A Bag of Strings representation for Image Categorization

Julien Ros, Christophe Laurent, Jean-Michel Jolion

A Bag of Strings Representation for Image Categorization

Julien Ros[1], Christophe Laurent[2], and Jean-Michel Jolion[1]

[1] Université de Lyon - LIRIS - UMR CNRS 5205 - INSA Lyon - 69621 Villeurbanne Cedex, FRANCE {julien.ros|jean-michel.jolion}@liris.cnrs.fr

[2] France Télécom, RO&SI/DSIS/SIFAC, 15, avenue Léonard de Vinci, 33608 Pessac, FRANCE christophe2.laurent@orange-ftgroup.com

Abstract. This paper presents an architecture well suited for natural images classification or visual object recognition applications. The method proposes to integrate a spatial representation into the well known "bag of local signatures" approach. For this purpose, it combines the power of a string representation which provides an ordered view of local features with the vectorial histogram representation allowing to recognize efficiently and quickly an image by using a machine learning classifier. To reach this goal, we propose to represent an image by a set of strings of local signatures obtained by tracking the detected salient points along image edges. We propose here to conjointly use the Hölder exponents and the direction of minimal regularity of the bidimensionnal signal singularities to compute a signature describing precisely a region of interest centered on an interest point. As we will see, an alphabet of strings is easily obtained by using a typical self organizing map architecture. As a consequence, a "bag of strings" representation is used, providing a compact representation encoding both local signatures and spatial information. This representation is particularly well suited to train a support vector machine classifier used for the last classification step. This architecture obtains good classification rates on different well known datasets.

1 Introduction

With the increasing availability of digital images shot by mobile phones and digital cameras (Infotrends[3] expects that the number of images captured on camera phones will reach 227 billion by 2009), efficient image management solutions should be built. Their goal is to group images into semantic categories giving thus the opportunity of fast and accurate image search. Until now, most of these systems use textual information to index images (as Google images system does) and as a consequence, it imposes a human intervention not conceivable in the case of large databases. Using computer vision algorithm can greatly help and it is why providing efficient algorithms for image content analysis applications is one of the most important research field in the computer vision area. Nevertheless, analyzing an image is difficult because it is a very complex structure. Indeed, images from the same categories can be very different in term of geometric transformations, illumination conditions, resolution or size leading to the difficulty to classify an image into a unique category.

Since 90's, a lot of methods have been proposed in order to be integrated in suitable solutions and they can be divided into two categories:

1. Content-based image retrieval (CBIR) systems were the first applications proposing solutions to the problem of searching for digital images in large image databases [49]. For this purpose, they first index each image in a database by a feature vector representing low level features. This vector permits to reduce the quantity of information to be later processed by discarding irrelevant information contained in each image. When a user queries the system with an image, the application searches within the whole database for similar images thanks to the feature vectors by using some similarity measures. However, CBIR systems are very inconclusive in results because there

[3] http://www.infotrends-rgi.com/home/Press/itPress/2005/1.11.05.html

is an important difference between what the user wants and what the feature vectors really describe. This is the well known *semantic gap* problem.

2. Supervised natural image classification systems are now more and more studied because they are more efficient due to the introduction of a statistical learning step. This one allows to deduce the feature vector components that should be used to discriminate the semantic concept interesting for the user and described by a training dataset. In spite of the limited vocabulary of the training dataset and of the different semantic concept within an image, supervised natural image classification systems were first successfully applied to group images into a limited number of semantic meaningful categories and recent works [47, 33] have shown their ability to capture several concepts within one image allowing their use for semantic labelling and automatic annotation.

This article addresses the problem of supervised natural image classification consisting of automatically recognizing scenes and object categories in natural photographs. In practice, a supervised image classification solution requires three main steps [8]: pre-processing, feature extraction and classification. Based on this architecture, many image classification systems have been proposed, each one distinguished from the others by the method used to compute the image signature and/or the decision method used in the classification step. Regarding the signature computation, the most efficient methods are probably the local approaches firstly introduced in [46]. In this case, local signatures are computed around some interest points and their values are chosen in a dictionary obtained from the training database. Once the image is locally described, two major problems should be underlined. First, the inherent local representation results in a lack of ordering between signatures and consequently fundamental information about the image content is lost. Secondly, it is difficult to directly classify this

orderless representation because the image is no longer considered as a vector. To overcome these disadvantages, different solutions have been proposed in the literature.

The first one consists in representing the image as a string of local signatures by adding an implicit order between salient points and then to compare them with classical string-based distances [40, 41, 24]. Although it is an interesting approach, it does not allow to embed an image into a vectorial space due to the non-vectorial nature of the string representation. Consequently, the only classifier that can be used is a k-nn classifier but its high computing time restricts the approach to small datasets.

A second solution is to represent the image by a distribution of local image features easily classifiable as in [6, 25]: this is the "bag of local signatures" representation and its computation is directly inspired from the text categorization domain. This analogy can be justified by observing that image classification using local approaches are closely related to the text categorization problem when local signatures have been quantized into visual keywords. These visual keywords have the ability to summarize well the image content providing thus a compact representation. The advantage is that by embedding images into a high dimensional vectorial space, this representation is well suited to be processed by a fast machine learning classifier instead of a computationally expensive k-nn algorithm. These methods have been successfully applied all along the different PASCAL challenges to detect and categorize images in a difficult datasets. However, it has been emphasized that this monodimensional representation does not consider any spatial information between subregions omitting an important quantity of information about the image content.

As a consequence, an interesting way investigated in this paper consists in exploiting the complementary capabilities of both approaches by merging them.

It is not the first attempt in the computer vision community to integrate spatial relation between local signatures or to consider the position of them when computing a model subsequently used for the last classification step. However, while most approaches integrate the mutual relations between interest points directly into a probabilistic part-based model [1, 34, 5, 16], some other encode locations of the interest points in the model [4, 50, 15, 29, 13, 12]. Finally the last approaches proposed in the literature describe coocurences between local features [31] or subdivide the image into several regions [32]. Excepting the recent work of [11], none attempt has been made to incorporate spatial relations between local signatures or their positions within the image in a "bag of salient features" framework. This is due to the fact that a structure easily identifiable and quantizable should be defined in order to compute a kind of "bag of structures" representation. Whereas, the definition of a structure can be easily done by using a graph or a string approach, the design of a dictionary of graphs (or strings) is not straightforward.

We propose in this paper to use a self organizing map for structured data [20] to produce a large dictionary of strings. This dictionary can be subsequently used to describe images by "bag of strings" representations. Finally, this representation is employed to learn a support vector machine classifier. The solution presented have been tested on three well know datasets described in Appendix A. The SIMPLICTY and Scene datasets contain images of natural scenes whose background information is essential to recognize the category. The PASCAL dataset is composed of object images with useless background for the categorization. The novelty of our work is (1)to propose a method to extract string of local feature vectors from an image; (2)to use a self organizing map for structured data [20] to produce a large dictionary of strings; (3)to employ this dictionary to embed images into "bag of strings" representations well suited to train a support

vector machine classifier and (4) to compare the method proposed with state of the art approaches of supervised image classification.

Paper Organization This paper is organized as follows: Section 2 describes the method which was introduced earlier in [27] to detect interest points. This section ends by describing the descriptor that characterizes both orientation and regularity of the singularities in a region of interest. The construction of strings is presented in section 3 and the clustering algorithm used to generate a dictionary of strings is deeply described in section 4. Section 5 presents the method used to generate the final feature vector representing the image. Experiments testing this approach with a support vector machine classifier are presented in section 6 and finally section 7 concludes the paper.

2 Local Features Extraction

The goal of feature extraction is to reduce the amount of data contained in an image by extracting relevant and discriminating features. In local approaches, this extraction phase results in feature vectors computed around interest points and an image I is thus represented by a set of local signatures $S(I) = \{s(1), \ldots, s(n)\}$. It is important to mention here that local approaches result in a lack of ordering between signatures.

2.1 Interest Points Detection

The goal of interest point detectors is to find image locations that are perceptually relevant for the next recognition step. Many detectors have been proposed in the literature, each one focusing on a particular local property of the image content such as contrast [3], corners [19, 17, 38], edges [27, 30], etc.

The salient points detector presented in [27] uses a wavelet analysis in order to find relevant pixels located on sharp region boundaries. The use of wavelet analysis is motivated by observing that multi-resolution, orientation and frequency analysis are of prime importance for the human visual system during the recognition step. This detector has proven its efficiency in many vision applications [27] and thus will be used in the present work.

Finally, it is important to note that the number of points to be detected can be automatically tuned by using an energy threshold. It allows to consider only interest points with large saliency value obtained from the wavelet coefficients. Consequently, it is possible to extract a number of salient points which depends on the complexity of the image content.

2.2 Description of Local Singularities

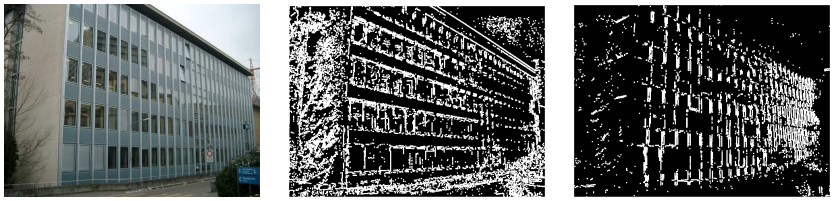
Most local descriptors describe the local neighborhood of salient points by characterizing edges in this area. Edge information thus appears fundamental in the process of local neighborhood description. To describe edges, gradient orientation and magnitude are generally used. Nevertheless, from a mathematical point of view, an edge or more generally a singularity can also be efficiently characterized by considering its Hölder exponents. We propose to use this mathematical notion to design our local descriptor.

Definition 1. *$f : [a, b] \to \mathbb{R}$ is Hölder $\alpha \geq 0$ at $x_0 \in \mathbb{R}$ if $\exists K > 0$, $\delta > 0$ and a polynom P of degree $m = \lfloor \alpha \rfloor$: $\forall x$, $x_0 - \delta \leq x \leq x_0 + \delta$, $|f(x) - P(x - x_0)| \leq K|x - x_0|^\alpha$.*

Definition 2. *The Hölder exponent $h_f(x_0)$ of f at x_0 is the superior bound value of all α. $h_f(x_0) = sup\{\alpha$, f is Hölder α at $x_0\}$.*

The local regularity of a function at a point x_0 is thus measured by the value $h_f(x_0)$. It is worth noting that the smaller $h_f(x_0)$, the more singular is the signal

at the point considered. For example, the Hölder exponent of a Dirac impulse is -1 and 0 for a step function. For an image, the Hölder exponent is measured in the direction of the minimal regularity of the singularity (in the gradient direction). The different singularities met in an image are shown on figure 1.



Original Image $-0.2 \leq h \leq 0.2$ $-1.2 \leq h \leq -0.8$

Fig. 1. different kind of singularities

To describe an ROI associated to an interest point in an image I, both orientation and Hölder regularity of singularities contained in that ROI are characterized. For this purpose, orientation $\theta(x, y)$ and gradient magnitude $m(x, y)$ at each pixel location (x, y) of the ROI are first computed. Then, for each singularity, the Hölder exponent h is estimated with foveal wavelets as presented in [35]. Orientations and Hölder exponents maps are then conjointly used to construct different 3D histograms. To build such histograms, each ROI is first partitioned into 4×4 blocks and each histogram is computed in a particular block before being normalized by the block size (See figure 2). This last step of the signature design is realized in the same spirit as the construction of the SIFT descriptor presented in [28].

Finally, the signature is obtained by concatenating the different 3D histograms and thus has a size of $n \times r \times o$ where n is the number of subregions, r is the number of Hölder exponents bins into the range $[-1.5, 1.5]$ and o is the number of orientations bins into $[-\frac{\pi}{2}, \frac{\pi}{2}]$. We typically use 8 orientations, 16 subregions and 3 Hölder exponents bins resulting in a signature size of 384. Indeed,

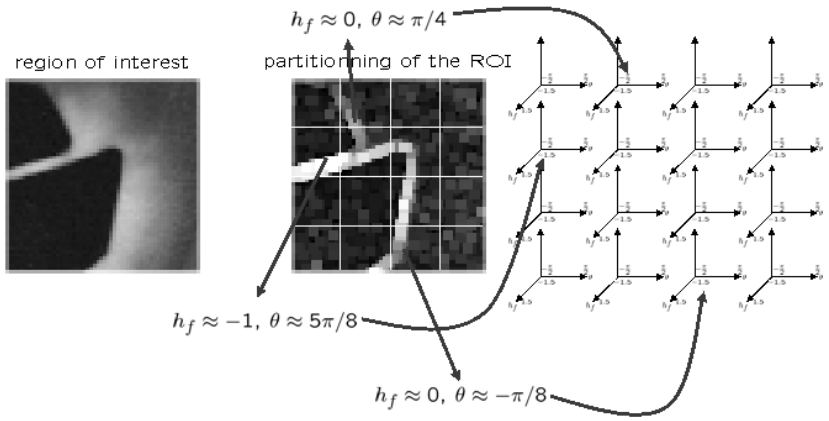


Fig. 2. principle of the singularity descriptor

with such a parameterization, previous experiments [43] have shown that this descriptor is better than classical SIFT and PCA-SIFT descriptors in the case of an image registration application.

It is worth noting that we have chosen to not select a representative scale for each keypoint when computing the descriptor. Indeed, there are currently some algorithms showing that using interest points is as efficient as using random points [37] due to the large training that can anyway capture the relevant information. In the same way, we have observed that using orientation and scale normalization is not shown crucial because the relevant information is also captured in the training step without this computational stage.

3 String Construction

In this section, we keep the local approach in which an image is still represented by a set of local signatures $S(I) = \{s(1), \ldots, s(n)\}$. Now, we focus on grouping some of these signatures in order to represent an image by a set of strings. The difficulty resides in the construction of these strings. If we consider the major

psycho-visual work of Biederman presented in [2] called the theory of recognition by components, the human decomposes an object into a subset of elementary cones called geons (geometry ions). These ones are the boundary fragments of the object to be recognized and thus the recognition is performed by assembling these different components.

Moreover, in [39], the authors present a framework in the field of object detection. The proposed architecture represents an object by a set of boundary fragments used for the next detection step.

Consequently, we have decided to generate strings from the set of salient points by tracking edges. The process is composed of three major steps:

1. computation of the minimal regularity direction for each salient point. This direction is used to group salient points that are on the same edge;
2. salient point tracking algorithm where the salient point strings are constructed;
3. suppression of small strings in the images which are considered as noise.

3.1 Direction of Minimal Regularity Computation

Salient points considered in this work are located on sharp region boundaries due to the use of a wavelet analysis to detect them. For each salient point (x_0, y_0), a direction of minimal regularity, say $\theta(I, x_0, y_0)$, exists and is the gradient direction at this point. To compute this direction, we used Sobel filters. The convolution of the image by these kernels is then used to compute the direction of minimal regularity.

On figure 3, we illustrate the direction of minimal regularity computed from salient points. In this figure, an energy threshold of 0.3 is used to detect salient points. It can be observed that salient points are located on the singularities and that the directions of minimal regularity are correctly computed with Sobel

(a) (b) (c)

Fig. 3. original image (a) salient points (b) directions of minimal regularity

filters. Moreover, nearby salient points sharing the same edge have closed directions of minimal regularity. As a consequence, it can be interesting to use this fact to design a method to group them in an edge tracking algorithm fashion explained in next section.

3.2 Salient Points Tracking

In this section, we present a method that aims at chaining the detected salient points to build a set of strings. The proposed method links points if they are sufficiently close in the image space and if the directions of the associated min-

imal regularities are also sufficiently close. This algorithm is inspired from the work presented in [36] aiming at tracking edge pixels.

This algorithm has two parameters:

- The first NB_NEIGHBORS is used to choose the number of nearest neighbors that should be considered to link salient points. Indeed, when a point is processed, the candidates that can be linked with it are chosen among its NB_NEIGHBORS closest neighbors in the image.
- The second parameter ANGLE_THRESHOLD defines the maximum value authorized between the two directions of normal regularity to link salient points.

Figure 4 illustrates our method. In this figure, two regions are represented and salient points are drawn as circles. First and without loss of generality, the algorithm starts with p_1 (cf. figure 4(a)) and it searches the first neighbor which is p_2 and compare the direction of minimal regularity and the normal of the segment $[p_1, p_2]$. If the comparison exhibits a small angle difference (below a threshold denoted ANGLE_THRESHOLD), the points are linked (cf. figure 4(b)). Next, the point p_2 is considered and with the same process explained above, the point p_3 is added to the string (cf. figure 4(c)). p_3 is then considered, but its neighbors are not compatible with regard to the angle tolerance. As a consequence, p_3 is an extremity of the string and p_1 is considered another time. p_4 is the next closest neighbor of p_1 not already processed and compatible with p_1, resulting in its concatenation at the beginning of the string (cf. figure 4(d)). Finally, on figure 4(e), the first string is completely built and other points can be considered to construct new strings. The final image is shown on figure 4(f) and contains two strings S_1 and S_2.

On figure 5, the strings obtained from real images are shown. These representations are obtained using five nearest neighbors (i.e. $NB\_NEIGHBORS = 5$)

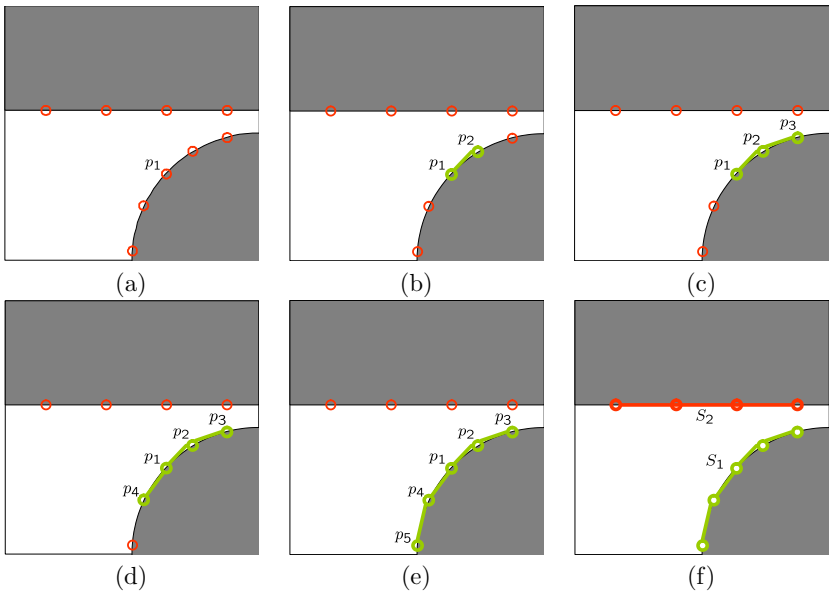


Fig. 4. salient points tracking principle

and by varying the number of angles (which is given by $\frac{2\pi}{ANGLETRESHOLD}$) to discriminate edges. Indeed, the number of angles is very important because, the fewer the angles are, the longer the strings are. For example, it can be seen on figure 5 that when only four angles are used, some strings don't follow correctly the contour of the Lenna's hat. On the contrary, when eight angles are used, the strings are reduced to one point on the cameraman trench coat. As a consequence, it seems that choosing six angles is visually better.

3.3 About Small Strings Suppression

Small strings can be viewed as noise because they don't represent structural value added information. It can thus be interesting to eliminate them. To suppress small strings, only a threshold on the string length can be used. However, we have experimentally seen that it does not permit to improve classification results when the representation is used in a natural image classification task. The work

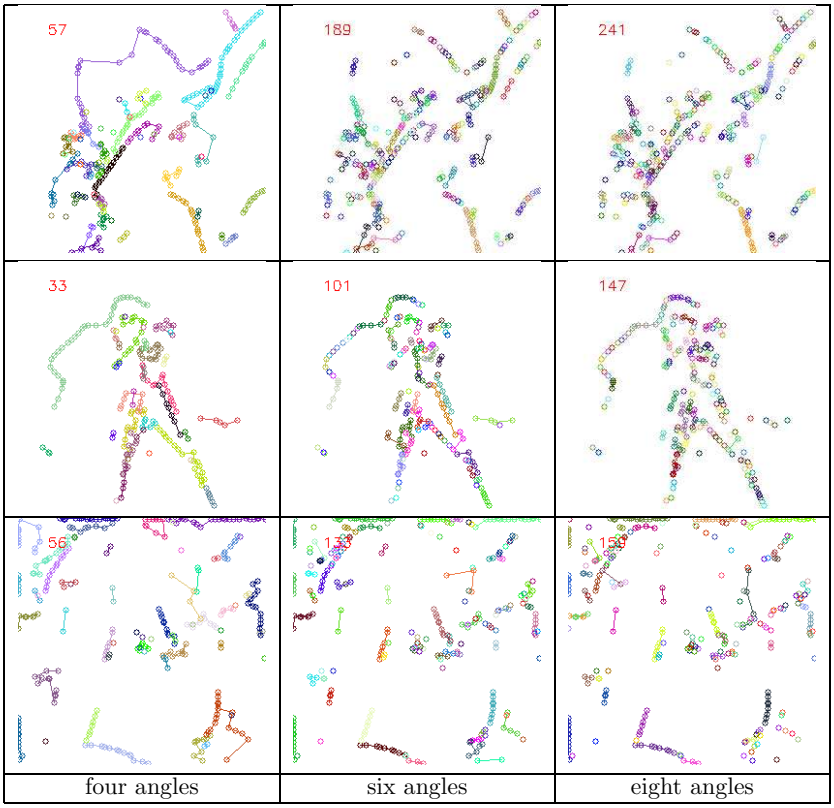


Fig. 5. strings representations on well known images

presented in [1] has lead to the same observation. Indeed, the authors try to eliminate signatures corresponding to small clusters in a kind of bag of key points approach and they have also seen that it does not improve results and that sometimes, it decreases it. As a consequence, in the following, all strings are kept in the final representation.

3.4 Discussions

At the beginning of this section, an image I was represented by a set of n local signatures denoted by $S(I) = \{s(1), \ldots, s(n)\}$. In order to add spatial information to this orderless representation, we have presented a simple method

to describe I by a set of m strings $\Lambda(I) = \{\lambda(1), \ldots, \lambda(m)\}$. We hope that this representation will improve the recognition process by providing a better description of the image content. However, there is always a lack of ordering between the constructed strings. Consequently, it cannot be used immediately by a machine learning classifier. To solve this problem, a vectorial representation of strings should be elaborated.

As far as we know, no similar case of string representations was proposed in the literature. However, there were some attempts to add spatial information into the local representations. For example in [31], the coocurrence of local signatures are used to encode the final image signature. In [46] and [17], neighborhood constraints are added. The most similar approach is probably the one presented in [39] representing an image by a set of edges. However, it does not use interest points but employs directly an edge detector to extract boundary fragments.

4 Unsupervised Classification of Strings

As previously emphasized, we aim at merging a string representation with the "bag of local signatures" representation introduced in [6]. For this purpose, we have to cluster strings in order to represent an image as a "bag of strings". This approach permits to add local information to the representation while preserving the advantages of the "bag of local signatures" approach. For this purpose, a clustering algorithm able to deal with structured data (i.e. strings) should be used.

Many solutions have been proposed to cluster structured data. The K-medoid algorithm, which is an adaption of the k-means algorithm to data which do not live in a vectorial space, is probably the simplest way to cluster non vectorial data. This algorithm constrained the codewords to take value in the training dataset such that each cluster is represented by a medoid. The medoids are

members of the learning dataset. However, it suffers from the same drawbacks than the k-means (the convergence into local minima).

In [42], a classical Self-Organizing Map (SOM) has been used to cluster the local signatures in order to construct a "bag of local signatures" representation. The SOM aims at projecting the input data space D into a lower dimensional space (1D, 2D,...) defined by a regular discrete lattice L composed of N_L nodes. Therefore, it is a vector quantization algorithm which preserves the topology of the input space because each node c of the lattice is a neuron with a codebook vector $w(c) \in \mathbb{R}^n$ such that if c_1 and c_2 are close then $w(c_1)$ and $w(c_2)$ are close in $\mathbb{R}^n$. We will see in this section that we can use some existing extensions of the SOM to cluster the strings of local signatures.

4.1 SOM Clustering of Structured data - A State of the Art

In their original form, the Self-Organizing Map can only deal with vectorial data. Thus, more complex data such as graphs or strings (or more generally temporal sequences) cannot be fed into a SOM. However, different solutions have been proposed to extend the SOM capabilities to process more complex data.

Temporal Kohonen Map Temporal Kohonen Map (TKM) have been introduced in [7] and are probably the earliest extension of SOM to the structured data. It can only process temporal sequences $X = (x(1), \ldots, x(n))$ (thus strings) and can be viewed as a classical SOM with different output neurons. Indeed, when an input $x(t)$ is fed into a classical SOM, the processing of the next input $x(t+1)$ by the SOM does not consider the previous state of the network (the value of the N_L output neurons in the lattice). To overcome this drawback, TKM proposes to consider the previous state of the map. For this purpose, when an input $x(t)$ is presented to a TKM, the output value y_i of the neuron i in the

lattice is computed as follows:

$$y_i(t) = d \times y_i(t-1) - \frac{1}{2}||x(t) - w(i,t)||^2 \tag{1}$$

$$\Rightarrow y_i(t) = d^n y_i(0) - \frac{1}{2}\sum_{k=0}^{t-1} d^k ||x(t-k) - w(i,t-k)||^2 \tag{2}$$

where $0 < d < 1$ is a time constant. At time t, the best matching unit is chosen such that it maximizes $y_i(t), i \in \{1, \ldots, N_L\}$. As a consequence, the output of previously activated neurons decrease with time and loose their activity. It is worth noting that the TKM differs only in the competitive rule because according to [26], a classical SOM update rule is used to adapt the weights associated to the nodes. This update rule uses only the last elements $x(t+1)$ presented and does not consider the entire sequence.

Recurrent Self-Organizing Map Recurrent Self-Organizing Map (RSOM) has been proposed as an improvement of the TKM in [26]. It proposes to consider the sequence processed in the update rule. For this purpose, the output y_i of the node i in the lattice is now a vector. This output is computed as follows:

$$y_i(t) = (1-\epsilon)y_i(t-1) + \epsilon(x(t) - w(i,t)) \tag{3}$$

$$\Rightarrow y_i(t) = \epsilon\sum_{k=1}^{t}(1-\epsilon)^{t-k}(x(k) - w(i,k)) \tag{4}$$

where $0 < \epsilon \leq 1$ is the "leaky coefficient". It is worth noting the closer ϵ is from zero, the longer the memory of the RSOM is. The best matching unit $y_c(t)$ at step t is chosen by searching the node having the minimum output norm:

$$c = min_i\{||y_i(t)||\} \tag{5}$$

and the update rule is then:

$$w(i, t+1) = w(i, t) + \alpha(t) h_{ci}(t) y_i(t). \tag{6}$$

where $\alpha(t)$ is the learning rate $0 < \alpha(t) < 1$ that monotonically decreases. Furthermore, h_{ci} denotes a neighborhood function that governs the strength of weight adaptation as well as the number of reference vectors to be updated. Generally, a Gaussian function is used:

$$h_{ci}(t) = exp\left(-\frac{||r_c - r_i||^2}{2\delta(t)^2}\right). \tag{7}$$

where r_c and r_i are the coordinates of the cells c and i in the lattice L and $\delta(t)$ governs the width of the neighborhood function.
However, the RSOM suffers from the same drawback that the TKM: the context is only examined during the competitive and update rule. The SOM itself does not learn context because as emphasized in [45], a sequence is coded with a vector of the same dimension than its node attributes. It could be criticism when long sequences are considered.

Richer methods have thus been proposed in order to better represent the context. They propose to encode the context of a string node directly into the weight vector of the node lattice. For this purpose, several methods have been investigated, based on the same principle: increasing the dimension of the weight vector by a value encoding the previous state of the network.

Recursive Self-Organizing Map The Recursive Self-Organizing Map (RecSOM) introduced in [48] proposes to associate two vectors to each node $1 \leq c \leq N_L$ of the discrete lattice. The first one is the classical weight vector denoted by $w(c, t) \in \mathbb{R}^n$ (the codeword associated to each cluster of the input data space

at time t). The second vector $C(c,t) \in \mathbb{R}^{N_L}$ is the context vector at time t that encodes the previous state of the RecSOM. This principle is shown on figure 6.

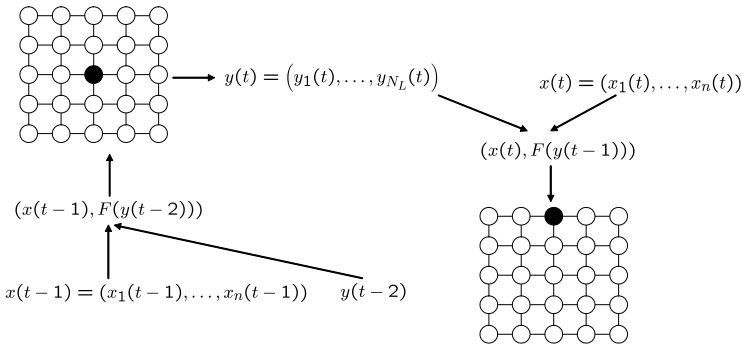


Fig. 6. RecSOM principle

At each iteration, the RecSOM algorithm computes the state $y_i(t)$ of each cell:

$$y_i(t) = \alpha||x(t) - w(i,t)||^2 + \beta||F(y_i(t-1)) - C(i,t)||^2. \tag{8}$$

The vector $y(t) = (y_1(t), \ldots y_{N_L}(t))$ encodes thus the RecSOM state at time t. It is worth noting that for the stability of the representation, $y(t)$ is not directly used as the context vector. Instead, $F(y(t))$, where F is a tranfert function, is employed. In [48], the author proposes to use the following transfert function:

$$F(y(t)) = \left(exp(-y_1(t)), \ldots, exp(-y_{N_L}(t))\right). \tag{9}$$

During the competitive rule, the best matching unit is chosen such that it minimizes $y_i(t)$. The update rule is a little bit more different than the classical Hebbian rule used in the SOM algorithm because it considers both vectors associated to a node of the lattice. For this purpose, it updates the weights of the

SOM as follows:

$$w(i,t+1) = w(i,t) + \alpha(t)h_{ci}(t)[x(t) - w(i,t)] \tag{10}$$

$$C(i,t+1) = C(i,t) + \alpha(t)h_{ci}(t)[F(y_i(t-1) - C(i,t)] \tag{11}$$

It has been shown that the RecSOM permits to obtain a good quantization of time series [48]. It is worth noting that this result has been obtained thanks to a new definition of the quantization error adapted to the time series. However, the context vector has the same dimension than the lattice resulting in a computational inefficiency [45].

Self-Organizing Map for Structured Data SOM for structured data (SOM-SD) have been introduced in [20] and can be viewed as a simplification of the RecSOM by reducing its complexity. However, it is still a powerful alternative and, compared to other approaches, it has the advantage of dealing not only with sequential data but also with more complex structures such as Direct Acyclic Graphs.

SOM-SD is in the same spirit than RecSOM in the sense that it associates two vectors to each node of the discrete lattice. The first one is still the classical weight vector denoted by $w(i,t) \in \mathbb{R}^n$ whereas the second one $C(i,t)$ is the location in the map of the best matching unit at the previous step. Obviously, this difference permits to drastically reduce the complexity of the RecSOM because the SOM is often two dimensional and so is the context vector. The principle of the SOM-SD is shown on figure 7.
At each iteration t, the best matching unit denoted by $bmu(t) = (x_{bmu}(t), y_{bmu}(t))$

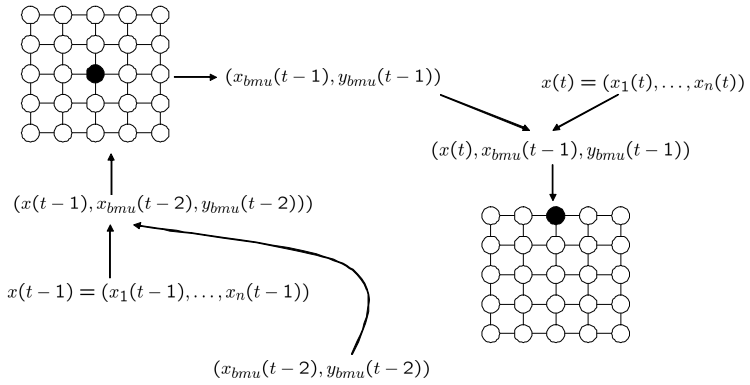


Fig. 7. principle of SOM-SD

has to be found by computing:

$$bmu(t) = argmin_{1 \leq i \leq N_L} \left(\alpha ||x(t) - w(i,t)||^2 + \beta ||bmu(t-1) - C(i,t)||^2 \right). \tag{12}$$

An important emerging problem (also inherent in the RecSOM algorithm) is the choice of both α and β because they balance the importance of the label versus the importance of the context vector. Moreover, it has been experimentally shown in [20] that the final clustering is very sensible to these values.

In SOM-SD, the update rule is the same than the Hebbian rule used in both SOM and RecSOM:

$$w(i, t+1) = w(i,t) + \alpha(t) h_{ci}(t)[x(t) - w(i,t)]; \tag{13}$$

$$C(i, t+1) = C(i,t) + \alpha(t) h_{ci}(t)[bmu(t-1) - C(i,t)]. \tag{14}$$

It is worth noting that compared to the RecSOM, several extensions have been proposed in the field of the SOM-SD. These extensions concern the supervised classification already proposed in the classical SOM algorithm with LVQ (Learning Vector Quantization) algorithm. For example in [23] and [21], the authors

propose to extend the weight vector associated to each node by a target label denoting the cluster of the string considered. More recently, in [22] an LVQ algorithm using SOM-SD has been proposed.

4.2 Discussions and Selected Approach

In [44] and [18], the different SOM architectures have been compared and one of the major conclusions is that RecSOM and the SOM-SD are the best unsupervised way to discover clusters in a structured space. However, as underlined above, the RecSOM is computationally expensive representing thus a major drawback for the applications considered in this work. Indeed, an image is composed of hundreds strings and an efficient approach, both with regard to computing times and cluster quality, should be used.

As a consequence, we have chosen to use the SOM-SD to learn string prototypes that will be later used for the "bag of strings" representation.

5 The "Bag of Strings" Representation

We have shown that it is possible to cluster structured data such as attributed strings with an adapted self organizing map algorithm. It is thus possible to construct similar representations than those presented in [6, 25] in order to represent an image by a unique vector easily classifiable and that consider the spatial information contained in the image.

5.1 Presentation

Similarly to [6], we propose to represent the image content by the probabilistic distribution denoted H over local strings. This distribution is numerically easy to compute because the space embedding strings has been quantized by using the SOM-SD algorithm. Consequently, each string activates a particular cell of

the SOM-SD representation (the best matching unit called bmu), increasing thus the activity histogram of the SOM-SD $H(\Lambda(I)) = [h_1, \ldots, h_{N_L}]$ (where $\Lambda(I)$ is the set of strings representing I) such that:

$$h_l(I) = Card\{\lambda(k) \in \Lambda(I) | l \textit{ is the SOM-SD bmu associated to } \lambda(k)\}. \quad (15)$$

The histogram $H(\Lambda(I))$ is then normalized by the number of strings in I in order to obtain a probabilistic-like distribution. The computation of the "bag of strings" representation is illustrated on figure 8. As in a "bag of local signatures"

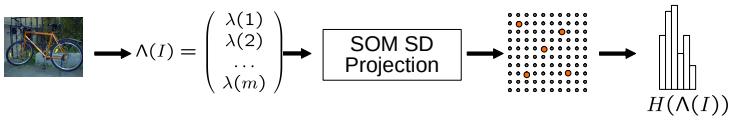


Fig. 8. computation of the "bag of strings" representation

representation, an image is finally represented by a feature vector $H(\Lambda(I))$. This feature vector embeds both the distribution of local features and the spatial relation between them. As a consequence, this representation is richer than the "bag of local signatures" approach.

6 Experiments

We have experimented the "bag of strings" approach on different well known databases splitted into a training set and a testing set as presented in Appendix A. The main goal of the experiments is to determine if the representation proposed is better than the classical "bag of local signatures" representation because previous works [31] have shown that the use of spatial information is not very important in the case of natural image classification. Consequently, results and discussions concerning these experiments are presented in this section.

Due to the size of the discrete SOM-SD lattice used in the experiments (a rectangular 50×50 map), the final signatures obtained are embedded into a high dimensional space. As a consequence, the classification results shown in this section are obtained with a basic linear SVM classifier. Indeed, SVM classifiers are more adapted for dealing with high dimensional data and this is obviously the case here.

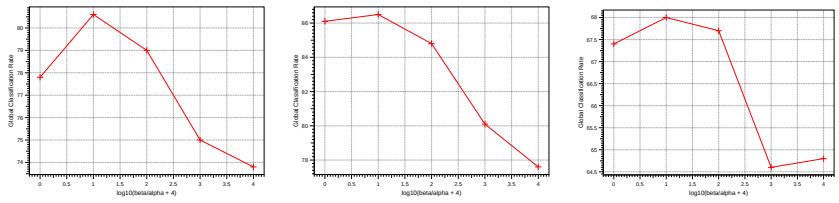
Finally, the detection of salient points has been performed by setting the energy threshold of the detector to 0.5 because experiments have shown this value leads to good classification results. Moreover, it permits to adapt automatically the number of salient points to the complexity of the image.

6.1 Study of Classification Results

When testing the approach, we have found that the global classification rates obtained depend strongly on the edge tracking algorithm and on the choice of the SOM-SD parameters (α and β). We will discuss in this section the influence of these parameters on the final results.

Influence of the SOM-SD Parameters As previously emphasized, the choice of the values of α and β is related to the importance of the label versus the importance of the context vector. At the present time, there does not exist a generic method to find them automatically and as a consequence, only an experimental study can be done in order to find the optimal values. For this purpose, we have chosen to vary β from 0.00001α to α with a logarithmic step. Moreover, the edge tracking algorithm uses five neighbors and six angles because we will see in the section 6.1 that this is the best parameters.

The global classification rates obtained with this setup and a 50×50 SOM-SD for the databases presented in appendix A are presented on figure 9.
The curves show that for the configuration considered (the regularity descriptor



SIMPLICITY Pascal Scenes

Fig. 9. evolution of the global classification rate with $\frac{\beta}{\alpha}$

with eight orientations and three Hölder bins and an edge tracking algorithm with five neighbors and six angles), the best choice for the SOM-SD parameters α and β is to take $\beta = 0.0001\alpha$ for all databases.

A deeper study permits to say that this value seems logical because it corresponds to a kind of normalization between the context vector and the regularity descriptor element. Indeed, due to its normalization and its high dimension, the signature values are very low whereas the context vector takes value in the range $[0, 50]$ which is the size of the SOM-SD. As a consequence, to take into account the signatures in the distance computed in the competitive step α should be larger than β.

In the following experiments, we will thus use $\beta = 0.0001\alpha$ because we consider it as a kind of optimal value.

Salient Point Tracking Algorithm The salient point tracking algorithm presented in section 3.2 uses two parameters. The first NB_NEIGHBORS is used to choose the number of nearest neighbors that should be considered to link salient points. We have experimentally observed that it has not a great influence on classification results. Moreover, the greater NB_NEIGHBORS is, the larger are the computing times. Consequently, we use NB_NEIGHBORS= 5 here because it permits to reach good classification rates in reasonable computing times.

The second parameters ANGLE_THRESHOLD defines the maximum value authorized between the two directions of normal regularity to link salient points. As presented in section 3.2, this threshold influences the length of the generated strings. Indeed, the strings are long when this value is small and as a consequence they do not follow edges. On the contrary, salient points are rarely linked, strings are thus very short and a kind of "bag of local signatures" representation is obtained if a large value is chosen. We present on figure 10, the global classification rates obtained for an ANGLE_THRESHOLD of 4, 6 and 8. We can see on this diagram that for all databases, choosing four angles leads

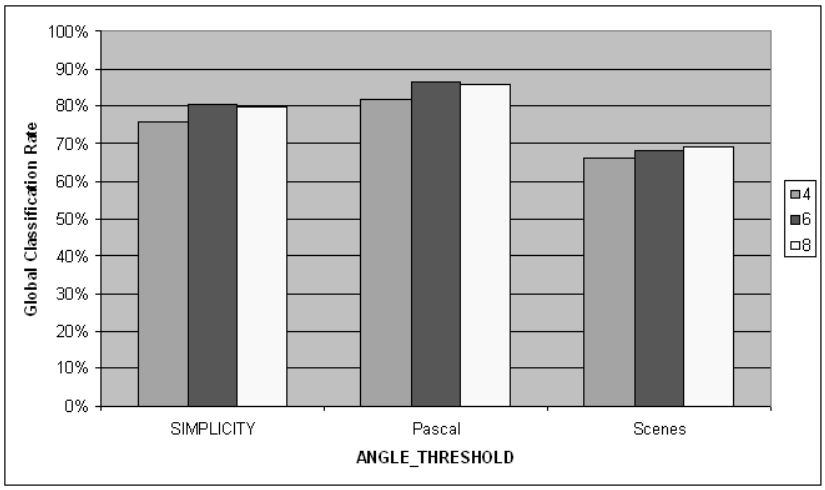


Fig. 10. influence of the edge tracking algorithm on the global classification rates

to the worst global classification rates proving that strings should follow edges in order to bring structural information to the representation. The results concerning six and eight angles are similar. However, six angles is better than eight angles for two database (SIMPLICITY and Pascal) and worse than eight angles for the scenes dataset. In the following we will thus use six angles as the optimal configuration for the salient points tracking algorithm.

Comparison with Other Approaches As previously emphasized, we would like to know if adding spatial constraints between salient points can improve classification results. For this purpose, we directly compare the best results obtained with a "bag of local signatures" approach using a classical self organizing map of size 50×50 to generate the dictionary to those obtained with a "bag of strings" approach with the optimal configuration found in the previous sections. The comparison of global classification results is presented in table 1. We can

Approach	Bag of Local Signatures	Bag of Strings
SIMPLICITY	77.6%	80.6%
PASCAL (Test1)	83.5%	86.5%
Scenes	67.9%	70.6%

Table 1. comparison of the "bag of local signatures" and the "bag of strings" approaches

see that for all databases, the global classification results obtained with the "bag of strings" approach are better than those obtained with the "bag of local signatures" approach. As a consequence, it proves that adding spatial constraints brings some information in the final signature describing the image considered. However, the improvement is not so important proving that the essential information contained in an image is encoded in the local signatures and not between them. This could be explain by the fact that spatial information is already partly taken into account when computing the local descriptor over the region of interest centered on the keypoint. The spatial information brought by our method is more a whole information that depending on the query, on the object, on the images, does not always brings significant information. Indeed, a deeper study has shown that the images newly well classified with our proposed approach always exhibits a significant spatial organization. For the Pascal dataset, classification results is really improved by more than 5% for the cars categorie but remains

the same for the people categorie. It is due to the fact that contrarly to people, cars are hand manufactured objects whose images exhibit strong long edges well described by our method.

Developing new methods for natural image classification is interesting and relevant for the computer vision domain if and only if the approaches proposed can be compared with other major methods presented in the literature.

For the Pascal dataset, the methods are often evaluated by measuring the Area Under the ROC curve (Receiver Operating Characteristic curve) denoted AUC in the following. The comparison of classification results presented during the PASCAL 2005 recognition challenge with our results are presented on table 2. Compared to the results obtained during the challenge, we are ranked 6/18

Class	Method Proposed	Worst	Best	Mean
Bicycles	0.957	0.724	0.982	0.905
Cars	0.961	0.578	0.992	0.916
Motorbikes	0.992	0.722	0.998	0.956
People	0.926	0.597	0.979	0.901

Table 2. comparison for the PASCAL dataset

on the motorbikes class, 5/16 on bicycles class, 9/16 on people class and 9/18 on the cars class. Consequently, our method perform well but is not the best on this database. However, we have not searched to tune our parameters to obtain the best results on this particular dataset because we would like to exhibit a generic architecture whose aim is to perform well on different datasets. It is motivated by the fact that in a professional application, the parameters are initially fixed to reach good classification rates for different kind of databases.

For the SIMPLICITY dataset and according to [37], the worst classification rate ever met is 37.5%. The best results found are those presented in [37] and [9] with a global classification rate of 84.1%. It is better than our classification rate of 80.6%. However in these papers, the experiments are made with a leaving-one-

out cross validation method. It consists of testing each image with a classifier trained with the remaining 999 images of the whole database. We have also tested our methods with the same approach and we obtain a global classification rate of 82%. Consequently, it could be said that our method is one of the best on this dataset.

On the difficult scenes dataset, we obtain a classification rates of 70.6%. In [32], the authors obtain 72.2% with a bag of features approach similar to ours and 81.4% by adding to this approach global constraints which are shown important for this kind of database where global invariance is not needed. For the 13 categories used firstly in the paper [14], we attain 73.5% of correctly classified instances compared to 65.2% in [14] and 74.7% in [32]. Consequently, the architecture presented in this paper gives also good results. Nevertheless, adding global constraint as in [32] seems essential to improve classification results.

6.2 Computing Times

The examination of computing times is very important in the evaluation of a natural image classification algorithm. Indeed, a commercial application should be user friendly and it is thus difficult to envisage that a user waits ten seconds to get the classification results. Due to the architecture of a supervised image classification application, most of the computing times used is due to:

1. the training step which is in fact the computation of the feature vectors for the whole training set and the learning of the classifier. It is often realized offline and can thus be long. As a consequence, it is thus not discussed here.
2. The classification of a new image depending on the computation of the signatures and on the classification algorithm.

Yet, the representation of an image by a set of strings whose node attributes are local salient signatures is more computationally difficult than representing

an image by a set of local salient signatures due to the addition of the string creation algorithm and the SOM-SD projection employed to construct the histogram representation. However the SOM-SD projection is negligible because compared to a classical self organizing map, it only adds two values representing the context vector and increases slightly the time required for the competitive and the adaptation steps.

On the opposite, the string construction imposes to compute a matrix storing distances between all interest points in the image considered and its computational complexity is $O(n^2)$ where n is the number of interest points extracted. A deeper analysis of our implementation has shown that it increases the computing times of the classical bag of local signatures representation by approximately 50%.

7 Conclusions and Discussions

In this paper, we have proposed an image representation whose goal is to overcome the major drawback of the "bag of local signatures" representation that does not consider any spatial ordering between interest points. For this purpose, we have proposed to link interest points and thus local signatures with an edge tracking algorithm in order to represent an image by a set of strings whose nodes are the local signatures. This set of strings has then been used in order to summarize the image content into a unique but large feature vector that encodes the frequency of appearance of the different strings. For this purpose, the string space has been quantized with a dedicated self organizing map.

We have shown that this monodimensional representation of the image can be easily integrated in a natural image classification environment and that a SVM classifier is suitable for classifying these high dimensional data. The representation achieves promising results on different well known databases. Indeed,

the results obtained are slightly better that those obtained with a "bag of local signatures" representation proving thus that it is important to consider the mutual information between interest points in order to better represent the image content. The computing times are important in a natural image classification system and at the present time several seconds are necessary for our system to classify one image depending on its size. We currently work on the algorithm optimisation.

To improve classification performance, it can be interesting to implement a star model [13] instead of a string model. Indeed, an image can be also represented by a set of stars representing close interest points and the star space can also be easily discretized by using a SOM-SD algorithm.

Finally, as in the "bag of local signatures" representation, the string dictionary size can be reduced by using a feature selection algorithm and other kernels can be used in the SVM classifier in order to take into account the topological ordering property of the SOM-SD.

References

1. Agarwal S. and Awan A. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.
2. Bierderman I. Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115–147, 1987.
3. Bres S. and Jolion J.-M. Detection of Interest Points for Image Indexation. In *VISUAL '99: Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 427–434. Springer-Verlag, 1999.
4. Burl M.C. and Perona P. Recognition of Planar Object Classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 223–230, June 1996.

5. Bouchard G. and Triggs B. Hierarchical Part-Based Visual Object Categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 710–715, 2005.

6. Csurka G., Bray C., Dance C., and Fan L. Visual Categorization with Bags of Keypoints. In *The 8th European Conference on Computer Vision*, pages 327–334, Prague, Czech Republic, May 2004.

7. Chappell G.J. and Taylor J.G. The Temporal Kohonen Map. *Neural Networks*, 6(3):441–445, 1993.

8. Duda R.O, Hart P.E., and Stork D.G. *Pattern Classification*. John Wiley & Sons, 2^{nd} edition edition, 2001.

9. Deselaers T., Keysers D., and Ney H. Features for Image Retrieval – A Quantitative Comparison. In *DAGM'04: 26th Pattern Recognition Symposium*, Tübingen, Germany, September 2004.

10. Everingham M., Zisserman A., Williams C., Van Gool L., Allan M., Bishop C., Chapelle O., Dalal N., Deselaers T., Dorko G., Duffner S., Eichhorn J., Farquhar J., Fritz M., Garcia C., Griffiths T., Jurie F., Keysers D., Koskela M., Laaksonen J., Larlus D., Leibe B., Meng H., Ney H., Schiele B., Schmid C., Seemann E., Shawe-Taylor J., Storkey A., Szedmak S., Triggs B., Ulusoy I., Viitaniemi V., and Zhang J. The 2005 PASCAL Visual Object Classes Challenge. In *The First PASCAL Challenges Workshop*, 2006.

11. Ferrari V., Fevrier L., Jurie F., and Schmid C. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36–51, 2008.

12. Fei-Fei L., Fergus R., and Perona P. One-Shot Learning of Object Categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.

13. Fergus R., Fei-Fei L., Perona P., and Zisserman A. Learning Object Categories from Google's Image Search. In *International Conference on Computer Vision*, pages 1816–1823, Beijing, China, October 2005.

14. Fei-Fei L. and Perona P. A Bayesian Hierarchical Model for Learning Natural Scene Categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, San Diego, USA, June 2005.

15. Fergus R., Perona P., and Zisserman A. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.

16. Fergus R., Perona P., and Zisserman A. A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 380–387, 2005.

17. Gouet V. and Boujemaa N. Object-Based Queries Using Color Points of Interest. In *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 2001)*, pages 30–36, Kauai, Hawaii, USA, 2001.

18. Hammer B., Micheli A., Strickert M., and Sperduti A. A General Framework for Unsupervised Processing of Structured Data. *Neurocomputing*, 57:3–35, March 2004.

19. Harris C. and Stephens M. A Combined Corner and Edge Detector. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, Manchester, England, 1988.

20. Hagenbuchner M., Sperduti A., and Tsoi A.C. A Self-Organizing Map for Adaptive Processing of Structured Data. *IEEE Transactions on Neural Networks*, 14(3):491–505, 2003.

21. Hagenbuchner M. and Tsoi A.C. A Supervised Self-Organizing Map for Structures. In *IEEE International Joint Conference on Neural Networks*, volume 3, pages 1923–1928, July 2004.

22. Hagenbuchner M. and Tsoi A.C. A Supervised Training Algorithm for Self-Organizing Maps for Structures. *Pattern Recognition Letters*, 26(12):1874–1884, May 2005.

23. Hagenbuchner M., Tsoi A.C., and Sperduti A. A Supervised Self-Organizing Map for Structured Data. In *WSOM 2001 - Advances in Self-Organising Maps*, pages 21–28, June 2001.

24. Jolion J.M. and Simand I. Représentation d'images par des chaînes de symboles : application la recherche par le contenu. In *20e colloque GRETSI sur le traitement du signal et des images*, volume 1, pages 18–32, Louvain-La-Neuve, Belgique, September 2005.

25. Jurie F. and Triggs B. Creating Efficient Codebooks for Visual Recognition. In *International Conference on Computer Vision*, pages 604–610, Beijing, China, 2005.

26. Koskela T., Varsta M., Heikkonen J., and Kaski K. Temporal Sequence Processing Using Recurrent SOM. In *KES'98, 2nd International Conference on Knowledge-Based Intelligent Engineering Systems*, volume 1, pages 290–297, Adelaide, Australia, April 1998.

27. Laurent C., Laurent N., Maurizot M., and Dorval T. In Depth Analysis and Evaluation of Saliency-based Color Image Indexing Methods using Wavelet Salient Features. *Multimedia Tools and Application*, 31(1):73–94, 2006.

28. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

29. Leibe B. and Schiele B. Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In *DAGM'04: 26th Pattern Recognition Symposium*, pages 145–153, Tübingen, Germany, August 2004.

30. Loupias E., Sebe N., Bres S., and Jolion J.-M. Wavelet-based salient points for image retrieval. In *Proceedings of the International Conference on Image Processing*, volume 2, pages 518–521, October 2000.

31. Lazebnik S., Schmid C., and Ponce J. A maximum entropy framework for part-based texture and object recognition. In *IEEE International Conference on Computer Vision*, volume 1, pages 832–838, Beijing, China, October 2005.

32. Lazebnik S., Schmid C., and Ponce J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, USA, June 2006.

33. Li J. and Wang J.Z. Real-time computerized annotation of pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

34. Lyu S. Mercer Kernels for Object Recognition with Local Features. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 223–229, San Diego, CA, USA, June 2005.

35. Mallat S. Foveal Approximations for Singularities. *Applied and Computational Harmonic Analysis*, 14(2):133–180, 2003.

36. Marcel M. and Cattoen M. Détection de contours et de lignes dans les procédures de bas-niveau. In *3rd Workshop on Electronic Control and Measuring System*, pages 89–97, Toulouse, France, June 1997.

37. Maree R., Geurts P., Piater J., and Wehenkel L. Random subwindows for robust image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 34–40, San Diego, USA, June 2005.

38. Mikolajczyk K. and Schmid C. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

39. Opelt A., Pinz A., and Zisserman A. A Boundary-Fragment-Model for Object Detection. In *Proceedings of the 9th European Conference on Computer Vision*, volume 2, pages 575–588, Graz, Austria, May 2006.

40. Robles-Kelly A. and Hancock E.R. Edit Distance From Graph Spectra. In *Proceedings of the 9th International Conference on Computer Vision*, volume 1, pages 234–241, Nice, France, 2003.

41. Ros J., Laurent C., Jolion J.M., and Simand I. Comparing String Representations and Distances in a Natural Images Classification Task. In *GbR'05, 5th IAPR-TC-15 workshop on graph-based representations*, pages 72–81, Poitiers, France, April 2005.

42. Ros J., Laurent C., and Lefebvre G. A Cascade of Unsupervised and Supervised Neural Networks for Natural Image Classification. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 92–101, tempe, USA, july 2006.

43. Ros J. and Laurent C. Description of Local Singularities for Image Registration. In *Proceedings of the International Conference on Pattern Recognition*, pages 61–64, Hong-Kong, China, august 2006.

44. Strickert M. and Hammer B. Neural Gas for Sequences. In *Workshop on Self-Organizing Networks (WSOM)*, pages 53–58, 2003.
45. Strickert M. and Hammer B. Unsupervised recursive sequence processing. In *European Symposium on Artificial Neural Networks'2003*, pages 27–32, 2003.
46. Schmid C. and Mohr R. Local Grayvalue Invariants for Image Retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
47. Vasconcelos N. From pixels to semantic spaces: Advances in content-based image retrieval. *Computer*, 40(7):20–26, 2007.
48. Voegtlin T. Recursive Self-Organizing Maps. *Neural Networks*, 15(8–9):979–991, 2002.
49. Wang J.Z., Li J., and Wiederhold G. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.
50. Weber M., Welling M., and Perona P. Unsupervised Learning of Models for Recognition. In *The 6th European Conference on Computer Vision*, volume 1, pages 18–32, Dublin, Ireland, 26th june – 1st july 2000.

A Databases Presentation

In this paper, different datasets have been used to perform experiments and we propose in this chapter to briefly present them. It is worth noting that all datasets involved in this paper have been divided into two parts. Indeed, our work concerns supervised natural image classification and as a consequence a training set and a testing set are needed to test the algorithms presented.

A.1 PASCAL Dataset

The PASCAL dataset has been built for the PASCAL 2005 recognition challenge[4]. The goal of this challenge was to invite various research teams to compare

[4] http://www.pascal-network.org/challenges/VOC/voc2005/index.html

their object recognition methods on a common new dataset. More information about this challenge are presented in [10]. The PASCAL dataset is divided into a training set, a validation set and two test sets. All images contain one or more instances of an object class considered. The number of images in the different parts are summarized in the table 3. The database is quite large and some ex-

	Training		Validation		Training+validation		Test1	
	images	objects	images	objects	images	objects	images	objects
Motorbikes	107	109	107	108	214	217	216	220
Bicycles	57	63	57	60	114	123	113	123
People	42	81	42	71	84	152	84	149
Cars	136	159	136	161	272	320	275	341

Table 3. Statistics of the PASCAL 2005 Dataset

amples of this dataset are presented on figure 11. It is worth noting that for the supervised classification experiments involving this database we do not use a validation set and as a consequence we have chosen to build a training set by merging the initial training and validation set.

A.2 Scene Dataset

The scene dataset[5] is a difficult dataset due to the totally subjective frontier between clusters. It contains fifteen scene categories: bedroom, suburb, industrial, kitchen, livingroom, coast, forest, highway, insidecity, mountain, opencountry, street, tall building, office and store. Each category contains 200 to 400 images and the average image size is 300 pixels. Several images of this dataset are shown on figure 12, and it is worth noting that this dataset is only composed of graylevel images. For the experiments, the same procedure is used than in [14]. It proposes to use 100 images per class for training and there are 3000 test images.

[5] http://www-cvr.ai.uiuc.edu/ponce_grp/data/scene_categories/scene_categories.zip

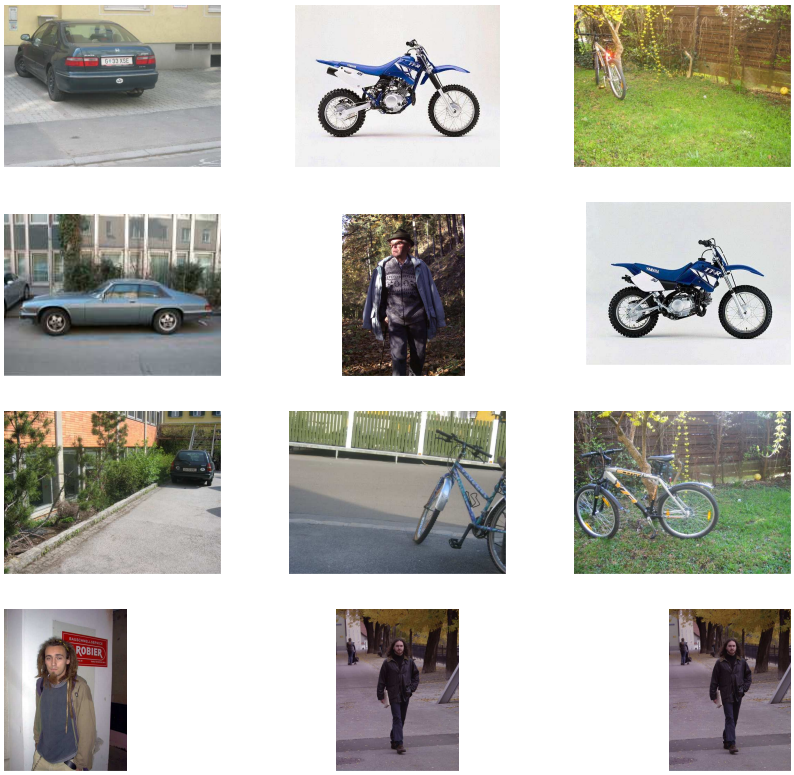


Fig. 11. PASCAL 2005 Dataset

A.3 SIMPLIcity Dataset

The SIMPLIcity database contains 1000 images of size 384×256 extracted from the well known old commercial COREL database. It can be downloaded on the James Z. Wang website[6] and has been first used to test the SIMPLIcity content based image retrieval system presented in [49]. The database contains ten clusters representing semantic meaningful categories such as Africa people and villages, beaches, buildings, buses, dinosaurs, elephants, flowers, food, horses and mountains and glaciers. There are 100 images per cluster and some images of this database are presented on figure 13. Whereas the clusters elephants, flowers,

[6] http://wang.ist.psu.edu/ jwang/test1.tar

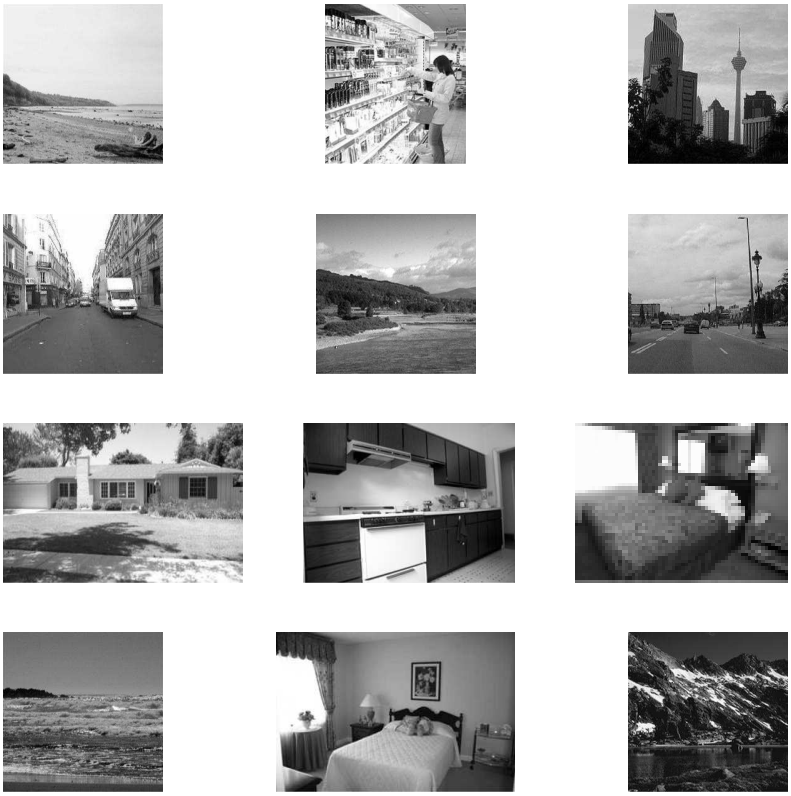


Fig. 12. Scenes Dataset

dinosaurs, foods, horses, buses, Africa people and villages can be used to test a visual object class categorization applications, the clusters beaches, buildings and mountains and glaciers are more suited to test a natural scenes images classification system. In order to test classification algorithm presented in this thesis, we have divided the database into two equal parts: 500 images are used for the training and the 500 others are used for testing.

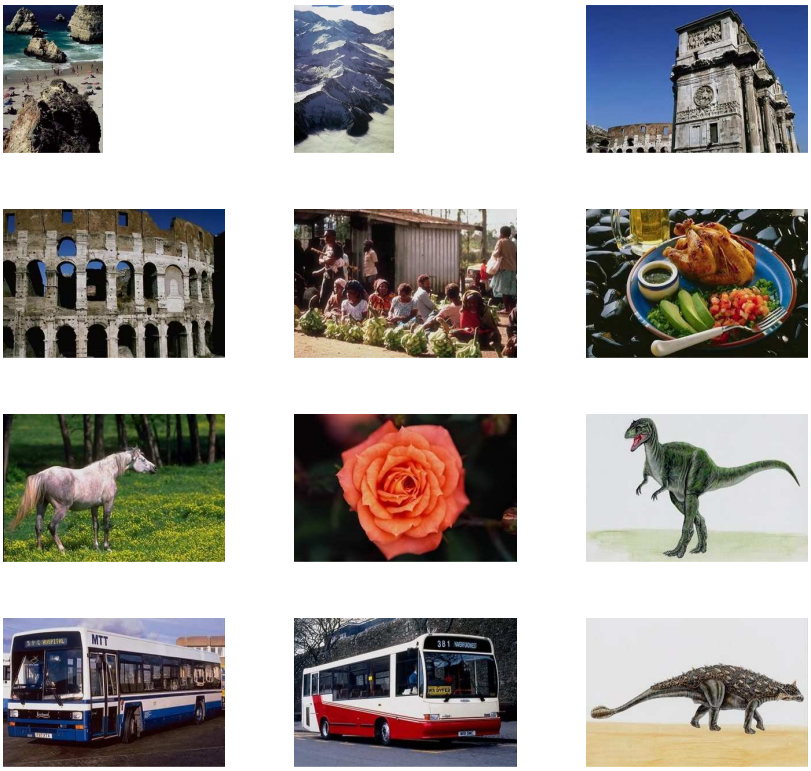


Fig. 13. SIMPLIcity Dataset

Affiliation of authors

1. Julien Ros - Université de Lyon - LIRIS - UMR CNRS 5205 - INSA Lyon - 69621 Villeurbanne Cedex, FRANCE.
2. Jean-Michel Jolion - Université de Lyon - LIRIS - UMR CNRS 5205 - INSA Lyon - 69621 Villeurbanne Cedex, FRANCE.
3. Christophe Laurent - France Télécom, RO&SI/DSIS/SIFAC, 15, avenue Léonard de Vinci, 33608 Pessac, FRANCE.

Footnotes

1. http://www.infotrends-rgi.com/home/Press/itPress/2005/1.11.05.html.
2. http://www.pascal-network.org/challenges/VOC/voc2005/index.html.
3. http://www-cvr.ai.uiuc.edu/ponce_grp/data/scene_categories/scene_categories.zip
4. http://wang.ist.psu.edu/ jwang/test1.tar.