) (Lazebnik et al., 2006) or sparse coding (Yang et al., 2009b). In (Wu and Rehg, 2009), Wu and al were first to introduce LBP (via CENTRIST) into BoF framework coupled with histogram intersection kernel (HIK).

At least two disadvantages can be addressed against the BoF framework, mainly concerning the second stage. Firstly, and more specifically for FGVC, the trained dictionaries don’t have enough representative basis vectors for some (rare and detailed) local patches that are crucial for discriminativity. Secondly, during quantification/encoding a lot of important information can be lost (Boiman et al., 2008). For these reasons, dictionary-free approaches have been recently introduced. In (Yao and Bradski, 2012), they performed an efficient template matching coupled with a bagging classification procedure. In (Bo et al., 2010; Bo et al., 2011a), they bypass BoF with efficient but computationally expensive hierarchical kernel descriptors. In (Larios et al., 2011; Choi et al., 2012), they proposed patch’s supervised learning (respectively supervised projection) with random forest (respectively with PLS).

In order to improve the encoding scheme, it has been shown that localized soft-assignment (Avila et al., 2011), local-constrained linear coding (LLC) (Oliveira et al., 2012), Fisher vectors (FV) (Perronnin et al., ; Krapac et al., 2011), orthogonal matching pursuit (OMP) (Bo et al., 2011b) or Sparse coding (Sc) (Yang et al., 2009b; Gao et al., 2010) can easily be plugged into the BoF framework as a replacement for VQ. Moreover, pooling techniques coupled with SPM (Lazebnik et al., 2006) can be effectively used as a replacement for the global histogram based image representation.

Our contributions in this paper are two-fold. We first re-introduce two multi-scale variants of the LBP operators and extend two novel multi-scale variants of the LTP operators (Tan and Triggs, 2010). Secondly, we propose to plug HLBP/HLTP into the Sc framework as a second analyzing layer and call this procedure Bag-of-Scenes (BoS). This new approach is efficient as well as for scene categorization, object recognition or FGVC. The novel features can be trained efficiently with simple large-scale linear SVM solver such as Pegasos (Shalev-Shwartz et al., 2007) or LIBLINEAR (Hsieh et al., 2008). BoS can be seen as a two layer Hierarchical BoF analysis: a first fast contractive low-dimension manifold encoder via HLBP/HLTP and a second inflating high-dimension encoder via Sc.

![Figure 1: Comparison of the different frameworks. Left: direct framework. Middle: BoF/Sc framework, Right: Our proposed BoS/Sc framework.](image-url)
2 Histogram of Multi-Scale local patterns

For an image/patch $I$ ($n_1 \times n_2$), we present two existing multi-scale versions of the LBP operator, denoted by the $B$ operator and for its improved variant by the $IB$ operator. We also introduce two novel multi-scale versions of the LTP, denoted by the $T$ operator and for its improved variant by the $IT$ operator.

2.1 Multi-Scale LBP/ILBP

Basically, operator $B$ encodes the relationship between a central block of $(s \times s)$ pixels located in $(y_c,x_c)$ with its 8 neighboring blocks (Liao et al., 2007), whereas operator $IB$ adds a ninth bit encoding a term homogeneous to the differential excitation (see left Fig. 2). Both can be considered as a non-parametric local texture encoder for scale $s$. In order to capture information at different scales, the range analysis $s \in S$, is typically set at $S = \{1,2,3,4\}$ for this paper, where $S = \text{Card}(S)$. These two micro-codes are defined as follows:

$$B(y_c,x_c,s) = \sum_{i=0}^{i=7} 2 I(A_i \geq A_c)$$

$$IB(y_c,x_c,s) = B(y_c,x_c,s) + 2^8 I\left(\sum_{i=0}^{i=7} A_i \geq 8A_c\right).$$

(1)

For $\forall (y_c,x_c) \in R \subset I$, $B(y_c,x_c,s) \in \mathbb{N}_{2^8}$ and $IB(y_c,x_c,s) \in \mathbb{N}_{2^9}$ respectively.

![Figure 2: Left: I and B(y_c,x_c,4) overlaid. Right: corresponding image integral H and the central block A_c. A_c can be efficiently computed with the 4 corner points.](image)

2.2 Multi-Scale LTP/ILTP

We introduce the multi-scale version of LTP and its improved variant. The idea behind LTP is to extend the LBP for $b = 3$ with the help of a single threshold parameter $t \in \mathbb{N}_{25}$. With the same neighborhood configuration with $N = 8$ (see left Fig. 2), a direct extension would conduct to have $3^8 = 6561$ different patterns. In (Tan and Triggs, 2010), they proposed to break the high dimensionality of the code by splitting the ternary code into two binary operators $T_p$ and $T_n$ such as:

$$T_p(y_c,x_c,s,t) = \sum_{i=0}^{i=7} 2^i \left(\frac{1}{2} (A_i - A_c) \geq t\right)$$

$$T_n(y_c,x_c,s,t) = \sum_{i=0}^{i=7} 2^i \left(\frac{1}{2} (A_i - A_c) \leq -t\right).$$

(2)

The improved multi-scale LTP operators (denoted $IT_p$ and $IT_n$) are derived similarly from MSLBP by:

$$IT_p(y_c,x_c,s,t) = T_p(y_c,x_c,s,t) + 2^8 \left(\frac{1}{2} \left(\sum_{i=0}^{i=7} A_i - 8A_c\right) \geq t\right)$$

$$IT_n(y_c,x_c,s,t) = T_n(y_c,x_c,s,t) + 2^8 \left(\frac{1}{2} \left(\sum_{i=0}^{i=7} A_i - 8A_c\right) \leq -t\right).$$

(3)

Now, for $\forall (y_c,x_c) \in R \subset I$, both codes $\{T_p(y_c,x_c,s,t),T_n(y_c,x_c,s,t)\} \in \mathbb{N}_{2^8}$ while the improved version $\{IT_p(y_c,x_c,s,t),IT_n(y_c,x_c,s,t)\} \in \mathbb{N}_{2^9}$ respectively.

2.3 Integral image for fast areas computation

The different areas $\{A_i\}$ and $A_c$ in eq.(1), eq.(2) and eq.(3) can be computed efficiently using the image integral technique (Viola and Jones, 2004). Let’s define $H$ the image integral of $I$ by:

$$H(y,x) \triangleq \sum_{y'\leq y} \sum_{x'\leq x} I(y',x').$$

(4)

Any square area $A(y,x,s) \in R$ (see right Fig. 2) with upper-left corner located in $(y,x)$ and side length $s$ is the addition of only 4 values:

$$A(y,x,s) = H(y+s,x+s) + H(y,x) - H(y+x,s) - H(y+s,x).$$

(5)

2.4 Histogram of local patterns

For all previously defined operators $op \in \{B,IB,T_p,T_n,IT_p,IT_n\}$, efficient features are obtained by counting occurrences of the $j^{th}$ visual LBP/LTP at scale $s$ in a ROI $R \subset I$:

$$z_{op}(R,j,s) = \sum_{(y_c,x_c) \in R} \mathbb{1}_{\{op(y_c,x_c) = j\}}.$$
where \( j = 0, \ldots , b - 1 \) is the \( j \)th bin of the histogram and \( b = \{ 256, 512, 256, 256, 512, 512 \} \) for \( op \in \{ B, IB, T_p, T_n, IT_n, IT_p \} \) respectively.

Full histogram of LBP and variant its ILBP, denoted \( z_B, z_{IB} \), are computed by:
\[
z_{op}(R, s) \triangleq [z_{op}(R, 0, s), \ldots , z_{op}(R, b - 1, s)],
\]
with a total size for patches \( d = b = \{ 256, 512 \} \) respectively.

For LTP, full histograms, denoted \( z_T, z_{IT} \) are defined by:
\[
z_{op}(R, s) \triangleq [z_{op}(R, 0, s), \ldots , z_{op}(R, b - 1, s), \ldots , z_{op}(R, 0, s), \ldots , z_{op}(R, b - 1, s)],
\]
with a total size for patches \( d = 2b = \{ 512, 1024 \} \) respectively.

To end the path extraction stage, regardless the type of histogram of local patterns used, a \( \ell_2 \) clamped normalization procedure (\( \ell_2 \) normalization followed by a saturation with the clamp value and again a \( \ell_2 \) normalization) is performed on each histogram (clamp value = 0.2).

### 3 Sparse Coding on patches of multi-scale local patterns

Following the same framework as in (Lazebnik et al., 2006; Yang et al., 2009b; Bouraeu et al., 2010a; Chatfield et al., 2011), we show here that the traditional BoF approach can be advantageously replaced by i) Sc, ii) max-pooling technique and iii) a simple linear SVM as a classifier since the produced features are mostly linearly separable (see Fig. 1 for synopsis).

#### 3.1 Patches of HB/HIB/HT/HIT

Here, we replace the collection of usual SIFT patches densely sampled on a grid by our HB/HIB/HT/HIT patches seen previously. Specifically, \( F \) patches of size \( (m \times m) \) associated with ROI’s \( \{ O_k \} \) (possibly overlapping) are extracted for \( k = 0, \ldots , F - 1 \) and \( \forall s \in S \) (see Fig. 3). For a faster computation for each scale \( s \), the integral image \( II \) is first computed from \( I \).

For a complete dataset containing \( N \) images and \( \forall s \in S \), we obtain a collection of \( P = TS \) patches \( Z \triangleq \{ z_i \}, i = 1, \ldots , P \), where \( T = NF \). We define, the subset of patches \( z_i \) at scale \( s \) by \( Z(s) \subseteq Z \) with \( T \) elements.

#### 3.2 Sparse coding overview

In order to obtain highly discriminative visual features, a common procedure consists of encoding each patch \( z_i \in Z(s) \) at scale \( s \) through an unsupervised trained dictionary \( D \triangleq \{ d_1, \ldots , d_K \} \in \mathbb{R}^{d \times K} \), where \( K \) denotes the number of dictionary elements, and its corresponding weight vector \( c_i \in \mathbb{R}^K \). In the BoF framework, the vector \( c_i \) is assumed to have only one non-zero element:
\[
\arg \min_{D, c} \sum_{i=1}^T \| z_i - D c_i \|_2^2 \quad s.t. \quad \| c_i \|_0 = 1,
\]
where \( C \triangleq \{ c_1, \ldots , c_K \} \) and \( \| \|_0 \) defines the pseudo zero-norm, where here only one element of \( c_i \) is non-zero. In eq. (8), under these constraints, \( (D, C) \) can be optimized jointly by a K-means algorithm for example.

In the Sc approach, in order to i) reduce the quantization error and ii) to have a more accurate representation of the patches, each vector \( z_i \) is now expressed as a linear combination of a few vectors of the dictionary \( D \) and not only by a single one. Imposing the exact number of non-zero elements in \( c_i \) (sparsity level) involves a non-convex optimization (Mairal et al., 2009). In general, it is preferred to relax this constraint and to use instead an \( \ell_1 \) penalty which also involves sparsity. The problem is then reformulated using the following equation:
\[
\arg \min_{D, C} \sum_{i=1}^T \| z_i - D c_i \|_2^2 + \beta \| c_i \|_{\ell_1} \quad s.t. \quad \| c_i \|_{\ell_1} = 1,
\]
where the sparsity is controlled by the parameter \( \beta \). The last equation is not jointly convex in \( (D, C) \) and a common procedure consists of optimizing alternatively \( D \) given \( C \) by a block coordinate descent and then \( C \) given \( D \) by a LASSO procedure (Tibshirani, 1996). At the end of the process, for each scale \( s \in S \), a trained dictionary \( D(s) \) is obtained.

#### 3.3 Spatial Pyramidal Matching and Max pooling

For an image \( I \) and given a trained dictionary \( D(s) \) for a type of code at scale \( s \), \( F \) sparse vectors \( \{ c_k(s) \} \) are computed by a LASSO algorithm. The final efficient descriptor \( x(s) \triangleq \{ x^0(s), \ldots , x^{K-1} (s) \} \in \mathbb{R}^K \) is obtained by the following max-pooling procedure (Yang et al., 2009b; Bouraeu et al., 2010b):
\[
x^j(s) = \max_{k: O_k \in R} (|c_k^j(s)|), \quad j = 0, \ldots , K - 1,
\]
where each element of \( x(s) \) represents the max-response of the absolute value of sparse codes belonging to the ROI \( R \). In order to improve accuracy, a spatial pyramidal matching procedure helps to perform a more robust local analysis. The spatial pyramid...
\( \mathbf{A} \) has \( V = \sum_{l=0}^{L-1} V_l \) ROIs \( \{ \mathbf{R}_{l,v} \} \) with \( l = 0, \ldots, L-1, v = 0, \ldots, V_l - 1 \) (see Fig. 4 for an example). The quantity \( \mathbf{z}_{l,v}^{j}(s) \) for each ROI \( \mathbf{R}_{l,v} \) is computed by:

\[
x_{l,v}^{j}(s) = \max_{k : \mathbf{O}_{l,v} \in \mathbf{R}_{l,v}} \langle \mathbf{v}^{j}(s) \rangle, \quad j = 0, \ldots, K - 1.
\]

(11)

We reinforce our model by an important normalization step, improving considerably accuracy, consists of the \( \ell_2 \) normalization of all vectors \( \{ \mathbf{x}_{l,v}(s) \} \), \( v = 0, \ldots, V_l - 1, s \in \mathcal{S} \), i.e. belonging to the same pyramidial layer \( l \). This step is also very important and often hidden in the existing literature.

The final descriptor \( \mathbf{x}(\mathbf{A}) \) will be defined by the weighted concatenation of all the \( \mathbf{x}_{l,v}(s) \) vectors, i.e. \( \mathbf{x}(\mathbf{A}) \triangleq \{ \lambda_v \mathbf{x}_{l,v}(s) \} \), \( l = 0, \ldots, L - 1, v = 0, \ldots, V_l - 1 \) and \( \forall s \in \mathcal{S} \). The total size of the feature vector \( \mathbf{x}(\mathbf{A}) \) is \( d = K.V.S \), where typically in our simulations, we fixed \( K = \{1024,2048\} \), \( V = \{10,21,26\} \) and \( S = 4 \). A final \( l_2 \) clamped normalization step is performed on the full vector \( \mathbf{x}(\mathbf{A}) \).

### 4 Linear SVM for supervised training

Let’s assume available a training data set \( \{ \mathbf{x}_{i}(\mathbf{A}), y_{i} \}_{i=1}^{N} \), where \( \mathbf{x}_{i}(\mathbf{A}) \in \mathbb{R}^{d} \) is one of four previously defined features and \( y_{i} \in \{1, \ldots, M\} \), where \( M \) is the number of classes. As in (Yang et al., 2009b; Boureau et al., 2010a), we will use a simple large-scale linear SVM such as LIBLINEAR (Hsieh et al., 2008) with the 1-vs-all multi-class strategy. The associated binary unconstrained convex optimization problem to solve is:

\[
\min_{\mathbf{w}} \left\{ \frac{1}{2} \mathbf{w}^{T} \mathbf{w} + C \sum_{i=1}^{N} \max \left( 1 - y_{i} \mathbf{w}^{T} \mathbf{x}_{i}, 0 \right) \right\},
\]

(12)

where the parameter \( C \) controls the generalization error and is tuned on a specific validation set. LIBLINEAR converges to a solution linearly in \( O(dN) \) compared to \( O(dN^2) \) in the worst case for classic SVM where \( N_{sv} \leq N \) defines the number of support vectors.

### 5 Experimental Results

We test our BoS framework on Scene-15 (Lazebnik et al., 2006), UIUC-Sport (Li, 2007), Caltech101 (Fei-Fei et al., 2007), USCD-Birds200 (Welinder et al., 2010) and Stanford-Dogs120 datasets(Khosla et al., 2011a).

We define our SPM matrix \( \mathbf{A} \) with \( L \) levels such as \( \mathbf{A} \triangleq \{ \mathbf{r}_y, \mathbf{d}_x, \mathbf{d}_y, \mathbf{k} \} \). \( \mathbf{A} \) is matrix of size \( (L \times 5) \). For a level \( l \in \{0, \ldots, L-1\} \), the image \( \mathbf{I} \), with size \( (n_x \times n_y) \), is divided into potentially overlapping sub-windows \( \mathbf{R}_{l,x} \) of size \( (h_y \times w_l) \). All these windows are sharing the same associated weight \( \lambda_l \). In our implementation, \( h_y \triangleq [n_y, r_y, l] \) and \( w_l \triangleq [n_x, r_x, l] \) where \( r_y \) and \( r_x \) and \( \lambda \) are the \( l^{th} \) element of vectors \( \mathbf{r}_y, \mathbf{r}_x, \mathbf{d}_x \) and \( \mathbf{d}_y \) respectively. Sub-window shifts in \( x-y \) axis are defined by integers \( \delta_{y,l} \triangleq [n_y, d_y, l] \) and \( \delta_{x,l} \triangleq [n_x, d_x, l] \) where \( d_y \) and \( d_x \) are elements of \( \mathbf{d}_y \) and \( \mathbf{d}_x \), respectively. Overlapping can be performed if \( d_x, l \leq r_x, l \) and/or \( d_y, l \leq r_y, l \). The total number of sub-windows is equal to:

\[
V = \sum_{l=0}^{L-1} V_l = \sum_{l=0}^{L-1} \left( \frac{1 - r_y, l}{d_y, l} + 1 \right) \left( \frac{1 - r_x, l}{d_x, l} + 1 \right).
\]

(13)

For all dataset used, we used SIFT patches with block size \((16 \times 16)\) pixels and \((26 \times 26)\) pixels for ours HB/HIB/HIT/HIT respectively. For SIFT/HT/HIB/HIT/HIT, we extract \( F = 35.35 \approx 1225 \) patches per scale. For both dictionary learning and sparse codes computation, we fix \( \beta = 0.2 \) and \( N_{it} = 50 \) iterations to train dictionaries. We uses our own modified version of the SPAMS toolbox (Mairal et al., 2009). Finally, we performed 10 cross-validation to
compute the average overall accuracy and its standard deviation using the LIBLINEAR solver and fixing parameter $C = 15$.

5.1 Scene-15 dataset

The Scene-15 dataset contains a total of 4485 images in grey color assigned to $M = 15$ categories. The number of images in each category is ranging from 200 to 400. 100 images per class are used to train, the rest for testing. For this dataset, we define $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix}$, i.e. a two layer spatial pyramid dividing image in third and an overlapping of 50% representing a total of $1 + 25 = 26$ ROIs. For HT and HIT patches, we fix $t = 1$. We select 15000 patches per class (a total of 225000 patches) to train dictionaries via Sc. In Fig. 5, we plot accuracy versus the number of words $K$ in the dictionary training. With our particular choice of $\mathbf{A}$ and for one unique scale, we retrieved results comparable to (Yang et al., 2009b), i.e. 80.28% vs. 81.24% for our implementation. Whatever, the number of scale used and the type of patch, our BoS framework outperforms the SIFT-ScSPM approach. In Tab. 1, we compare our results with the state-of-the-art for this dataset (with $S = 4$ scales). The best performance is actually obtained with the SIFT-LScSPM involving a more sophisticated dictionary training through the Laplacian sparse coding. The latter is very time and memory consuming but is improving results with normal SIFT patches from 80.28% ± 0.93 with simple Sc to 89.75% ± 0.5 with LSc. The second best result is obtained with spatial FV following by the kernel descriptors. For FV, they reduced SIFT to 64 dimension (total size equal to $K(1 + 2d) = 12800$) and used a multi-class logistic regression. It is also worth noting that KDES-EKM uses a concatenation of 3 descriptors coupled with an efficient feature mapping (KDES-A+L SVM got 81.9% ± 0.60 for a fair comparison). However, our results with a single HIT patch and a simple linear SVM are very close. More, if FV or LSc would be used, one can expect better results.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy ± Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-ScSPM ($K = 1024$) (Yang et al., 2009b)</td>
<td>80.28% ± 0.93</td>
</tr>
<tr>
<td>SIFT-MidLevel ($K = 2048$) (Boureau et al., 2010a)</td>
<td>84.20% ± 0.30</td>
</tr>
<tr>
<td>SIFT-LScSPM ($K = 1024$) (Gao et al., 2010)</td>
<td>89.75% ± 0.50</td>
</tr>
<tr>
<td>KDES-EKM ($K = 1000$) (Bo et al., 2010)</td>
<td>86.79%</td>
</tr>
<tr>
<td>PCASIFT-SFV ($K = 100$) (Krapac et al., 2011)</td>
<td>88.29%</td>
</tr>
<tr>
<td>SIFT-DITC ($K = 1000$) (Elfiky et al., 2012)</td>
<td>85.4%</td>
</tr>
<tr>
<td>SIFT-ScSPM ($K = 1024$, our implementation)</td>
<td>81.24% ± 0.73</td>
</tr>
<tr>
<td>HB-ScSPM ($K = 2048$, our work)</td>
<td>86.04% ± 0.36</td>
</tr>
<tr>
<td>HB-ScSPM ($K = 2048$, our work)</td>
<td>86.45% ± 0.44</td>
</tr>
<tr>
<td>HT-ScSPM ($K = 2048$, our work)</td>
<td>86.24% ± 0.43</td>
</tr>
<tr>
<td>HIT-ScSPM ($K = 2048$, our work)</td>
<td>86.53% ± 0.37</td>
</tr>
</tbody>
</table>

Table 1: Recognition rate (and standard deviation) for Scene-15 dataset.

5.2 UIUC-Sport dataset

The UIUC-sport dataset contains a total of 1579 images assigned to $M = 8$ categories. 60 images per class are used to train, 70 for testing. For this dataset, we define $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{bmatrix}$ representing a total of $1 + 9 = 10$ ROIs for SPM. Color (R,G,B) information channels are used, sampling patches and training dictionaries on each of them. For HT and HIT patches, we fix $t = 5$. We select 30000 patches per class (a total of 240000 patches) to train dictio-
**Scenes15 with one scale**

- SIFT-ScSPM, $\sigma = [1]$
- HB-ScSPM, $s = [1]$
- HIB-ScSPM, $s = [1]$
- HT-ScSPM, $s = [1]$
- HIT-ScSPM, $s = [1]$

**Scenes15 with four scales**

- SIFT-ScSPM, $\sigma = [0.5, 0.65, 0.8, 1]$
- HB-ScSPM, $s = [1, 2, 3, 4]$
- HIB-ScSPM, $s = [1, 2, 3, 4]$
- HT-ScSPM, $s = [1, 2, 3, 4]$
- HIT-ScSPM, $s = [1, 2, 3, 4]$

Figure 5: Results for Scenes 15. Left: one scale are used for all kind of patches. Right: four scales are used for all kind of patches.

**UIUC-Sport with one scale**

- SIFT-ScSPM, $\sigma = [1]$
- HB-ScSPM, $s = [1]$
- HIB-ScSPM, $s = [1]$
- HT-ScSPM, $s = [1]$
- HIT-ScSPM, $s = [1]$

**UIUC-Sport with four scales**

- SIFT-ScSPM, $\sigma = [0.5, 0.65, 0.8, 1]$
- HB-ScSPM, $s = [1, 2, 3, 4]$
- HIB-ScSPM, $s = [1, 2, 3, 4]$
- HT-ScSPM, $s = [1, 2, 3, 4]$
- HIT-ScSPM, $s = [1, 2, 3, 4]$

Figure 6: Results for UIUC-Sport. Left: one scale are used for all kind of patches. Right: four scales are used for all kind of patches.

In Fig. 6, we plot accuracy vs. $K$. Notice, that our implementation of SIFT-ScSPM outperforms results from (Yang et al., 2009b). Our choice of $\Lambda$, color information used in training and our specific normalization procedure may explain these improved results. We can also notice, especially for a small dictionary size, that our BoS framework is far superior to SIFT-ScSPM. In Tab. 2, we compare our results with the state-of-the-art (with $S = 4$ scales). To our best of knowledge, our BoS framework, with HIT patch, obtains the state-of-the-art performances with **89.85%** of overall accuracy.

**5.3 Caltech101 dataset**

The Caltech101 dataset contains a total of 9144 images assigned to $M = 102$ categories. 30 images per class are used to train, the rest for testing. For this dataset, we define $\Lambda = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$. We extract 2000 HIT patches per class (a total of 204000 patches) for $S = 4$ scales to train dictionaries via Sc.

In Tab. 3, we compare our results with the state-of-the-art. We separate methods using more sophisticated approaches such as prior detection to localize

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy ± Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-ScSPM ($K = 1024$) (Yang et al., 2009b)</td>
<td>82.70% ± 1.50</td>
</tr>
<tr>
<td>SIFT-LScSPM ($K = 1024$) (Gao et al., 2010)</td>
<td>85.30% ± 0.31</td>
</tr>
<tr>
<td>SIFT-HOMP ($K = 2 \times 1024$) (Bo et al., 2011b)</td>
<td>85.79% ± 1.30</td>
</tr>
<tr>
<td>SIFT-ScSPM ($K = 1024$, our implementation)</td>
<td>87.98% ± 1.08</td>
</tr>
<tr>
<td>HB-ScSPM ($K = 2048$, our work)</td>
<td>87.42% ± 1.27</td>
</tr>
<tr>
<td>HIB-ScSPM ($K = 2048$, our work)</td>
<td>88.44% ± 1.25</td>
</tr>
<tr>
<td>HT-ScSPM ($K = 2048$, our work)</td>
<td>89.35% ± 1.42</td>
</tr>
<tr>
<td>HIT-ScSPM ($K = 2048$, our work)</td>
<td><strong>89.85% ± 1.28</strong></td>
</tr>
</tbody>
</table>

Table 2: Recognition rate (and standard deviation) for UIUC-Sport dataset.
more precisely objects or using complex supervised segmentation with methods classifying directly images. To the best of our knowledge, we have the highest recognition rate (81.05%) for a unique feature coupled with a simple linear SVM. With a medium dictionary size (K = 1024), we are competitive with sophisticated and time-consuming methods using supervised segmentation, graph matching or complex MKL.

5.4 USCD-Birds200 dataset

The USCD-Birds200 dataset is containing a total of 6033 images assigned to M = 200 categories. We crop all images with the provided bounding-box ground-truth. 15 images per class are used to train, the rest for testing. This dataset represents a challenging FGVC task, where categorization must exploits details difference between species. We particularize this task. We particularize the rest for testing. This dataset represents a challenging FGVC task, where categorization must exploits details difference between species. We particularize with methods classifying directly images. To the best of our knowledge, we have the highest recognition rate (81.05%) for a unique feature coupled with a simple linear SVM. With a medium dictionary size (K = 1024), we are competitive with sophisticated and time-consuming methods using supervised segmentation, graph matching or complex MKL.

\[
\Lambda = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Color (R,G,B) information channels are used, sampling patches and training dictionaries on each of them. For the HIT patches, we fix \( t = 5 \). We select 2000 patches per class (a total of 400000 patches) for \( S = 4 \) scales to train dictionaries via Sc. In Tab. 4, we compare our results with the state-of-the-art. To our best of knowledge, our BoS framework, with HIT patch, obtains the state-of-the-art performances with 36.36% of overall accuracy with a unique descriptor and linear SVM. A simple late fusion of SIFT-ScSPM with HIT-ScSPM (product of \( p(y = 1|x) \)) gives a score of 40.03%.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy ± Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCOS-MT (Chai et al., 2011)</td>
<td>16.20%</td>
</tr>
<tr>
<td>Discr. Decision Trees + RF (Yao et al., 2011)</td>
<td>19.20%</td>
</tr>
<tr>
<td>Mult.-Cue+DITC (K = 5000) (Khan et al., 2011)</td>
<td>22.40%</td>
</tr>
<tr>
<td>HIT-ScSPM (K = 1024, our work)</td>
<td>27.93% ± 1.16</td>
</tr>
</tbody>
</table>

Table 4: Recognition rate and standard deviation on the USCD-Birds200 dataset.

5.5 Stanford-Dogs120 dataset

The Stanford-Dogs120 dataset is containing a total of 20580 images assigned to M = 120 categories. We crop all images with the provided bounding-box ground-truth. 100 images per class are used to train, the rest for testing (we use the provided train/test set). This dataset represents also a challenging FGVC task. We particularize

\[
\Lambda = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Color (R,G,B) information channels are used, sampling patches and training dictionaries on each of them. For the HIT patches, we fix \( t = 5 \). We select 2000 patches per class (a total of 240000 patches) for \( S = 3 \) scales (\( S = \{1, 2, 3\} \)) to train dictionaries via Sc. In Tab. 5, we compare our results with the state-of-the-art. To our best of knowledge, our BoS framework, with HIT patch, obtains the state-of-the-art performances with 36.36% of overall accuracy with a unique descriptor and linear SVM. A simple late fusion of SIFT-ScSPM with HIT-ScSPM (product of \( p(y = 1|x) \)) gives a score of 40.03%.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Accuracy ± Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-ScSPM (Khosla et al., 2011b)</td>
<td>26.10%</td>
</tr>
<tr>
<td>SIFT-ScSPM (K = 2048, our implementation)</td>
<td>32.05%</td>
</tr>
<tr>
<td>HIT-ScSPM (K = 2048, our work)</td>
<td>36.36%</td>
</tr>
<tr>
<td>SIFT-ScSPM+HIT-ScSPM (K = 2048, our work)</td>
<td>40.03%</td>
</tr>
</tbody>
</table>

Table 5: Recognition rate and standard deviation on the Stanford-Dogs120 dataset.

6 Conclusions and Perspectives

We have presented in this article the 2-layer BoS architecture mixing HB/HIB/HIT/HT as a fast local textures encoder for the first layer and Sc as scenes encoder for the second. This first hand-graft layer can advantageously replace complex hierarchical feature extractors such as Deep Belief Networks and the patch extraction are even faster than SIFT ones, thanks to the integral image technique. Achieved performances outperform state-of-the-art results with a simple linear SVM as well for object recognition tasks as for FGVC ones.

As potential future works, many perspectives can be investigated. For example, complementary patch, multi-scale variants of LPQ could be coupled with our HB/HIB/HIT/HT approach, in order train a unique dictionary with these fused patches. Higher dimension local pattern can be also associated with the Sc framework such those proposed by (Hussain and Triggs, 2012). Finally, experimenting with LSc (Gao et al., 2010) or FV (Krapac et al., 2011) should improve the encoding part of the pipeline, while supervised pooling techniques (Jia et al., 2011) will surely also improve results.

REFERENCES

Table 3: Recognition rate (and standard deviation) for Caltech101 dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Algorithms</th>
<th>Accuracy ± Std (15 Train)</th>
<th>Accuracy ± Std (30 Train)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Matching + SVM</td>
<td>MLMRF+Curv. Expen.</td>
<td>75.30% ± 0.70</td>
<td>80.30% ± 1.20</td>
</tr>
<tr>
<td>Detec. + Multi Non-Lin Ker</td>
<td>Multiway-SVM (Bosch et al., 2007)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Superv. Segm+Classif</td>
<td>Subcat. Relevances</td>
<td>72.00% ± 2.00</td>
<td>82.00% ± 0.70</td>
</tr>
<tr>
<td>Superv. Segm+Classif</td>
<td>Sift-CDBN (Li et al., 2010)</td>
<td>72.60% ± 2.00</td>
<td>79.20% ± 0.70</td>
</tr>
<tr>
<td>Superv. Segm+Classif</td>
<td>Sift-LaRank (Li et al., 2012)</td>
<td>74.70% ± 2.00</td>
<td>82.30% ± 0.70</td>
</tr>
<tr>
<td>Superv. Segm+Classif</td>
<td>GS-MKL (Yang et al., 2009a)</td>
<td>73.20% ± 0.70</td>
<td>84.30% ± 0.70</td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>Sift-CDBN (K = 1024, our work)</td>
<td>77.30% ± 0.60</td>
<td></td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>Sift-LaRank (K = 4096) (Oliveira et al., 2012)</td>
<td>73.09% ± 0.77</td>
<td>80.02% ± 0.36</td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>HT-ScSPM (K = 2048, our work)</td>
<td>74.24% ± 0.69</td>
<td>81.05% ± 0.43</td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>HT-ScSPM (K = 2048, our work)</td>
<td>73.92% ± 0.81</td>
<td>80.90% ± 0.38</td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>HT-ScSPM (K = 2048, our work)</td>
<td>73.23% ± 0.69</td>
<td>80.51% ± 0.46</td>
</tr>
<tr>
<td>Superv. Segm+Non-Lin Ker</td>
<td>HT-ScSPM (K = 2048, our work)</td>
<td>72.54% ± 0.70</td>
<td>80.27% ± 0.44</td>
</tr>
</tbody>
</table>


