Approximate Image Matching using Strings of Bag-of-Visual Words Representation

Hong-Thinh Nguyen, Cécile Barat, Christophe Ducottet

Approximate image matching using strings of bag-of-visual words representation

NGUYEN Hong Thinh[1], BARAT Cecile[1] and DUCOTTET Christophe[1]
[1]*Université de Lyon, F-42023, Saint-Étienne, France ;*
CNRS, UMR 5516, Laboratoire Hubert Curien, F-42023, Saint-Étienne, France ;
Université de Saint-Étienne, Jean-Monnet, F-42023, Saint-Étienne, France.
{*hong.thinh.nguyen, cecile.barat, ducottet* }*@univ-st-etienne.fr*

Keywords: Edit distance, string of histograms, bag-of-visual words, image classification

Abstract: The Spatial Pyramid Matching approach has become very popular to model images as sets of local bag-of-words. The image comparison is then done region-by-region with an intersection kernel. Despite its success, this model presents some limitations: the grid partitioning is predefined and identical for all images and the matching is sensitive to intra- and inter-class variations. In this paper, we propose a novel approach based on approximate string matching to overcome these limitations and improve the results. First, we introduce a new image representation as strings of ordered bag-of-words. Second, we present a new edit distance specifically adapted to strings of histograms in the context of image comparison. This distance identifies local alignments between subregions and allows to remove sequences of similar subregions to better match two images. Experiments on 15 Scenes and Caltech 101 show that the proposed approach outperforms the classical spatial pyramid representation and most existing concurrent methods for classification presented in recent years.

1 INTRODUCTION

Local feature histograms are widely employed to represent visual contents in various areas of computer vision. In particular, histograms of visual words based on SIFT features, in the well-known bag-of-words model, have proven to be very powerful for image classification or retrieval tasks (Sivic and Zisserman, 2003). However, such histograms only use a global information and discard spatial distribution of features. The trend in recent years is towards the use of a set of local histograms rather than only one to encode spatial information and provide richer representations. An image is partitioned into regions, using either a segmentation algorithm (Chen et al., 2009) or a division according to a grid pattern (Lazebnik et al., 2006; Battiato et al., 2009; Cao et al., 2010). Each region is then described with a local feature histogram.

In this context, the most significant work is certainly the "Spatial Pyramid Matching" approach (SPM), which gave excellent classification results on different image collections, as 15 Scenes and Caltech 101 (Lazebnik et al., 2006). SPM consists in dividing an image into predefined regular grids of different scales (1×1,2×2,4×4,..) and computing a bag-of words histogram in each cell. The different histograms are then ordered and combined with appropriate weights into a unique vector to form the final image representation. Two images are then compared using an intersection kernel. Since this approach was very efficient, it has received great research attention. The different aspects of the model have been investigated for the purpose of improving performance leading to systems that reach state-of-the-art results in the domain. For instance, some authors focus their attention on the coding of local visual features to improve the local bag-of-words representation (Yang et al., 2009a; de Avila et al., 2013). Sparse coding associated with max pooling have shown good results in (Yang et al., 2009a). Other works focus on optimizing the partitioning of images. In (Sharma and Jurie, 2011), Sharma et al. propose to learn the best discriminative grid splitting optimizing a given classification task. In (Viitaniemi and Laaksonen, 2009), Viitaniemi et al. compare techniques of soft tiling and hard tiling. Furthermore, some works propose to learn or adapt weights rather than using fixed ones, as in (Harada et al., 2011). All these kinds of approaches often associate the definition of new kernels for image comparison (He et al., 2008; Viitaniemi and Laaksonen, 2009; Harada et al., 2011).

Most of these SPM-based methods perform well

though they use rigid matching between corresponding regions limiting their invariance to geometric transformations. Indeed they assume that similar parts of a scene or an object generally lay in similar regions of the space. In the case of two images whose visual elements are located at different positions or have different extensions as on Figure 1, the matching fails while the visual content is quite similar. These methods actually compute an approximate global matching of the visual words among images. Our intuition is that computing an approximate matching of the histograms would make the method more robust.

One popular category of methods for approximate matching uses the edit distance. The standard edit distance is a string metric for measuring the difference between two sequences. It is defined as the minimum number of edit operations, *i.e.* insertion, deletion, substitution, which are required to transform one string to the other (Wagner and Fischer, 1974). It has the advantage of providing the matching of symbols located at different positions in the string taking into account the order of these symbols and some costs affected to each edit operation. Moreover, there exists an efficient calculation algorithm based on dynamic programming. In the image domain, this distance has been successfully used for text recognition applications (Seni et al., 1996; Christodoulakis and Brey, 2009; Khurshid et al., 2009) or shape matching (Klein et al., 2001). A key question is the representation of the visual content as strings.

In this paper, we propose a new method to represent images as strings of histograms and, to compare such representations, we introduce a string kernel that uses an extended edit distance tailored to the context of local histograms comparison. The histograms can correspond to local SIFT bag-of-words computed with recent developments as sparse coding. For each given pair of images, our distance not only takes into account the similarity between pairwise regions (with the substitution cost) as in the standard SPM model, but also integrates information about similarity between neighbouring regions (with the insertion/deletion cost). It allows to identify local alignments between subregions or groups of similar subregions in images. With the proposed approach, the number of subregions for different images may vary and is considered according to the visual content, which brings flexibility to the matching process in comparison with previous mentioned methods. We validate our approach on two well-known datasets: 15 Scenes and Caltech 101.

There has been some related work in the literature aiming to take into account topological relationships between regions (Iovan et al., 2012; Sharma and Jurie, 2011) or using an edit distance between strings of bag-of-words (Ballan et al., 2010). The work of (Yeh and Cheng, 2011) is the most similar to our approach. However, their representation is questionable. They build strings from the 16 histograms of the second level of a spatial pyramid, following a raster scan. Thus, consecutive histograms in the strings may have no spatial relationships. Moreover, they use fixed costs for insertion/ deletion operations which does not allow to adapt the image partitioning as we propose. Also, they use a single level of the pyramid while we combine different ones.

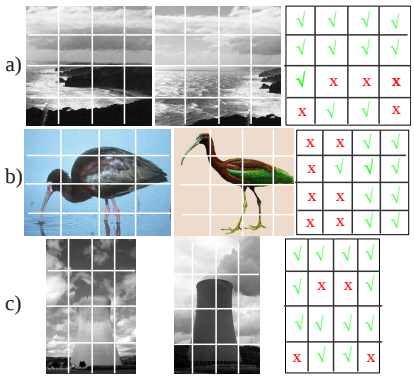


Figure 1: Examples of intra-class pairwise mismatches due to changes in viewpoint and moves of a scene component: (a) horizontal translation leading to additional sea regions and removal of land regions (b) move of bird's head leading to vertical displacement of similar regions on the left of the image (c) scale variation leading to additional sky regions replacing top chimney ones. Images are taken from 15 Scenes and Caltech 101 datasets.

The remainder of this paper is organized as follows. In section 2, we discuss the limits of the SPM model and present our model of images as strings of histograms. In section 3, we develop the edit distance adapted to strings of histograms and derive an edit kernel. Section 4 describes experiments and results of our edit distance on image classification tasks. We conclude in Section 5.

2 IMAGE REPRESENTATION

In this section, we discuss the limitations of pairwise matching using spatial pyramid scheme, then we introduce our image representation model.

2.1 Pairwise matching limitations

In SPM-based methods, a similar grid partitioning scheme is applied to all images. This approach is

not optimal to represent the image layout in two respects. First, using a predefined partitioning pattern, independently from the content, is sensitive to intra-class variations. Figure 1 illustrates changes in viewpoint, scale and a displacement of part of an image with level 2 of the spatial pyramid. These changes cause mismatches between pairwise regions due to local misalignments between images, while contents are quite similar. Note that such changes can often be seen as adding (or removing) regions similar to their neighbourood. For instance, in the case of Figure 1(a) and Figure 1(c), sea regions replace coast regions and sky regions replace the top of the nuclear chimney due to landscape continuity.

Second, partitioning images similarly along both directions may not always be the best strategy to describe the visual content. Indeed, in images, there exists a natural sequencing of objects or entities within objects. It is possible to find a principal direction along which the projection of local features may convey information about the image context or capture the essence of the form of an object. Intuitively, as suggested in (Cao et al., 2010), in natural scenes, vertical or horizontal directions can plausibly describe relationships among local features. For instance, the sky is above trees, and trees are above grass. For urban scenes, in (Iovan et al., 2012), the authors propose similarly to replace the SPM grid division with divisions along the vertical axis to better take into account the composition of this kind of images. For object images, as proposed in (Tirilly et al., 2008), the major axis of an object can be obtained from the first principal component in a principal component analysis. Distribution of local features along this major axis is similar whatever the orientation or scale the object is.

The graphs of Figure 2 highlight the two mentioned limitations of SPM on the 15 Scenes dataset. The classification accuracy is plotted with respect to the number of local regions, using either a grid partitioning or divisions along one axis, vertical or horizontal. Each region is described with a SIFT bag-of-words obtained following the protocol of (Lazebnik et al., 2006) and a vocabulary of 100 words. The classification accuracy was computed with intersection kernel SVM and 10-fold cross-validation. We observe that increasing the number of regions first improves the classification accuracy, but when the number of regions is too high, the accuracy decreases. It is explained by the fact that the number of mismatches is all the greater that the number of regions increases. Moreover, using a vertical directional partitioning gives higher results than a grid partitioning for this dataset composed mainly of natural scenes.

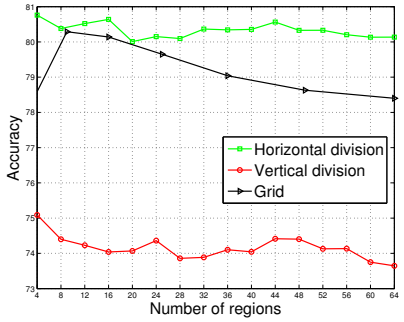


Figure 2: Classification accuracy versus number of local regions for the 15 Scenes dataset using pairwise matching and different partitioning schemes: grid, vertical divisions or horizontal ones.

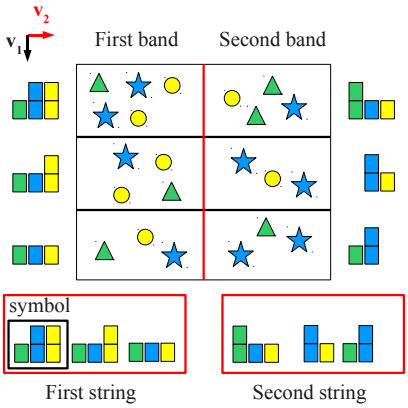


Figure 3: Example of an image representation as two strings of histograms.

2.2 Defining strings of histograms

We propose a representation of images based on strings of bag-of-words to better describe the intrinsic order of regions in a given direction. First, we choose an orthogonal basis $\mathbf{v_1}, \mathbf{v_2}$ that may best represent the image content. We divide an image into B bands of same width along direction $\mathbf{v_2}$. Then, each band is subdivided into N subregions of same size, along direction $\mathbf{v_1}$. For each band, traveling along $\mathbf{v_1}$ provides an ordered string of subregions. Figure 3 illustrates the construction of the strings associated to a two band case.

In this paper, we only consider the case of vertical and horizontal axes. The vertical direction being retained as the principal direction $\mathbf{v_1}$. Indeed, as shown after several evaluations over different datatsets, we found that the vertical direction always provides the best results (see Section 4). However, the method can be generalized to any other basis.

2.3 Pyramidal strategy

In practice, the number of bands B and the number of regions per band N is fixed for all images and determined according to the dataset used. Note that these parameters do not play the same role. Parameter B defines the main image partitioning and plays a similar role as the division parameters used in (Lazebnik et al., 2006). Thus, we can choose either to fix its value to a power of 2 (*e.g.* $B = 1, 2$ or 4) or to use a pyramidal strategy considering all the strings obtained for several splittings obtained with $B = 2^0, \ldots, 2^{L-1}$ where L is the number of pyramid levels. Parameter N defines the size of the strings. In our edit distance-based approach, this parameter is not too sensitive if it is large enough, as shown in Section 4.

3 AN EDIT DISTANCE FOR STRINGS OF HISTOGRAMS

In this section, we present an extended edit distance tailored to compensate for mismatches limiting performances of rigid matching approaches, as explained previously. We first recall the standard edit distance.

3.1 The standard edit distance

The standard edit distance allows to compute the optimal alignment of two strings. In its simple form, the edit distance between two strings $X = x_1 x_2 \ldots x_N$ and $Y = y_1 y_2 \ldots y_M$ is defined as the minimum cost of all sequences of edit operations which transform X into Y. The permitted edit operations with their associated cost functions are as follows:

- insertion of a symbol y_j into X with cost $c_{ins}(y_j)$
- deletion of a symbol x_i into X with cost $c_{del}(x_i)$
- substitution of a symbol x_i with the symbol y_j with cost $c_{sub}(x_i, y_j)$

Computing this distance can be formulated as an optimization problem and can be carried out with a dynamic programming algorithm. The algorithm consists in computing a $D(N,M)$ matrix, where $D(i,j)$ represents the minimum cost of transforming $X = x_1 x_2 \ldots x_i$ into $Y = y_1 y_2 \ldots y_j$, with allowable edit operations mentioned above. The computational complexity is proportional to the product of the length of the two strings, *i.e.* in $O(N \times M)$. The computation is carried out using the following recurrence relation:

$$\begin{cases} D_{0,0} = & 0 \\ D_{0,j} = & D_{0,j-1} + c_{ins}(y_j), \quad j = 1 \ldots N \\ D_{i,0} = & D_{i-1,0} + c_{del}(x_i), \quad i = 1 \ldots M \\ D_{i,j} = \min\big(& D_{i-1,j} + c_{del}(x_i), \\ & D_{i,j-1} + c_{ins}(y_j), \\ & D_{i-1,j-1} + c_{sub}(x_i, y_j)\big), \\ & \quad i = 1 \ldots M, j = 1 \ldots M \end{cases} \tag{1}$$

3.2 A new string matching distance

In our approach, symbols are histograms of visual words. Let us recall that our aim is to compute an approximate matching between strings of histograms in order to correct mismatches due to fixed grid partitioning seen in Section 2.1. By definition, the edit distance aims to find the optimal alignment between two strings, and thus allows naturally to correct local or global misalignment due to translation of viewpoint modifications between two images. An immediate strategy is to use fixed costs for insertion and deletion and a ground distance between histograms for substitution. To go further, we propose to adapt insertion and deletion costs to the local context. The goal is to virtually adjust the grid partitioning during the image comparison and compensate for mismatches that occur with homogeneous parts of a scene or object splitted in different regions. Our approach is to use deletions and insertions to get rid of repetitions of similar symbols respectively in the input string (deletion) or in the output string (insertion), relatively to the other string. More precisely, during the alignment of the two strings, if one symbol is more similar to its following than to the corresponding one in the other string, it will be removed. Formally, this rule comes to define costs functions as:

$$c_{sub}(x_i, y_j) = d(x_i, y_j) \tag{2}$$

$$c_{del}(x_i) = d(x_i, x_{i+1}) \tag{3}$$

$$c_{ins}(y_j) = d(y_j, y_{j+1}) \tag{4}$$

where d is any histogram distance. We use ℓ_1 distance in the following.

The new edit distance is then computed by transfering these specific cost functions into the original dynamic programming algorithm.

3.3 Examples

We first illustrate our string matching distance with a toy example (Figure 4). This example simulates

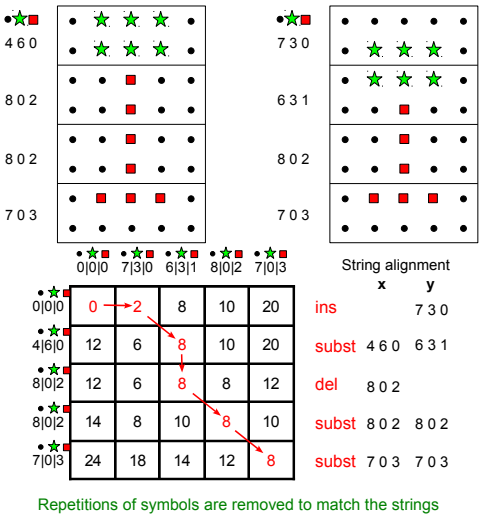


Figure 4: A toy example to illustrate the matching using *SMD* for the single band case.

a viewpoint change similar to the real case of Figure 1(c). The distance matrix gives minimum distances $D_{i,j}$ and arrows show the sequence with minimum cost, detailed on the right. To well understand the values, we detail calculations of three cells. First, the cell $D_{0,1}$ equal to 2 gives the insertion cost of symbol $7|3|0$, *i.e.* $d(7|3|0,6|3|1)$, while the cell $D_{1,0}$ is the deletion cost of symbol $4|6|0$, i.e. $d(4|6|0,8|0|2)$. The value of $D_{1,1}$ is the minimum of $D_{0,1}+d(4|6|0,8|0|2)$, $D_{1,0}+d(7|3|0,6|3|1)$ and $D_{0,0}+d(4|6|0,7|3|0)$, i.e. $min\{14,14,6\}=6$. As for the computation of $D_{1,1}$, each minimum distance takes into account similarity between neighbouring regions and direct pairwise similarity between corresponding regions, allowing to remove repetitions of symbols when necessary to adapt to the other string. In our toy example, the resulting edit sequence comes to consider the two similar regions $8|0|2$ as a unique one that matches the similar one in the second image. We now give a real-case example seen in Figure 1(b). As shown on Figure 5, two bands of four regions are used. The string matching sequence obtained for the first band is given showing a better alignment than with direct pairwise matching. Here, insertions and deletions enable to deal with a change of position of the head of the bird. In each case, region matchings that we have drawn correspond to the real computed edit scripts. These examples confirm the interest of our approach to better deal with possible changes in object size, position or shape in the direction of the string.

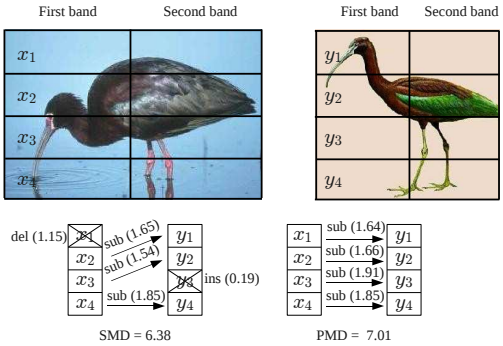


Figure 5: Real example of *SMD* matching for a two-band case.

3.4 Image comparison kernel

To be able to use a standard Support Vector Machine (SVM) algorithm for image classification, we define two kernels for measuring the similarity of two images with our edit distance: one for the single level image representation and one for the pyramidal one.

When considering one level of a pyramid with B bands of N symbols, the comparison of two images amounts to compute the distance between every 2-by-2 corresponding strings and sum the B distance results. This distance is denoted d^B_{SMD}. In the pyramidal case, several levels of splitting are combined using a weighted sum of d^B_{SMD} distances leading to the L levels pyramidal SMD distance $d^{P(L)}_{SMD}$. Formally, these distances between two images I and $\mathcal{J}$ are given by:

$$d^B_{SMD}(I,\mathcal{J}) = \frac{1}{BN}\sum_{b=1}^{B} d_{SMD}(\mathbf{x^B_b}(N),\mathbf{y^B_b}(N)) \quad (5)$$

$$d^{P(L)}_{SMD}(I,\mathcal{J}) = \sum_{B\in\{2^0,\ldots,2^{L-1}\}} \alpha_B\, d^B_{SMD}(I,\mathcal{J}) \quad (6)$$

where $\mathbf{x^B_b}(N)$ (resp. $\mathbf{y^B_b}(N)$) represents the bth string of the B bands splitting of image I (resp. image $\mathcal{J}$) and α_B are the weighting coefficients, chosen to be here those proposed by (Lazebnik et al., 2006).

Applying these two distances in the classical string edit kernel (Li and Jiang, 2005) leads to the following kernels:

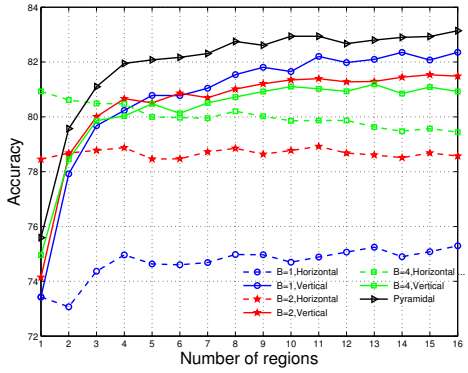
$$K^B_{SMD}(I,\mathcal{J}) = e^{-\gamma d^B_{SMD}(I,\mathcal{J})} \quad (7)$$

$$K^{P(L)}_{SMD}(I,\mathcal{J}) = e^{-\gamma d^{P(L)}_{SMD}(I,\mathcal{J})} \quad (8)$$

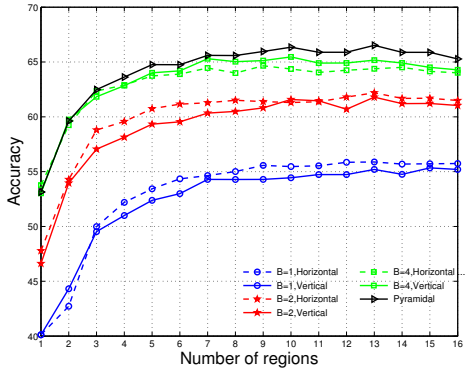
where γ is a scaling coefficient chosen to ensure the admissibility of the kernel for a given dataset.

4 RESULTS

In this section, we report experimental results on two popular datasets: 15 Scenes and Caltech 101.



(a) 15-scene



(b) Caltech-101

Figure 6: Classification performances for 15 Scenes and Caltech 101 datasets with SMD under different number of regions, number of bands and scanning directions.

The motivation of these experiments is twofold: first we study the influence of the parameters of our image representation model on the classification accuracy. Second, we evaluate our edit matching method against the pairwise matching approach, spatial pyramid matching and other concurrent methods.

As mentioned previously, in experiments, the local bag-of-words are computed as in (Lazebnik et al., 2006). For classification, we apply a SVM classifier using libSVM in a one-vs-all setting. Kernel matrices are computed either with our edit kernels presented in Section 3.4 or the histogram intersection kernel for comparison purpose. With Caltech 101, we chose randomly 30 images per class for training and up to 50 images per class for testing. With 15 Scenes, we train on 100 random images per class and test on remaining ones. Each experiment was repeated 10 times with randomly selected training and testing sets. The performance of all experiments are evaluated by the mean accuracy over the 10 runs.

4.1 Influence of string parameters

In our string based representation model, several parameters have to be set to compute classification results: the number of bands and the scanning direction, the number of regions and the size of the vocabulary. In this section, we study the influence of these parameters.

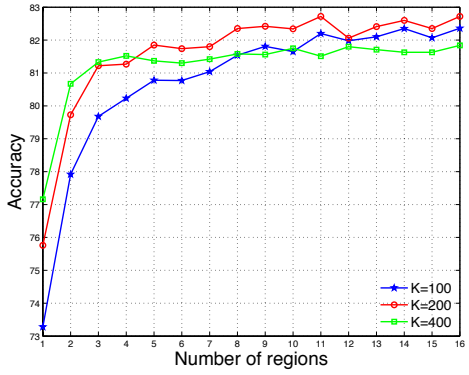
The classification accuracy is first computed with respect to the number of regions per band from 1 to 16, varying the number of bands ($B = 1$, $B = 2$ and $B = 4$) and the scanning direction (vertical or horizontal). The vocabulary is then fixed to 100 words. Results are presented on Figure 6.

Scanning direction. The results for the two data sets are different. For 15 Scenes dataset, all vertical case graphs are above horizontal case ones. These results confirm the intuition that the vertical direction in natural scenes provides a better characterization of the image structure than the horizontal one, as discussed in Section 2.2. For Caltech 101 dataset, differences between the horizontal and vertical case graphs are not that significant. It can be explained by the specificity of this collection where the objects of interest take up most of the image and are approximately centered. Comparing objects along one or the other direction does not really matter. Since on the overall the vertical direction performs better, we keep this direction in all subsequent experiments.

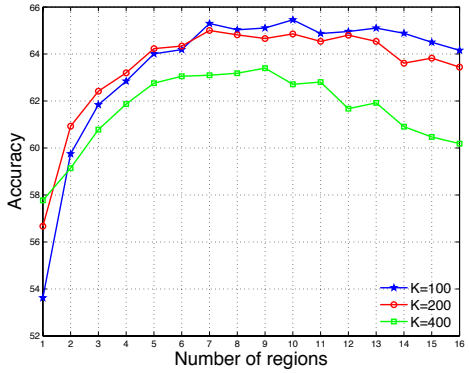
Number of bands. Again, the results depend on the collection. Indeed, considering the 1, 2 and 4 bands cases, the results behave inversely. For 15 Scenes, results decrease as the number of bands increase, *i.e.* one band is enough to get the best results while for Caltech 101, it is preferable to use four bands. As previously, it is inherent to the type of images. Observing a natural scene from top to bottom allows to identify the content. Using two parallel vertical bands does not convey much information. It even introduces confusion because of redundancy between bands, leading to worst results. For objects, a finer look at the different parts is necessary to identify them correctly.

It is worth noticing that for both datasets, a 2-level pyramid approach clearly outperforms single level splitting cases. This strategy is suitable to get the best results at the cost of a higher dimension representation.

Number of regions. Considering the influence of the number of regions, the global evolution of all curves is similar: the accuracy is almost monotonically increasing with a stabilization for $N = 8$ for Caltech 101 and $N = 10$ for 15 Scenes. From this value, the results remain roughly constant or slightly better. The highest accuracy is 66.53 % achieved with $N = 13$ for



(a) 15-scene

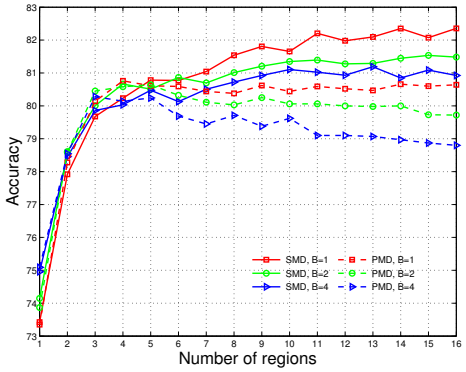


(b) Caltech-101

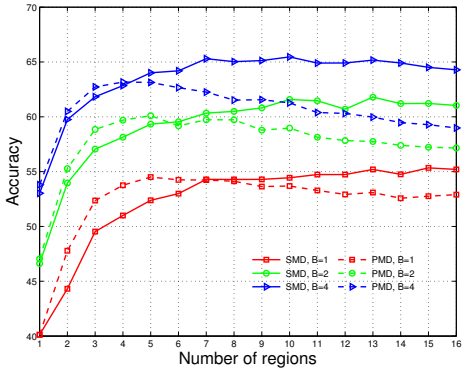
Figure 7: Influence of the vocabulary size

Caltech 101, and 83.16 % with $N = 16$ for 15 Scenes. But since performances were quite similar with lower number of regions, it is preferable to use $N = 8$ or $N = 10$ to reduce the computation time.

Vocabulary size. To investigate the effect of the vocabulary size, we fix the number of bands to the optimal values obtained previously, *i.e.* $B = 4$ for Caltech 101 and $B = 1$ for 15 Scenes. Figure 7 shows the classification results for three vocabulary sizes 100, 200 and 400 as a function of N. For Caltech 101, the best results are obtained with the smallest vocabulary ($k = 100$) and we note that the accuracy is decreasing over N for $k = 200$ or $k = 400$. For 15 Scenes, the influence of the vocabulary size is low and the results are slightly better for $k = 200$, but they are very close to $k = 100$ for a large N. Obtaining the best results for small vocabularies is unusual in the BoVW context. This very interesting behavior is due to the properties of the edit distance: to benefit from insertions and deletions, we must have enough similar symbols and thus no too large histograms. Thus, a clear advantage of our string based approach is to provide a more compact representation capable to exploit the spatial distribution of the visual information. In the following experiments, we will keep a vocabulary of 100 visual words.



(a) 15-scene



(b) Caltech-101

Figure 8: Performance of *SMD* vs *PMD*

4.2 Spatial vs pairwise matching

Here, experiments are to verify the performance improvement by our string matching approach (*SMD*) over a classic pairwise matching approach (*PMD*) when using a same partitioning. Figure 8 presents the results, still keeping the optimal number of bands for each dataset.

First, it is obvious that for both datasets, *SMD* is always above *PMD* for $N > 5$ and any given splitting. As seen in Section 2.1, the greater the number of regions, the greater the number of local mismatches, leading to a decrease of performances of a pairwise matching approach. With *SMD*, for large N, the accuracy stabilizes (Caltech 101) or slightly increases (15 Scenes). It proves that *SMD* naturally compensates the local mismatches.

Note that our method *SMD* achieves better results than those reported by Lazebnik et al (Lazebnik et al., 2006) for the same datasets. Indeed, for a 200 vocabulary size, we get 65.1 vs 64.6 for Caltech 101 and 82.0 vs 81.1 for 15 Scenes. These results are obtained respectively for $N = 7$ and $N = 14$ with only 100 visual

Table 1: Comparison of our approach over concurrent methods based on SIFT and k-means. The size of the codebook is given in brackets. We report the highest values obtained in pyramidal case only. − means there is no result available.

Method	Caltech 101	15 Scenes
SPM (pyr., K=100)	63.2 [100]	80.1 [100]
SPM (best pyr. result)	64.6 [200]	81.4 [400]
SPM+co-occurrence	-	82.51 [200]
Sequence matching	-	80.9 [200]
SPM+ spatial partition learning	-	80.1 [1000]
SMD	**66.5 [100]**	**83.2 [100]**

words which gives a much more compact representation. Again, the matching is improved thanks to the proposed insertion and deletion operations used in our string edit distance.

4.3 Comparison with existing methods

In Table 1, the proposed method is first compared with the concurrent techniques that use a single SIFT descriptor and the original BoVW coding, *i.e.* BoVW histogram (sum pooling) with a hard assignment for visual words for fair comparison. These methods are the original SPM method, SPM + co-occurrence (combination of SPM and the spatial relationship information between visual words inside each image (Yang and Newsam, 2011), sequence matching (Yang et al., 2009b) and optimal spatial partitioning (Sharma and Jurie, 2011). The table shows that for both datasets, our approach clearly outperforms all other methods. It is important to note that the best result is obtained with the smallest vocabulary of 100 words.

Also, to compare *SMD* with recent works based on sparse coding to create the vocabulary, we have integrated sparse coding in our method. For this, we use the Matlab code ScSPM from authors of (Yang et al., 2009b) , and following (Boureau et al., 2010), we use the max pooling to compute local BoVW due to it better performance than average pooling. We compare to methods Sc-SPM (Yang et al., 2009b) and Kernel Sparse Representation (KSR-SPM) (Gao et al., 2010). The Sc-SPM approach can be treated as spatial pyramid matching method using sparse coding. The KSR-SPM approach is the combination of SPM with a kernel sparse representation technique. Our method definitely outperforms both of them.

Table 2: Comparison with sparse coding based methods. − means there is no result available.

Method	Caltech 101	15 Scenes
ScSPM [1024]	73.2 ± 0.5	80.28 ± 0.9
KSR-SPM [1024]	-	83.68 ± 0.61
SMD [100]	$\mathbf{73.44 \pm 1.1}$	$\mathbf{84.59 \pm 0.7}$

5 CONCLUSION

In this paper, our contribution is twofold. First, we describe a novel image representation as strings of histograms which encodes spatial information, each histogram being a BoVW model of a subregion. Second, we introduce a new edit distance able to automatically identify local alignments between subregions and the removal of sequences of similar subregions. This characteristic makes our method more robust to translation or scale variations of objects in images than SPM-based approaches that compare rigidly corresponding parts of images.

The experiments confirm that our model is able to take into account spatial relationships between local BoVW and leads to a clear improvement of performance in the context of scene and image classification compared to the classical spatial pyramid representation. It is worth noticing that to the best of our knowledge, it is the first time that results better than SPR are reported with the standard BoVW coding and a lower dimension for the representation. Moreover, the proposed approach obtain similar or better accuracies than other recent methods trying to infuse spatial relationships into the original BoVW model with the great advantage of using a small codebook and a compact representation. In the future, we are interested in extending our edit distance to other data structures such as trees. Trees are indeed often used to represent image content, and some edit distances already exist.

REFERENCES

Ballan, L., Bertini, M., Del Bimbo, A., and Serra, G. (2010). Video event classification using string kernels. *Multimedia Tools and Applications*, 48(1):69–87.

Battiato, S., Farinella, G., Gallo, G., and Ravì, D. (2009). Spatial hierarchy of textons distributions for scene classification. In *Proceedings of the 15th International Multimedia Modeling Conference on Advances in Multimedia Modeling*, MMM '09, pages 333–343, Berlin, Heidelberg. Springer-Verlag.

Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE.

Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). Spatial-bag-of-features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3352–3359. IEEE.

Chen, X., Hu, X., and Shen, X. (2009). Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 867–874, Berlin, Heidelberg. Springer-Verlag.

Christodoulakis, M. and Brey, G. (2009). Edit distance with combinations and splits and its applications in ocr name matching. *International Journal of Foundations of Computer Science*, 20(06):1047–1068.

de Avila, S. E. F., Thome, N., Cord, M., Valle, E., and de Albuquerque Araújo, A. (2013). Pooling in image representation: The visual codeword point of view. *Computer Vision and Image Understanding*, 117(5):453–465.

Gao, S., Tsang, I. W.-H., and Chia, L.-T. (2010). Kernel sparse representation for image classification and face recognition. In *Computer Vision–ECCV 2010*, pages 1–14. Springer.

Harada, T., Ushiku, Y., Yamashita, Y., and Kuniyoshi, Y. (2011). Discriminative spatial pyramid. In *CVPR*, pages 1617–1624. IEEE.

He, J., Chang, S.-F., and Xie, L. (2008). Fast kernel learning for spatial pyramid matching. In *CVPR*. IEEE Computer Society.

Iovan, C., Picard, D., Thome, N., and Cord, M. (2012). Classification of Urban Scenes from Geo-referenced Images in Urban Street-View Context. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 339–344, États-Unis.

Khurshid, K., Faure, C., and Vincent, N. (2009). A novel approach for word spotting using merge-split edit distance. In *Computer Analysis of Images and Patterns*, pages 213–220. Springer.

Klein, P. N., Sebastian, T. B., and Kimia, B. B. (2001). Shape matching using edit-distance: an implementation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 781–790. Society for Industrial and Applied Mathematics.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006, IEEE Computer SocietyConference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178. IEEE.

Li, H. and Jiang, T. (2005). A class of edit kernels for svms to predict translation initiation sites in eukaryotic mrnas. *Journal of Computational Biology*, 12(6):702–718.

Seni, G., Kripasundar, V., and Srihari, R. K. (1996). Generalizing edit distance to incorporate domain information: Handwritten text recognition as a case study. *Pattern Recognition*, 29(3):405–414.

Sharma, G. and Jurie, F. (2011). Learning discriminative spatial representation for image classification. In *Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference, pages*, pages 6–1.

Sivic, J. and Zisserman, A. (2003). 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.

Tirilly, P., Claveau, V., and Gros, P. (2008). Language modeling for bag-of-visual words image categorization. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, CIVR '08, pages 249–258, New York, NY, USA. ACM.

Viitaniemi, V. and Laaksonen, J. (2009). Spatial extensions to bag of visual words. In *CIVR*.

Wagner, R. and Fischer, M. (1974). The string-to-string correction problem. *J. ACM*, 21(1):168–173.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009a). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009b). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE.

Yang, Y. and Newsam, S. (2011). Spatial pyramid co-occurrence for image classification. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1465–1472. IEEE.

Yeh, M.-C. and Cheng, K.-T. (2011). Fast visual retrieval using accelerated sequence matching. *Multimedia, IEEE Transactions on*, 13(2):320–329.