New Graph regularized Sparse Coding Improving Automatic Image Annotation

Céline RABOUY[1,2], Sébastien PARIS[1,2] and Hervé GLOTIN[1,2,3]

[1]Aix-Marseille Université, CNRS, ENSAM, LSIS UMR 7296, 13397 Marseille, France
[2]Université de Toulon, CNRS, LSIS UMR 7296, 83957 La Garde, France
[3]Institut Universitaire de France, 75005 Paris, France
{celine.rabouy,sebastien.paris}@lsis.org
glotin@univ-tln.fr

Abstract. Typical image classification pipeline for shallow architecture can be summarized by the following three main steps: i) a projection in high dimensional space of local features, ii) sparse constraints for the encoding scheme and iii) a pooling operation to obtain a global representation invariant to common transformation. Sparse Coding (SC) framework is one particular example of this general approach. The main problem raised by it is the local feature encoding which is done independently, loosing correlation of the input space. In this work we propose to simultaneously encode sparse codes to tackle this problem with Joint Sparse Coding (JSC) inspired by Graph regularized Sparse Coding (GSC). We experiment SC, GSC and JSC on UIUCsports and scenes15 database. We will show that results obtained, for UIUCsports, with SC (87.27 ± 1.33), JSC (84.17 ± 1.57) and the State-of-the-Art (88.47 ± 2.32 [23]) are tackled by a simple fusion (95.37 ± 1.29). Several assumptions will be advanced to explain this phenomenon which can't be generalized.

Keywords: Scenes categorization, Sparse Coding, Graph regularized Sparse Coding, Dictionary Learning, Scale Invariant Feature Transform, Spatial Pyramid Matching, Joint Sparse Coding.

1 Introduction

In the field of computer vision and signal processing, significant progress has been made since the 2000s with more general methods such as Bag of Words (BoW) [19]. We have at our disposal a significant number of databases as, for example, UIUCsportss [11], scenes from 15 databases [8], where the goal is to label images into a finite number of classes. The first way could be to evaluate the metric distance between two images. Unfortunately, due to the high dimensionality of this input space, most of these distances are concentrated into a sub-manifold whatever the image class, making the discrimination by direct distances not robust. To overcome this problem, a solution has to be designed to find a general application $\Psi^j(.;\mu^j)$ with parameter μ^j which characterizes the class $\mathcal{C}^j$ satisfying:

$$\begin{cases} \text{dist}(\Psi^j(\mathbf{I}_1;\mu^j),\Psi^j(\mathbf{I}_2;\mu^j)) \to 0 & \text{if } \mathbf{I}_1 \in \mathcal{C}^j \text{ and } \mathbf{I}_2 \in \mathcal{C}^j \\ \text{dist}(\Psi^j(\mathbf{I}_1;\mu^j),\Psi^j(\mathbf{I}_2;\mu^j)) \to \infty & \text{if } \mathbf{I}_1 \in \mathcal{C}^j \text{ and } \mathbf{I}_2 \notin \mathcal{C}^j, \end{cases} \quad (1)$$

where $\mathbf{I}_1$ and $\mathbf{I}_2$ are two images. The choice of Ψ^j represents a trade-off between its representation capacity versus the μ^j optimization difficulty. In general, in order to estimate/optimize μ^j, we have to start from a local representation (patches) $\mathbf{x} \in \mathbb{R}^d$ to obtain the global representation $\Psi^j(.;\mu^j)$. From Ψ^j associated to BoW, Sparse Coding (SC) [21], up to ConvNet [3, 9] follow the three main procedures: i) high dimension local feature projection, ii) sparsity constraints into the representation model and iii) non-linearity operation and pooling to obtain a global invariant representation.

In this article, we will focus on a new formulation of encoding method, which corresponds more specifically to procedure ii), inspired by SC and more generally by Graph regularized Sparse Coding (GSC) [25]. This new formulation allows to encode simultaneously testing patches as with the GSC model which has good properties. Although we will only work on a single layer, we will show that a simple fusion will allow to improve considerably the classification accuracy and that our results will be close to CNN (convolutional neural nets) [6, 18] initialized on Image Net as shown in [3]. This article is divided into five parts. The first part focuses on SC models and its derivatives (GSC especially). The second part presents our modeling Joint Sparse Coding (JSC). The third part presents Graph regularized Sparse Coding (GSC) dictionary inspired by [13]. A fourth part presents results we obtained on UIUCsports and scenes15 databases and in the last part, we conclude on our contribution.

2 Related Works

In this part, we will focus on the encoding step using linear coding to reconstruct inputs. An approximation of any patches $\mathbf{x} \in \mathbb{R}^d$ can be given by $\mathbf{x}_i = \mathbf{D}\alpha_i$, where $\mathbf{D} \triangleq [\mathbf{d}_1,\ldots,\mathbf{d}_K] \in \mathbb{R}^{d\times K}$ is a given/trained dictionary where $\forall k = 1,\ldots,K,\ \|\mathbf{d}_k^T\mathbf{d}_k\|_2^2 = 1$ and $d_k^j \geq 0$. A patch is a vector extracted from an image. A dictionary is a matrix of "words" allowing the patch reconstruction. In many encoding methods, three common steps can be found: i) a projection into a higher dimension space with ($K >> d$) ii) sparse constraints and iii) a non-linear operation procedure. If α_i^* is obtained with Ordinary Least Square (OLS), the solution is full dense (all elements are non zero). One way to get around this problem is the use of the ℓ_1-norm constraint which corresponds to Lasso problem [21] or Basis Pursuit [4]:

$$\mathcal{L}_{SC}(\alpha_i|\mathbf{x}_i;\mathbf{D}) = \min_{\alpha_i\in\mathbb{R}^K} \frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1, \quad (2)$$

with λ the regularization parameter associated to the SC formulation. This parameter controls the sparsity level as is shown in [15]. Thus, the more λ is large, the more α_i^* (solution of eq.2) will be sparse.

Usually in SC framework, if we take two neighbor patches $\mathbf{x}_i$ and $\mathbf{x}_j$ (with a strong correlation between them), their respective sparse codes, α_i and α_j, can lose this strong correlation, especially indexes of non-zero inputs can completely mismatch. It means they are involving different atoms for their patches' reconstructions. An atom is an element of the vector patch. There exist some SC variations which have been introduced to tackle this behaviour. Principles of this improvement can be divided into two categories: one plays on adding of proximity constraint into the loss directly while the second adds some extra terms into the regularization term. To illustrate the first category, we can cite two approaches: Local Constrained linear Coding (LCC) [24] and the Local Sparse Coding (LSC) [20]. In the second category, we can mention GSC [25].

We will define the set of pre-computed sparse codes of $\mathbf{X}^{train} \triangleq \{\mathbf{x}_1^{train}, \dots, \mathbf{x}_{N^{train}}^{train}\}$ by $\mathbf{A}^{train} \triangleq \{\alpha_1^{train}, \dots, \alpha_{N^{train}}^{train}\}$ where N^{train} designates the number of local features sampled from the training set. Indeed, this adds a spatial constraint in the regularization term. Its equation is:

$$\mathcal{L}_{GSC}(\alpha_i|\mathbf{x}_i, \mathbf{A}^{train}; \mathbf{D}, \lambda, \beta) = \min_{\alpha_i \in \mathbb{R}^K} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 + \beta L_{ii}\alpha_i^T\alpha_i + 2\beta\alpha_i^T\mathbf{h}_i, \quad (3)$$

where $\mathbf{h}_i = \sum_{j \neq i}^{N^{train}} L_{ij}\alpha_j^{train}$, $\mathbf{L} = \left\{L_{ij}\right\}_{i,j=1,\dots,N^{train}}$ is a Laplacian matrix and β a regularization parameter. The matrix $\mathbf{L}$ is defined by $\mathbf{L} = \mathbf{S} - \mathbf{W}$, where $\mathbf{W}$ is a weight matrix with and $W_{i,j} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j^{train}\|_2^2}{\sigma^2}\}$ if $\mathbf{x}_j^{train} \in V(\mathbf{x}_i)$ (where $V(\mathbf{x}_i)$ is the set of neighborhood of $\mathbf{x}_i$ excluding $\mathbf{x}_i$ itself), $W_{i,j} = 0$ else. The matrix $\mathbf{S}$ is diagonal and $S_{i,i} = \sum_{j=1}^{N^{train}} W_{i,j}$. We propose to improve SC by simultaneously encoding all the test local patches (for example associated with a test image). This new modeling will be inspired from the GSC.

3 Joint Sparse Coding - JSC

JSC principle is to jointly encode **all** local features $\mathbf{X}^{test} = \{\mathbf{x}_1^{test}, \dots, \mathbf{x}_{N^{test}}^{test}\}$ **simultaneously** to overcome the decorrelation problem. We also enforce $\alpha_i^k \geq 0$ in the previous optimization problem. This additional constraint improves pooling performances, thus avoiding to pool simultaneously on positive and negative sparse code values and decreasing as a consequence the final size vector by a factor by two. The equation of our modeling is very similar to GSC:

$$\mathcal{L}_{JSC}(\alpha_i|\mathbf{x}_i, \mathbf{A}^{test}; \mathbf{D}, \lambda) = \min_{\alpha_i \in \mathbb{R}^K} \|\mathbf{x}_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 + \beta L_{ii}\alpha_i^T\alpha_i + 2\beta\alpha_i^T\mathbf{h}_i, \; s.t. \; \alpha_i^k \geq 0, \quad (4)$$

where $\mathbf{h}_i = \sum_{j \neq i}^{N^{test}} L_{ij}\alpha_j^{test}$, $\mathbf{L} = \left\{L_{ij}\right\}_{i,j=1,\dots,N^{test}}$ is a Laplacian matrix, β a regularization parameter. Here, $\mathbf{L} = \mathbf{S} - \mathbf{W}$, where $W_{i,j} = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j^{test}\|_2^2}{\sigma^2}\}$ if $\mathbf{x}_j^{test} \in V(\mathbf{x}_i)$, $W_{i,j} = 0$ else and $S_{i,i} = \sum_{j=1}^{N^{test}} W_{i,j}$. Here, $\mathbf{A}^{test} \triangleq \{\alpha_1^{test}, \dots, \alpha_{N^{test}}^{test}\}$ are computed and stacked initially. In practice $N^{test} << N^{train}$, so we need to store only a sparse $K \times N^{test}$ matrix.

Our Laplacian matrix ($N^{test} \times N^{test}$) is very sparse. If we don't need to compute the full matrix, one way is to only calculate the non-zero elements ($(v+1) \times N^{test}$) with the previous formulation. Each column of this $((v+1) \times N^{test})$ matrix is denoted by $\mathbf{L}_i$. To realize this, we use a fast NN-search technical (FLANN) [14] which speeds up the computation considerably. Thus, the solution of eq.4 is given by a modified Feature Sign Search (FSS) algorithm [10] by adding a) a positivity constraint on sparse codes and b) integrating the two right terms (in β) of eq.4 in the gradient formulation used during the FSS algorithm. JSC is given by the algorithm 1. To illustrate the correlation problem, viewed with SC, we compare the normalized correlation computed between two inputs vectors with the normalized correlation computed with their respective output vectors. In this example, 300 different pairs, extracted from UIUCsports local features, are chosen to realize this. The normalized correlation formulation between $\mathbf{x}$ and $\mathbf{y}$ is given by $\rho(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2} \in [0,1]$. We also introduce the scalar value $\overline{\nabla\rho}^2 = \frac{2}{300\times299}\sum_{i=1}^{300}\sum_{j=1}^{j<i}[\rho(\mathbf{x}_i,\mathbf{x}_j)-\rho(\alpha_i,\alpha_j)]^2$ which measures the average quadratic difference between normalized correlation of the input space and the output space. The lower $\overline{\nabla\rho}^2$ is the better. Table 1 summarizes our results including the sparsity percentage. The last line presents $\rho(\alpha_i,\alpha_j)$ correlation associated to output space, for a strong correlation $\rho(\mathbf{x}_i,\mathbf{x}_j) = 90\%$ in input space. We note that the correlation gain is accom-

Algorithm 1 Joint Sparse Coding

```
Inputs: $\mathbf{D}$, $\lambda$, $\beta$, $\mathbf{X}^{test}$, $\sigma$ and $v$
for $i = 1 : N^{test}$ do
    [$\mathbf{V}_i$, $\mathbf{dist}_i$] = $v$-nn search of $\mathbf{x}_i^{test}$ into $\mathbf{X}^{test}$
    $\mathbf{V}_i$ are indexes of $\mathbf{x}_i$ neighbors in $\mathbf{X}^{test}$
    Compute $\mathbf{L}_i$ from $\mathbf{dist}_i$ and $\sigma$
end for
$\mathbf{A}^{test} = \text{lasso}(\mathbf{X}^{test};\mathbf{D},\lambda)$
for $i = 1 : N^{test}$ do
    $\alpha_i = \text{JSC}(\mathbf{x}_i^{test},\mathbf{A}^{test},\mathbf{D},\mathbf{L}_i,\mathbf{V}_i,\lambda,\beta)$
end for
Output: $\mathbf{A}^{test}$
```

Method	SC (0.2)	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
Level Sparsity	**5.82%**	9.36%	15.05%	17.66%	22.75%
$\overline{\nabla\rho}^2$	126.75	116.59	81.83	108.77	**73.35**
$\rho = 90\%$	31%	75%	63%	**79%**	70%

Table 1. $\overline{\nabla\rho}^2$ and correlation $\rho = 90\%$, as an example of strong correlation, for SC, GSC and JSC for two couples (λ,β), on testing patches. The lower $\overline{\nabla\rho}^2$ is obtained for JSC (0.2,0.2)and the best result for correlation parameter ρ is for GSC (0.2,0.2), however, the low sparsity level is obtained for SC.

panied by a sparsity level drop. Thus, λ is increasing sparsity while β is working in the opposite direction.

4 Dictionary Learning

The analytical solution to update a dictionary $\mathbf{D} \triangleq [\mathbf{d}_1,\ldots,\mathbf{d}_K]$ off-line exists and it is formulated as $\mathbf{D} = (\mathbf{X}\mathbf{A}^T)(\mathbf{A}\mathbf{A}^T)^{-1}$, where $\mathbf{A} \triangleq \{\alpha_i\}, i = 1,\ldots,N$ and $\mathbf{A} \in \mathbb{R}^{K\times N}$. The problems comes from the computation of $(\mathbf{A}\mathbf{A}^T)^{-1}$. It is a matrix of size $(K \times K)$ and the computational complexity of this matrix inversion is in $O(K^3)$. Moreover, we have to store the matrix $\mathbf{A}$ in central memory. Thus, we want efficient methods (in term of complexity and memory occupation) to train such dictionaries under basis constraints. One would minimize the regularized empirical risk $\mathcal{R}_n$:

$$\mathcal{R}_N(\mathbf{A},\mathbf{D}) \triangleq \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{x}_i; f(\alpha_i,\mathbf{D})) + \Gamma(\mathbf{A}), \tag{5}$$

where $f(\alpha_i,\mathbf{D}) = \mathbf{D}\alpha_i$, $l(.)$ is typically a quadratic loss function and $\Gamma(.)$ represents the regularization term (for example SC and GSC regularization terms). Eq. 5 would be optimized iteratively by a (stochastic) gradient descent. Unfortunately, the problem is not jointly convex but only conditionally convex. Alternatively, we can minimize:

$$\mathcal{R}_N(\mathbf{A}|\hat{\mathbf{D}}) \triangleq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{x}_i - \hat{\mathbf{D}}\alpha_i\|_2^2 + \Gamma(\alpha_i), \;\; s.t. \;\; \alpha_i^k \geq 1 \tag{6}$$

and

$$\mathcal{R}_N(\mathbf{D}|\hat{\mathbf{A}}) \triangleq \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\hat{\alpha}_i\|_2^2 \;\; s.t.\|\mathbf{d}_k^T\mathbf{d}_k\|_2^2 = 1 \;\text{ and }\; d_k^j \geq 0. \tag{7}$$

In order to obtain a suboptimal solution of eq. 5., eq. 6 can be solved efficiently in parallel *via* SC/GSC procedures while eq. 7 can be solved by a constrained linear system [13].

5 Experiments

5.1 Metrics

In this section we present some results obtained with SC and GSC dictionaries when we use SC and JSC for the encoding part. We fix the dictionary size to $K = 1024$ and a positivity constraint on dictionary columns and sparse codes are applied. The regularization parameters are $\lambda = 0.2$ for SC, ($\lambda = 0.4$; $\beta = 0.2$) and ($\lambda = 0.2$; $\beta = 0.2$) for GSC and JSC for encoding part. Only the GSC ($\lambda = 0.2$, $\beta = 0.2$) dictionary will be used. We measure a classification rate given by a 1-vs-all approach thanks to a linear Support Vector Machine (SVM). Its regularization parameter is fixed to $C = 0.07$. This classification is made by an Average Overall Accuracy (AOA):

$$AOA = \frac{1}{M}\sum_{m=1}^{N}\left\{\frac{1}{N}\sum_{i=1}^{N}\delta(\hat{y}_{i,m} - y_{i,m})\right\}, \tag{8}$$

where N represents the number of available data, δ the loss function chosen (mean square error), M, the number of cross validation and $\hat{y}_{i,m}$ and $y_{i,m}$, the true and predicted label. We realize our experiments on UIUCsportss database [11] and scenes15 database [8]. UIUCsportss database contains 1579 images from 8 different classes. The number of images in each class varies from 137 to 250. We randomly select 70 images from each class for training and 60 for testing. scenes15 database contains 4485 images belonging to 15 different categories and the number of images per class varies between 200 to 400. 100 images are selected for training part and the others for testing part. In our experiments, $M = 10$, $N_{UIUCsports} = 60 \times 8 = 480$ and $N_{scenes15} = 4485 - 15 \times 100 = 2985$. We extract densely SIFT patches (24×24) [12] with a grey level and on one scale. The grid size is 80×80 for UIUCsportss database and 30×30 for scenes15 database. We apply a Spatial Pyramid Matching (SPM) [8] which is defined on L levels. For UIUCsportss, $L = 2$, thus pooling is performed on the entire image ((1×1) - first layer) and the second layer on (2×2) grid with stride of 25%. For scenes15, $L = 3$, thus we use (1×1), (2×2) and (4×4) sub-regions for SPM. We apply μ-pooling ($\mu = 2.5$) for the pooling step [1].

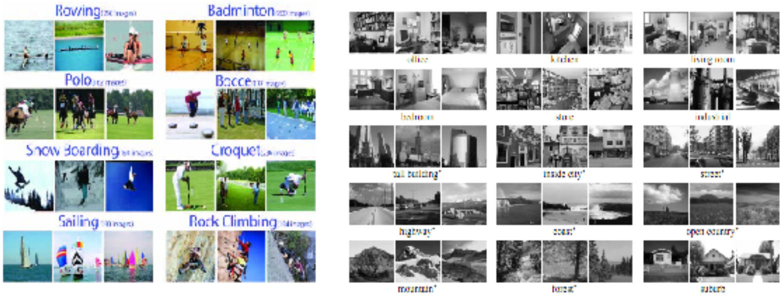

Fig. 1. UIUCsports dataset (left) - scenes15 (right)

5.2 Results on UIUCsports

Table 2 summarizes obtained results. We observe different behaviours. If we focus on encoding part variations (horizontal reading), we see that for all dictionaries choices, SC encoding is the best. Any gain is viewed for the others and a similar behavior is obtained if we read the table vertically. To go further more, in order to evaluate if SC and JSC models are complementary, we measure the accuracy of the arithmetic and geometric means of their estimates (AOA arithmetic and AOA geometric). AOA arithmetic is defined as the sum of probabilities of two selected models and AOA geometric as the

[1] As remind, μ-pooling is written as $f(\mathbf{v};\mathbf{w},\mu) = \sum_{m=1}^{c} w_m v_m^{\mu} = \mathbf{w}^T \mathbf{v}^{\mu}$ $s.t.\|\mathbf{w}\|_2^2 = 1$ and $\mu \neq 0$, where $\mathbf{v}^{\mu} = \left\{\alpha_m^{\mu}\right\}, m = 1,\ldots,c$ and w_m encodes the contribution of the m-image location for specific visual words [7]

Encoding \ Dictionary	SC (0.2)	GSC (0.4, 0.2)	JSC (0.4, 0.2)	GSC (0.2, 0.2)	JSC (0.2, 0.2)
SC (0.2)	**87.27 ± 1.33**	80.75 ± 1.69	83.6 ± 1.66	80 ± 2.01	84.17 ± 1.57
GSC (0.2, 0.2)	**84.81 ± 1.87**	80.71 ± 2.05	81.6 ± 1.77	80.92 ± 2.15	84.17 ± 1.02

Table 2. Evolution of the Average Overall Accuracy for UIUCsports database. The best result is obtained with the couple SC dictionary and SC encoding

square root of the product of two selected models. Tables 3 and 4, associated to figures 2 and 3 respectively (only the arithmetic fusion is showed here, because geometric fusion is lower than the first), summarize results obtained with initial models and their associated fusion. Table 3 corresponds to a horizontal reading (encoding fusion) and table 4 to a vertical reading (dictionary fusion) for UIUCsports. We notice an important relative gain (until +8 points) with SC dictionary. This is less significant with GSC (0.2,0.2) dictionary where few relative gains are observed. For dictionary fusion, strong relative

Dictionary \ Encoding fusion		SC + GSC (0.4,0.2)	SC + JSC (0.4,0.2)	SC + GSC (0.2,0.2)	SC + JSC (0.2,0.2)
SC	AOA arithmetic	94.31 ± 1.28	94.77 ± 1.31	94.23 ± 1.3	**94.94 ± 1.05**
	AOA geometric	93.33 ± 1.23	93.94 ± 1.19	93.37 ± 1.22	94.19 ± 1.2
GSC (0.2,0.2)	AOA arithmetic	84.42 ± 1.5	85.08 ± 1.67	84.37 ± 1.51	85.12 ± 1.62
	AOA geometric	84.48 ± 1.52	84.9 ± 1.62	84.5 ± 1.65	84.98 ± 1.61

Table 3. Evolution of the arithmetic and geometric Accuracy for UIUCsportss database (encoding fusion). The best result is obtained with the couple SC dictionary associated with SC and JSC (0.2,0.2) encodings. An illustration is given in figure 2.

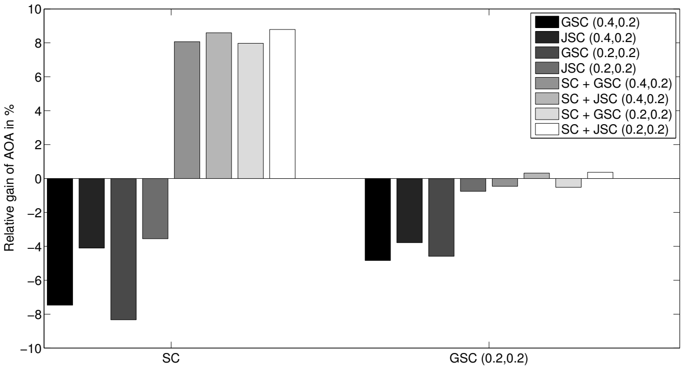

Fig. 2. Benefits and deficits obtained with GSC, JSC and arithmetic fusions encodings compared to SC encoding for the three different dictionaries for UIUCsports database.

Encoding / Dictionary fusion		SC	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
SC+GSC(0.2,0.2)	AOA arithmetic	**95.37 ± 1.29**	92.56 ± 1.11	83.33 ± 1.36	92.46 ± 1.15	84.25 ± 1.22
	AOA Geometric	94.62 ± 1.15	92.31 ± 1.42	83.89 ± 1.29	91.21 ± 1.58	84.31 ± 1.57

Table 4. Evolution of the arithmetic and geometric Accuracy for UIUCsportss database (dictionary fusion). The best result is obtained with the couple SC and GSC (0.2,0.2) dictionaries associated with SC encoding.An illustration is given in figure 3

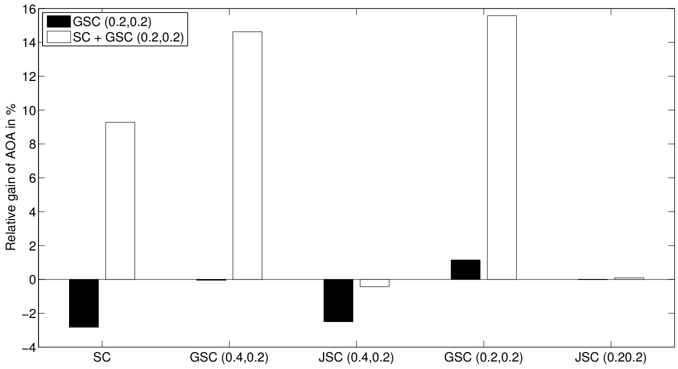

Fig. 3. Beneficits and deficits obtained with GSC and arithmetic fusions dictionaries compared to SC dictionary with five different encoding method choices for UIUCsports database.

gains are viewed for SC and the two GSC encoding models. There is no gain for the two JSC encoding models. The best result is for SC dictionary and encoding with SC and GSC (0.2,0.2) dictionary with SC encoding.

5.3 Results on scenes15

The table 5 summarizes our results: No gain is observed for this dataset. The best results are for SC dictionary and encoding. Fusion results which follow, are summarized in tables 6 and 7 which present fusion results obtained. Figures 4 and 5 illustrate the previous tables respectively. We notice that the behaviour is inverted for the two fu-

Dictionary / Encoding	SC (0.2)	GSC (0.4, 0.2)	JSC (0.4, 0.2)	GSC (0.2, 0.2)	JSC (0.2, 0.2)
SC (0.2)	**84.69 ± 0.6**	80.31 ± 0.6	80.82 ± 0.63	80.59 ± 0.64	81.47 ± 0.47
GSC (0.2, 0.2)	**83.35 ± 0.59**	78.79 ± 0.66	78.4 ± 0.79	79.06 ± 0.62	80.81 ± 0.66

Table 5. Evolution of the Average Overall Accuracy for scenes15 database. The best result is obtained with the couple SC dictionary and SC encoding

Dictionary \ Encoding fusion		SC + GSC (0.4,0.2)	SC + JSC (0.4,0.2)	SC + GSC (0.2,0.2)	SC + JSC (0.2,0.2)
SC	AOA arithmetic	82.97 ± 0.69	83.46 ± 0.51	83.09 ± 0.62	83.60 ± 0.43
	AOA geometric	83.04 ± 0.69	83.48 ± 0.46	83.59 ± 0.59	**83.67 ± 0.42**
GSC (0.2,0.2)	AOA arithmetic	82.66 ± 0.66	82.33 ± 0.76	82.67 ± 0.52	82.79 ± 0.73
	AOA geometric	82.62 ± 0.71	82.44 ± 0.74	82.85 ± 0.62	82.84 ± 0.72

Table 6. Evolution of the arithmetic and geometric Accuracy for scenes15 database (encoding fusion). No results improve tha of SC. An illustration is given in figure 4.

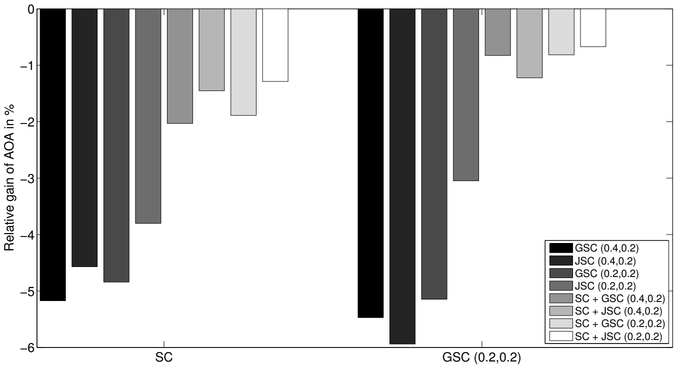

Fig. 4. Benefits and deficits obtained with GSC, JSC and arithmetic fusions encodings compared to SC encoding for the three different dictionaries for scenes15 database.

Dictionary fusion \ Encoding		SC	GSC (0.4,0.2)	JSC (0.4,0.2)	GSC (0.2,0.2)	JSC (0.2,0.2)
SC + GSC (0.2,0.2)	AOA arithmetic	**84.66 ± 0.64**	79.76 ± 0.62	81.4 ± 0.71	80.41 ± 0.67	82.35 ± 0.75
	AOA Geometric	84.62 ± 0.71	79.76 ± 0.63	81.38 ± 0.69	80.47 ± 0.57	82.2 ± 0.77

Table 7. Evolution of the arithmetic and geometric Accuracy for scenes15 database (dictionary fusion). The best result is obtained with the couple SC and GSC (0.2,0.2) dictionaries associated with SC encoding.An illustration is given in figure 5

sion cases. However, the deficits decrease with fusion and more specifically for GSC (0.2,0.2) dictionary. For the dictionaries fusion, it is between the two models that we obtain the most significant gain. The best result is for the couple (SC + GSC) dictionary associated with SC encoding.

5.4 Weighted fusion

To go further more, we plot the accuracy for a weighted arithmetic fusion. In a first time, the weights are the same for each classes and curves of figure 6 illustrate the weighted arithmetic fusion ($AOA_{arith} =_{SC} +(1-\mu)AOA_{GSC}$). We notice for UIUCsports, when we use adapted coefficients with fusion, no improvement is observed and the accuracy

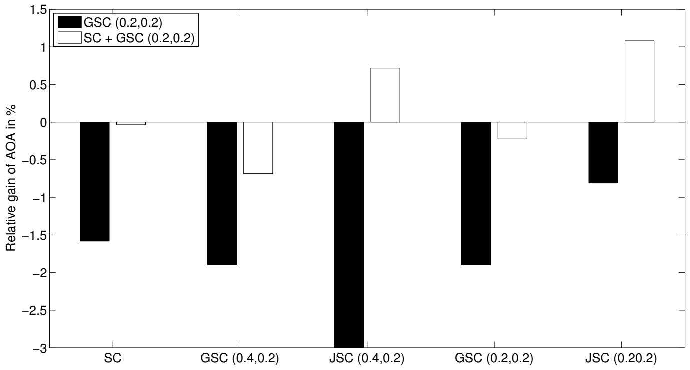

Fig. 5. Beneficits and deficits obtained with GSC and arithmetic fusions dictionaries compared to SC dictionary with five different encoding method choices for scenes15 database.

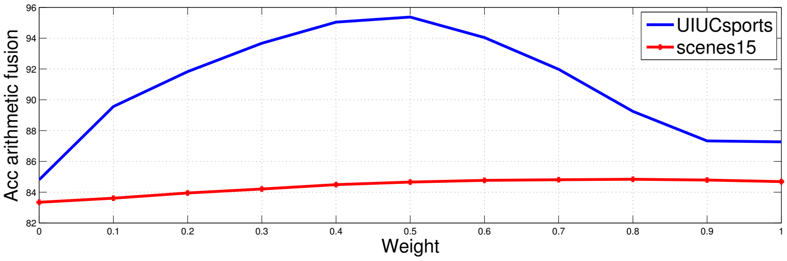

Fig. 6. Evolution of the accuracy with different coefficient. The first point corresponds to the chosen model for fusion and the last point is the SC model. Notice the best result for UIUCsports is obtained with a coefficient of 0.5, and for scenes15, it is 0.8 for SC and 0.2 for GSC (0.2,0.2) dictionary associated with SC. For these two examples, the fusion is between SC (dictionary and encoding) and GSC (0.2,0.2) dictionary with SC encoding.

decreases considerably for other couples. For scenes15, a very small improvement is seen but it does not allow us to conclude to the real benefit of the method. Another alternative would be to calculate others means as harmonic or energy means for examples. Also, the considerable gain obtained with UIUCsports database can be explained by putting forward two assumptions: the heterogeneity between images of training and testing sets and the correlation conservation between the input and output space. The study conducted so far shows that the second assumption is the one that goes in the right direction.

6 Conclusion

Although the results obtained with GSC and JSC alone are not living up to our expectations, we highlight the relevance of our proposal, thanks to the fusion procedure which

	Initial accuracy	Blinded fusion	Weighted fusion	State-of-the-Art
UIUCsports	87.27% ± 1.33	**95.37% ± 1.29**	**95.37% ± 1.29**	88.47 ± 2.32 [23]
scenes15	**84.69% ± 0.6**	**84.66% ± 0.64**	**84.88% ± 0.55**	81.04% ± 0.5 [8]

Table 8. Summarize of fusion results - details in Tables 2, 3, 4, 5, 6, 7.

greatly improves the State-of-the-Art for UIUCsports (88.47 ± 2.32) of [23] (our modeling: 95.37 ± 1.29). A complete study must be realized with different couples (λ, β) for dictionary and encoding parts to find the right setting for UIUCsports and scenes15 databases. Also, the nature of the images is to be considerate and a study of the heterogeneity level of images could be achieved [22] through the Shannon entropy measure. However, we think that our modeling can be improved by three ways. The first will be to get even better stabilized JSC results by adding an outer loop in the JSC algorithm. After multiple stages, we can expect some improvements. The second is a direct extension of the JSC by integrating some Laplacian regularization computed from a training set of local features. Here, sparse codes will be reconstructed by simultaneously minimize the deviation from both this training set and the image local features. The fusion could be improved by weighted average fusion using statistic from code image. Finally, it had been shown that adding some orthogonal constraints during the dictionary learning process can improves results [5, 17]. Here, too, a full study should be conducted with the two methods of sparse codes encoding.

Acknowledgement. We thank Direction Générale de l'Armement (DGA) for a financial support to this research. We thank Lucian ALECU for his comments.

References

1. C. Bauge, M. Lagrange, J. Andén, and S. Mallat. Representing environmental sounds using the separable scattering transform. In *ICASSP*, pages 8667–8671, 2013.
2. Y. Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, Jan. 2009.
3. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531, 2014.
4. S. S. Chen, D. L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.
5. A. Cherian. Nearest neighbors using compact sparse codes. In T. Jebara and E. P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML - 14)*, pages 1053–1061. JMLR Worshop and Conference Proceedings, 2014.
6. J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and Analysis of a Large Scale Image Ontology. Vision Sciences Society, 2009.
7. J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric ℓ_p-norm feature pooling for image classification. In *CVPR*, pages 2697–2704, 2011.
8. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

9. Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *ISCAS*, pages 253–256. IEEE, 2010.
10. H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.
11. L.-J. Li. What, where and who? classifying event by scene and object recognition. In *In IEEE International Conference on Computer Vision*, 2007.
12. D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.
13. J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA, 2009. ACM.
14. M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.
15. G. V. Pendse. A tutorial on the lasso and the "shooting algorithm". Technical report, P.A.I.N Group, Imaging and Analysis Group - McLean Hospital, Harvard Medical School, 8 February 2011.
16. F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg, 2010. Springer-Verlag.
17. I. Ramirez, F. Lecumberry, and G. Sapiro. Universal priors for sparse modeling. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pages 197–200, Dec 2009.
18. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2014.
19. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.
20. J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias. Local Sparse Coding for Image Classification and Retrieval. Technical report, 2012.
21. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
22. S. Tollari and H. Glotin. Lda versus mmd approximation on mislabeled images for keyword dependant selection of visual features and their heterogeneity. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume II, pages 413–416, may 2006.
23. X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu. Max-margin multiple-instance dictionary learning. In S. Dasgupta and D. Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 846–854. JMLR Workshop and Conference Proceedings, May 2013.
24. B. Xie, M. Song, and D. Tao. Large-scale dictionary learning for local coordinate coding. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.9. BMVA Press, 2010. doi:10.5244/C.24.36.
25. M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, and D. Cai. Graph regularized sparse coding for image representation. *IEEE Transaction on Image Processing*, 20(5):1327–1336, 2011.