On the use of Supervised Features for Unsupervised Image Categorization: an evaluation[☆]

Gianluigi Ciocca[a], Claudio Cusano[b,*], Simone Santini[c], Raimondo Schettini[a]

[a] *Department of Informatics, Systems and Communication (DISCo), Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy*

[b] *Department of Electrical, Computer and Biomedical Engineering, Università degli Studi di Pavia, Via Ferrata 1, 2700 Pavia, Italy*

[c] *Escuela Politécnica Superior, Universidad Autónoma de Madrid, C/ Tomas y Valiente 11, 28049 Madrid, Spain*

Abstract

Recently, new high-level features have been proposed to describe the semantic content of images. These features, that we call supervised, are obtained by exploiting the information provided by an additional set of labeled images. Supervised features were successfully used in the context of image classification and retrieval, where they showed excellent results. In this paper, we will demonstrate that they can be effectively used also for unsupervised image categorization, that is, for grouping semantically similar images. We have experimented different state-of-the-art clustering algorithms on various standard data sets commonly used for supervised image classification evaluations. We have compared the results obtained by using four supervised features (namely, classemes, prosemantic features, object bank, and a feature obtained from a Canonical Correlation Analysis) against those obtained by using low-level features. The results show that supervised features exhibit a remarkable expressiveness which allows to effectively group images into the categories defined by the data sets' authors.

Keywords: Unsupervised image categorization, supervised features, primitive features, image clustering

1. Introduction

Unsupervised categorization, often done through the use of clustering algorithms, is one of the most powerful techniques available to the designer of image management systems, as it allows categorization with no other information than

[☆] The authors contributed equally to this work.

[*] Corresponding author

Email addresses: ciocca@disco.unimib.it (Gianluigi Ciocca), claudio.cusano@disco.unimib.it (Claudio Cusano), simone.santini@uam.es (Simone Santini), schettini@disco.unimib.it (Raimondo Schettini)

that contained in the data themselves. Grouping images into semantically homogeneous classes is often a *sine qua non* for efficiently processing, structuring, querying, and browsing large collections of images. For instance, representative images can be extracted from each class to stand for the collection contents [1]; grouping similar images can also be useful for the design of effective user interfaces for browsing and visualization of image collections; image categories may be used to speed up database queries by pre-filtering the images to be searched [2], and so on. Alas, unsupervised categorization is also a very difficult problem. Without the information provided by class labels it is very difficult to obtain a reliable classification in semantically meaningful classes, and the performance of unsupervised classification is often nowhere near that of supervised methods. On the other hand, in applications one often faces the problem of categorizing a large, unstructured set of images not only without labeled training sets but, often, without *a priori* knowledge of the classes that are present in the collection.

Several authors have begun exploring features that, in addition to the image data, use semantic information in the guise of a set of labeled images belonging to a collection of pre-defined classes. These classes are not, in general, the same that we are interested in identifying in an unsupervised way, and the related labeled images come from a data set different from that which we are interested in classifying. In this paper we will consider specifically the work of Torresani *et al.* [3], Ciocca *et al.* [4], Li *et al.* [5] and Gordo *et al.* [6] . We shall refer to the features used in these papers as *supervised*, in a sense that will be clarified in the next section.

The purpose of this paper is to evaluate the performance of *supervised* features for unsupervised image categorization. First of all we verified if these features bring a significant improvement with respect to low-level features (which we shall call *primitive*). To this end we selected four data sets of different nature and four state-of-the-art clustering algorithms, and we compared the performance obtained by using supervised features with those obtained by using primitive features. We also verified how much the clustering performance depends on the dimensionality of the feature vectors. Finally, we determined whether the combination of a simple clustering algorithm and supervised features could outperform other strategies, specifically designed for unsupervised image categorization. With these experiments we try to identify strengths and weaknesses of the different supervised features in dealing with different type of images.

In the last years, a huge amount of work and resources have been devoted to the evaluation of algorithms and systems for the supervised classification of images. This effort led to the collection of standard data sets and to the definition of experimental protocols culminating with the organization of public contests and challenges. The same cannot be told for the problem of unsupervised categorization. In this context, even though the focus of this paper is the evaluation of supervised features, we believe that it could also serve as a useful source of information about the performance of low-level state-of-the-art features.

The paper is organized as follows: Section 2 provides the definition of primitive and supervised features; presents a brief review of state-of-the-art high-level descriptors; and details the features included in the evaluation. Section 3 describes the four clustering algorithms considered. The experiments, including the performance measure, the data sets, and the results are reported in Section 4 and discussed in Section 5. Finally, Section 6 concludes the paper.

1.1. Related work

In the literature there are several works dealing with the problem of unsupervised image categorization that use either low-level or high-level features.

Among the works exploiting low level features we can cite Tuytelaars *et al.* [7]. In their works, a comparison of different clustering algorithms and different bag-of-words representations of scale invariant features are presented and tested by identifying ten categories extracted from the Caltech-256 data set, and the MSCR2 data set.

SIFT-like region descriptors, within a probabilistic latent semantic analysis, framework are used to discover objects categories in unlabelled images in [8]. Five object categories from the Caltech-101 (faces, motorbikes, airplanes, cars rear, and background) are used for experimentation.

Differently, Sivic *et al.* [9] try to automatically discover a semantically meaning hierarchical structure for images based on the visual appearance of objects. A visual vocabulary of quantized SIFT descriptors are used as image representation. Learning of the objects hierarchy is achieved using a generative hierarchical latent Dirichlet allocation. The hierarchy is used to recognize nine object classes (faces, cows, grass, trees, buildings, cars, airplanes, bicycles and sky).

The problem of scene category discovering is explicitly tackled in [10]. Different representations (Gist, SIFT, PACT and color) are used to describe the images content and an information projection strategy is used to identify informative and discriminative features. The scene categorization is treated as a graph partition problem and experiments are performed on the LHI eight scene categories and MIT eight scene categories data sets. A more recent work of the same authors [11] introduces the concept of weak training sets to be used for categorization learning. Different partitioning of the data set are learned using a max-margin classifier and these partitioning are combined into an ensemble proximity matrix which is fed to a spectral clustering algorithm.

In order to cope with the possible large variability within each image category some authors incorporate into the clustering process a local analysis of relevant parts in the images common to images belonging to the same category. Lee and Grauman [12] use a novel semi-local features to describe the images in terms of neighborhood appearance and geometry. Clustering is performed by an initial grouping based on feature correspondences and then it is iteratively refined based on the evolving intra-cluster pattern of local matches. Faktor *et al.* [13], introduced a similar approach named ‘Clustering-by-Composition’. Categories are discovered by grouping images that share common statistically significant regions. These regions are those which have a low chance to occurring at random and are described in terms of HOG and Local Self-Similarity features.

Other recent studies have investigated unsupervised image categorization from a different perspective by exploring new clustering techniques and low-level descriptors. Käster *et al.* [14] tested k-means, Hierarchical Agglomerative Clustering, Partition Around Medoids and CLARA clustering algorithms on a subset of 1440 color images of 20 semantically disjoint object classes of the Columbia Object Image Library image collection. Images were described in terms of color moments, color distribution and structure. To evaluate the performance of the clustering algorithms with respect to semantically meaningful clusters the results were compared with a reference grouping by using the Rand-Index.

A spectral clustering algorithm named Locality Preserving Clustering has been presented by Zheng *et al.* [15]. The algorithm is based on a modified locality preserving projection algorithm and k-means clustering. The image descriptor is a 112 dimensional feature vector created by a combination of color histogram and color texture moments.

Grauman and Darrell [16] proposed a method where sets of local image features (SIFT descriptors compacted into ten-dimensional features via PCA) are compared in terms of partial match correspondences between component features, forming a graph between the examples that is partitioned via spectral clustering and normalized cut criterion.

Dueck and Frey [17] use affinity propagation to capture the underlying data structure. A non metric similarity function based on SIFT features is used to group similar images belonging to a subset of 20 of the 101 classes in the Caltech101 data set.

The lack of semantic information provided by the class labels could be mitigated by using suitable high-level features. In this paper we will investigate whether or not those features, learned from labeled training sets, make it possible to achieve effective unsupervised image categorization.

Image labels can be in the form of textual keywords. For example, Loeff *et al.* [18] present a method exploiting a latent space induced by pre-annotated words associated to images. This intermediate feature space is created by using a max-margin factorization model that finds a low dimensional subspace with high discriminative power for correlated image annotations. A spectral clustering approach is finally applied to the representations in the latent space.

2. Supervised features

Several approaches have been investigated to automatically incorporate semantics into image representations [19]. Recently, features that use the semantic information provided by additional labeled images have been proved to be effective in a variety of image classification and retrieval tasks [20, 3, 5, 21]. We argue that these features, which we call *supervised*, could perform well also in unsupervised image categorization.

2.1. Definition

Consider a reference database of images $D = \{x_1, \ldots, x_n\}$ and a reference partition of D into classes (according to some semantically meaningful criterion), $\mathcal{D}$, with $\mathcal{D} = \{D_1, \ldots, D_q\}$, $\bigcup_i D_i = D$, and for all $i \neq j$, $D_i \cap D_j = \emptyset$. The subsets D_i may or may not have associated labels.

A function φ for image x (that may or may not not belong to D) is *primitive* if it can be expressed as a function $\varphi = \varphi(x, D)$ while, a function ψ is *supervised* if it can be expressed as a function $\psi = \psi(x, D, \mathcal{D})$ (where the dependence on $\mathcal{D}$ is non-trivial).

According to our definition, all feature extracted solely from the image data ($\mathcal{D} = \emptyset$) are primitive, as are the features that take into account the statistical properties of the database ($\mathcal{D} = \{D\}$, e.g. those based on principal component analysis [22]). Supervised features exploit the semantic information provided by a suitable categorization of the images of the reference database.

Our definition establish a taxonomy of features that seems to be analogous to the standard (but more ill-defined) *high-level* vs. *low-level* one. We should like to emphasize that this is not the case. In fact, even though all low-level features are primitive, among high-level features there are some that we consider as primitive (such as those obtained by unsupervised learning [23]) and some which are neither primitive nor supervised (e.g. those based on textual captions and annotations).

In practice, all the supervised features that we consider here represent data as distances from a reduced number of *decision surfaces* that are used to classify them with respect to a certain number of categories. The idea is illustrated schematically in Figure 1 for a one-dimensional supervised feature. Suppose the images are described, at low-level, in a two-dimensional feature space and that they are separable in two *semantically significant* categories. We train a classifier to separate the two categories, and the classifier gives us a separation surface like that in Figure 1(a). A one-dimensional supervised representation

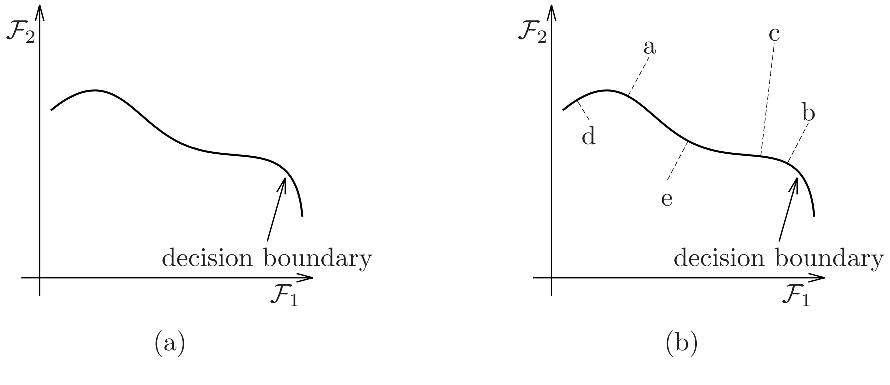

Figure 1: A one-dimensional supervised feature.

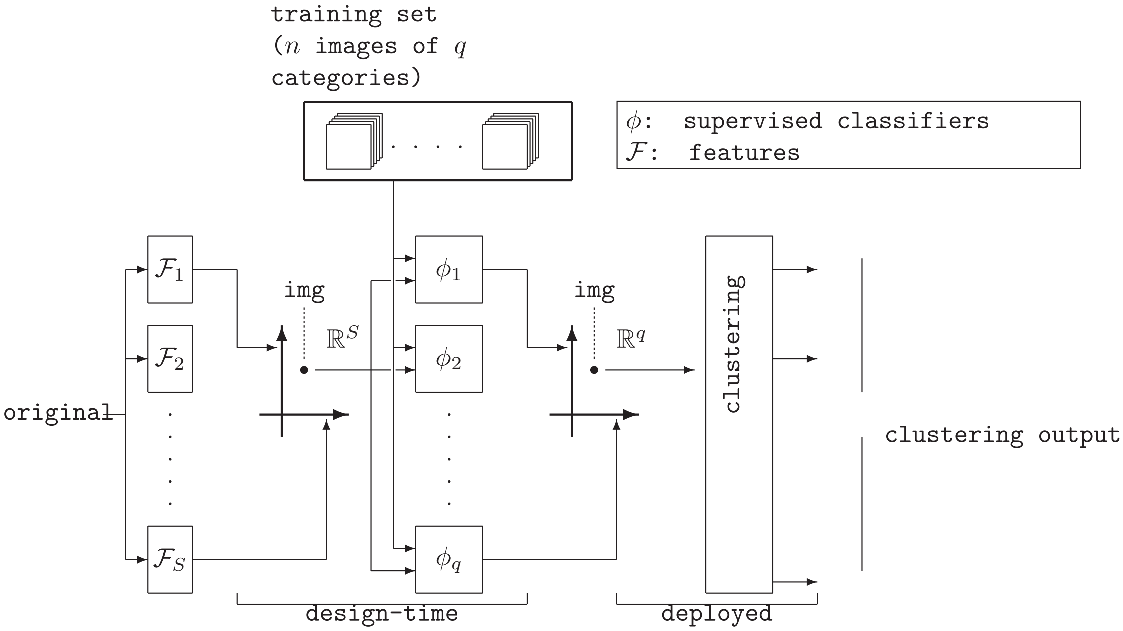

Figure 2: Unsupervised classification using supervised features.

of the set of images is then obtained by representing each image as its distance (with sign) from the decision surface as in figure in Figure 1(b). The distance having sign implies that images a and e do not have the same representation, while images a and b do. Adding a second classifier, and using the distance from both decision surfaces we obtain a two-dimensional supervised representation, and so on.

Note that this representation is not related to the statistics of the input data or to their distribution in the input space. It is a purely semantic representation and, rather than on statistical hypotheses, its validity rests on semantic ones. In statistical methods, one makes the assumption that there are dependencies among the different dimensions of the data representation. These dependencies are, of course, statistical rather than functional: knowing that on the ith axis a datum d has representation (value) v_i will not determine the value v_j on the jth axis, but will skew the probability distribution $P(v_j|d)$; that is, in general, $P(v_j|d) \neq P(v_j|d, v_i)$. (In practice, of course, we need something more: we require that $P(v_j|d, v_i)$ have a smaller variance than $P(v_j|d)$.) Supervised features make a similar hypothesis in the semantic realm regarding categories: that an image d belongs to a category c_i (with a certain probability) will not determine whether the same image belongs to a category c_j, but it will modify the probability that it be so.

In operative terms, this entails that the output of a limited number of classifiers can be embedded into a feature space that can be used as a base for unsupervised classification by a clustering algorithm. The general schema of operations is that of Figure 2. An image x is first represented as a collection of primitive features $\mathcal{F}_1(x), \ldots, \mathcal{F}_S(x)$, that is, as a point in a *primitive feature space* $\mathbb{R}^S$. During the deployment of the system, a labeled training set of n images, divided into q categories, is used to train a collection of q weak classifiers, $\phi_1, \ldots, \phi_q$, whose output spans the q-dimensional supervised feature space $\mathcal{W} \subseteq \mathbb{R}^q$ with, in general, $q \ll S$.

After system deployment, a generic image is first represented as a point in $\mathbb{R}^S$ using the low-level features extractors. The point is then used as an input to the pre-trained classifiers whose output (a point in $\mathcal{W}$) is the *supervised feature representation* of the image. This is a general-purpose feature vector that can be used for a variety of purposes, from scene categorization [5] to image retrieval [3, 21].

In this paper, we are evaluating these features on an unsupervised classification problem. So, on the deployed system (that is, with the classifiers $\phi_1, \ldots, \phi_q$ trained and considered as fixed) we consider a second data set of images (independent on that used to train $\phi_1, \ldots, \phi_q$). Each image of this second set is represented by its supervised feature vector, and these features are used as input to a clustering algorithm.

The data set is originally divided up into p, semantically meaningful, categories. This information is not available to the system (since the images are classified in an unsupervised manner), but it is used as a ground truth in order to measure the quality of the unsupervised classification.

Note that, by contrast,unsupervised classification based on primitive features uses the output of the feature extractors $\mathcal{F}_1, \ldots, \mathcal{F}_s$ directly as input to the clustering algorithm.

In the case of supervised features, both supervised and unsupervised learning play a rôle: one does first a supervised classification of a number of "base" classes in order to build the feature representation, then one uses these features for the unsupervised creation of new classes. This "mixed" schema mirrors quite well the situation that we find in application, and allows an optimal use of the data available.

It is today relatively easy to find good labeled databases that can be used to train certain classifiers in a supervised manner on the classes that they contain. On the other hand, once a system has been deployed "in the field" one encounters images belonging to new categories, categories never seen nor predicted before. Once a system has been deployed, therefore, unsupervised categorization is paramount, while during the system design one has access to labeled databases. The idea of supervised features is to take advantage of the information available at design-time to improve the performance of execution-time unsupervised categorization.

In this paper we will try to measure whether this idea can work in practice.

2.2. Approaches

A variety of high-level features has been proposed in the literature, and several of them fall into our definition of supervised features. For instance, images can be represented by sets of attributes (such as "has wheel", "is furry", etc.) [24, 25], assigned to images by suitably trained detectors. The combination of attributes that characterizes a class of images can be obtained by supervised learning, or they can be manually specified in a "zero-shot" learning framework where novel categories can be defined without training samples. Attributes have been used in a variety of scenarios such as image and video classification and retrieval [26, 27, 28, 29].

Supervised descriptions of images can be also obtained by identifying parts of them. Vogel and Schiele presented an image representation formed by local semantic descriptions [30]. They classify local image regions into semantic concept classes such as water, rocks, or foliage. Images are represented as the frequency of occurrence of these local concepts in them. In the context of image retrieval, Vogel and Schiele defined a learning procedure to obtain a perceptually plausible distance measure that led a high correlation between the human and the automatically obtained ranking. A conceptually similar descriptor as been proposed by Li *et al.* [5]. Their descriptor (called "object bank") is based on the response of a high number of object detectors run at multiple locations and scales.

Rasiwasia and Vasconcelos proposed a variation of the latent model approaches where the intermediate space is formed by "semantic themes" which are explicitly defined [31]. Each theme induces a probability density on the low-level feature space, and the images are represented by the vector of posterior theme probabilities. They shown that their low dimensional representation correlates well with human scene understanding, and outperforms the unsupervised latent space approaches.

Ciocca *et al.* presented an image descriptor, that they called "prosemantic features", based on the output of a number of image classifiers [4, 21]. These features were designed to see whether a relatively small set of concepts could work as a *base* of the concept space, so that further concepts, not explicitly designed into the system, could be derived from them (a significant property for unsupervised categorization). Prosemantic features are built by concatenating the output of 56 different soft classifiers trained to identify 14 different classes on the basis of four different low-level features.

More recently, Torresani *et al.* presented a descriptor that, while different in inspiration, is technically very similar to the prosemantic features [3]. The components of their descriptor (called "classemes") is formed by the output of classifiers trained to identify 2659 visual concepts. Wang *et al.* proposed another similar framework where images are described in terms of their affinity with respect to 103 Filckr groups [20]. They evaluated the effectiveness of such a representation in a variety of tasks, including unsupervised categorization.

Some of the descriptors mentioned above need to be tailored for the specific task in which they are employed. For instance, suitable attributes, concepts or classes must be defined and the corresponding detectors must be trained. In supervised learning it is common to make these choices with a model selection procedure that identifies the best configuration for the discrimination of a specific set of classes by using, for instance, cross-validation or independent validation sets. However, in unsupervised categorization the classes must be discovered from the data, and we cannot make any assumption about them. To prevent the introduction of any bias in the evaluation, we experimented with the supervised features as they have been defined by their original authors without any further customization. In the experiments we considered those descriptors which have been defined to be fairly general and that have been verified on a reasonable range of image categories. Moreover, the feature extraction algorithms cannot be implemented properly without the additional training data they rely on. Therefore, we decided to use only those descriptors for which the source code and data were available at the time of the experiments. The following sections describe in greater detail the descriptors that we used.

2.2.1. Classemes

Torresani *et al.* introduced an images descriptor consisting of the output of a large number of weakly trained object category classifiers [3]. Their intuition is that a novel category will be expressed in terms of the outputs of base classifiers (which they call "classemes"), describing either similar objects, or objects seen in conjunction with the target category.

A set of C category labels is drawn from an appropriate term list. For each category $c \in \{1, \ldots, C\}$, a set of training images is gathered by issuing a query on the category label to an image search engine. A one-versus-all classifier ϕ_c is trained for each category. The classifier output is real-valued, and is such that $\phi_c(x) > \phi_c(y)$ implies that x is more similar to class c than y is. Given an image x, then, the feature vector used to represent x is the classeme vector $\psi(x) = (\phi_1(x), \ldots, \phi_C(x))$.

To train the classemes Torresani *et al.* considered 2659 categories taken from the LSCOM ontology [32]. Each classifier has been trained with the LP-β multi-kernel algorithm [33]. They used 13 non-linear kernels based on a set of low-level features (Color Gist [34], Pyramid of Histograms of Oriented Gradients [35], Pyramid self-similarity [36], and bag of SIFT descriptors).

Classemes has been presented as a descriptor for image retrieval. Torresani *et al.* have shown that classification accuracy on object category recognition is comparable with the state of the art, but with a computational cost orders of magnitude lower.

2.2.2. Prosemantic features

Prosemantic features are based on the classification of images into a very small set of 14 categories: animals, city, close-up, desert, flowers, forest, indoor, mountain, night, people, rural, sea, street, and sunset. Some classes describe the image at a scene level (city, close-up, desert, forest, indoor, mountain, night, rural, sea, street, sunset), while other describe the main subject of the picture (animals, flowers, people).

For each class, several SVM classifiers are trained by using different low-level features (RGB histogram, first and second YUV moments on a 9×9 subdivision, edge direction histograms (EDH) computed on a 8×8 subdivision, and bag of SIFT descriptors). Given an image, each classifier $\phi_{c,p}$ provides a membership value which indicates how much that image is compatible with the class c from the point of view of the visual property p. Given a new image x the prosemantic feature vector ψ is obtained by concatenating the membership values: $\psi(x) = (\phi_{1,RGB}(x), \phi_{1,YUV}(x), \phi_{1,EDH}(x), \phi_{1,SIFT}(x), \ldots, \phi_{14,RGB}(x), \phi_{14,YUV}(x), \phi_{14,EDH}(x), \phi_{14,SIFT}(x))$.

Each classifier is a non-linear SVMs with Gaussian kernel, and has been independently trained on images downloaded from various image search engines with different parameters. The lack of calibration between the different components of the prosemantic features has been addressed, in the original formulation of the algorithm [21], by relevance feedback. In this work, we normalized the classifiers' output by a linear transformation

$$\phi'_{c,p}(x) = a_{c,p}\phi_{c,p}(x) + b_{c,p}, \tag{1}$$

where the parameters $a_{c,p}$ and $b_{c,p}$ are determined by a logistic regression which maps the score of the classifier to an estimate of the posterior probability

$$p(c|x) \simeq (1 + \exp(-\phi'_{c,p}(x)))^{-1}. \tag{2}$$

2.2.3. Object bank

Object Bank is an image representation constructed from the responses of many object detectors, which can be viewed as a "generalized object convolution" [5]. Two state-of-the-art detectors are used: the latent SVM object detectors [37] for most of the blobby objects such as tables, cars, humans, etc, and a texture classifier [38] for more texture-based objects such as sky, road, sand, etc. Object detectors are run across an image at different scales. Each scale and each detector yield an initial response map of the image. The authors used 177 object detectors at 12 detection scales. Each response map is then aggregated according to a spatial pyramid of three levels ($1+4+16 = 21$ blocks). The final descriptor has therefore $177 \times 12 \times 21 = 44,604$ components.

The authors evaluated the object bank descriptor in the context of scene categorization. By using linear classifiers, they obtained a significant improvement against low-level representations on a variety of data sets.

2.2.4. CCA-based features

Canonical Correlation Analysis (CCA) is a statistical tool that can be used to discover linear relationships between a pair of multivariate random variables [39]. Briefly, CCA searches for a pair of linear projections mapping the two original variables into a single space where their correlation is maximized.

Consider n items, each one described by two vectors x_i and y_i of dimensionality m_x and m_y ($i \in \{1, \ldots, n\}$) . CCA is obtained from the corresponding mean-centered matrices $X \in \mathbb{R}^{n\times m_x}$ and $Y \in \mathbb{R}^{n\times m_y}$. On the basis of the covariance matrices $\Sigma_{XX} = X^TX$, $\Sigma_{yy} = Y^TY$, $\Sigma_{XY} = X^TY$, and $\Sigma_{YX} = Y^TX$, CCA looks for the projection vectors $u \in \mathbb{R}^{m_x}$ and $v \in \mathbb{R}^{m_y}$ maximizing the correlation of the projected data:

$$\max_{u,v} \frac{u^T\Sigma_{XY}v}{\sqrt{u^T\Sigma_{XX}u}\sqrt{v^T\Sigma_{YY}v}}. \tag{3}$$

The solution to this optimization problem is given, for u by the principal eigenvector of the matrix $\Sigma_{XX}^{-1}\Sigma_{XY}\Sigma_{YY}^{-1}\Sigma_{YX}$, and for v by the principal eigenvector of $\Sigma_{YY}^{-1}\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$. The procedure can be extended to find a k dimensional common space: to do so it is sufficient to take the k leading eigenvectors of the two matrices and to use them to form the projection matrices $U \in \mathbb{R}^{k\times m_x}$ and $V \in \mathbb{R}^{k\times m_y}$.

CCA has been already used by Gordo et al. [6] to define supervised features. They considered a set of more than one million images labeled with one of 1000 categories, and extracted from them 4096-dimensional feature vectors to form the matrix X. The matrix Y has been defined by setting y_{ij} equal to 1 or 0 on the basis of whether or not the i-th image belong to the j-th category. After the solution of (3), the supervised features can be computed for new, unlabeled images by simply computing the low-level features and by applying the linear transformation defined by U.

In this work we used the same low-level features on which prosemantic features are defined (Bag of SIFT, RGB histogram, YUV moments and edge direction histogram, for a total of 2606 components). To learn the projection U, we used the labeled images from a subset of the SUN data set (described in Section 4.3) that is composed of more than 100,000 images labeled with 397 categories. In the experiments we have considered a variable number k of CCA components in the range 5–250.

2.2.5. Primitive features

We compared supervised descriptors with several state of the art primitive features. We considered three descriptors defined by the MPEG-7 standard (SCD, CLD and EHD), the Color and Edge Directivity Descriptor (CEDD), the Gist features, bag of features, and the spatial pyramid representation.

The MPEG-7 Scalable Color Descriptor (SCD) is a color histogram encoded with a Haar transform [40]. It uses the HSV color space uniformly quantized to 255 bins. To arrive at a more compact representation, the histogram bin values are non uniformly quantized. In this work we have used a 64 coefficients SCD.

The Color Layout Descriptor (CLD) represents the spatial distribution of colors in an image [40]. The RGB image is subsampled to 8×8 pixels and converted to YCbCr color space. A Discrete Cosine Transform (DCT) is applied and the CLD color descriptor is formed by reading in zigzag order six coefficients from the DCT matrix of the Y component and three components from each DCT matrix of the two chrominance components. The CLD feature is thus composed of 12 values.

The Edge Histogram Descriptor (EHD) describes edges distribution in an image [40]. Specifically it represents the distribution of five types of edges in a local region defined by subdividing the overall image into 4×4 non overlapping sub-images. Using five directional filters a five-bin directional histogram is computed from all the sub-images. The $16 \times 5 = 80$ values are then non linearly quantized and coded using three bits/bin.

The Color and Edge Directivity Descriptor (CEDD) incorporates color and texture information in a single histogram [41]. Color information is acquired by applying a set of fuzzy rules to obtain in output a 24-bin quantized histogram where each bin corresponds to a predefined color. Texture information is obtained by classifying edged into six classes by using an approach similar to the one used in the EHD feature. Combining and quantizing the the color and texture histograms, a composite descriptor of $6 \times 24 = 144$ values is obtained.

Gist features are texture features computed from a wavelet image decomposition [42]. Each image location is represented by the output of filters tuned to different orientations and scales. The resulting representation is then downsampled to 4×4 pixels. Here, we used eight orientations and four scales thus, the dimensionality of the feature vector is $8 \times 4 \times 16 = 512$ values.

Bag of visual words descriptors have become widely used for image classification and retrieval [43, 44, 45]. The basic idea is to select a collection of representative patches of the image, compute a visual descriptor for each patch, and use the resulting distribution of descriptors to characterize the whole image. In this work we considered bag of visual words based on SIFT [46] descriptors. The descriptors extracted from an image are quantized into a vocabulary of 1096 "visual words" (we used the same settings adopted for the definition of prosemantic features, as described in [21]). The final feature vector is a normalized histogram of the occurrences of the visual words in the image (1096 components).

The Fisher encoding extends the bag of words approach by considering the average first and second order differences between the image descriptors and the centers of a Gaussian Mixture Model learned from a set of local descriptors (e.g. SIFT) extracted from a set of training images [47]. Using a model with k Gaussians and d-dimensional local descriptors, the dimensionality of the Fisher vectors is $2dk$. In this work we used 64 Gaussians and SIFT descriptors reduced by PCA to 64 dimensions for a total of 8192 components[1]. Finally, take the square roots of the components and L_2-normalized the resulting vectors, as suggested by Perronnin et al. [47].

The spatial pyramid representation extends bag of visual words descriptors to incorporate spatial information [48]. Local descriptors are extracted and quantized to form multiple histograms. Histograms are organized in a pyramid: the first histogram counts the descriptors on the whole image; the following four correspond to a 2×2 subdivision of the image; and the last 16 correspond to a 4×4 subdivision (here we considered a pyramid of three levels). Each histogram is normalized according to its level to emphasize the matching at finer levels: the histograms in the first two levels are normalized so that their sum is 1/4, while the third level is normalized to 1/2. The dimensionality of the feature vector we used is $1096 \times (1 + 4 + 16) = 23,016$.

Table 1 shows the dimensionality of the features that form the object of our study. As can be seen, many of these features result in general in very high dimensional spaces composed of the juxtaposition of sub-spaces that are often

[1] We used the pre-trained version available at `http://lear.inrialpes.fr/src/inria_fisher/`

Table 1: Dimensionality of the features analyzed in our study and average time required to extract them from one image (times have been measured on a machine equipped with a 3.30 GHz Intel Core i5-2500K processor and with 16 GB of memory).

Type	Name	Dim.	Time (ms)
Supervised	Classemes	2659	1447
	Prosemantic	56	699
	Object Bank	44,604	5586
	CCA	5–250	361–362
Primitive	Fisher Vectors	8192	141
	Gist	512	348
	Bag of SIFT	1096	343
	Sp. Pyramid	23,016	371
	CEDD	144	13
	SCD	64	8
	CLD	12	5
	EHD	80	10

incoherent, in the sense their integration into a single space with a unified metric can be problematic. Even if the various components are comparable and one can define a unified Minkowski distance function, the high-dimensionality of the space limit the usefulness of this metric. It is a well known fact, for example, that in a very high dimensional space all pairs of points are virtually at the same distance: given an element x, and its closest neighbor y, then with probability $1-\epsilon N$, almost all the elements of an arbitrary set of images are contained within a sphere with center x and radius $d(x,y)+\epsilon(N)$, where $\epsilon(N)=o(1/N)$ and N is the dimensionality of the space. In content-based image retrieval this phenomenon is known as the *curse of dimensionality* [49]. In the experiments we investigated this issue in the context of unsupervised categorization (Section 4.2). We verified, in practice, how much the categorization performance is hindered by high-dimensional features, and whether or not it is possible to mitigate this problem by using Principal Component Analysis to reduce the dimensionality of the feature spaces.

Table 1 also reports the average time taken to extract each feature from a single image with the implementations used in our experiments. The times have been measured on a machine with a 3.30 GHz Intel Core i5-2500K processor and 16 GB of memory. Clearly, the computation of supervised features require more time than primitive feature extraction (particularly in the case of MPEG-7 features).

3. Unsupervised categorization

The role of good image features is to make it easier to numerically characterize the image content so that machine learning algorithms can be effectively applied. Recent work in image retrieval and classification demonstrated how supervised features are able to effectively describe the image in a way that correlates well with the user's interpretation. As we will show later, this capability allows general purpose clustering algorithms to identify groups of semantically related images. We considered four clustering algorithms, which are representative of different clustering strategies: center-based, affinity propagation, agglomerative and spectral.

The k-means algorithm is probably the most widely used clustering method [50]. The algorithm iteratively repeats two steps: in the assignment step each data point is assigned to the cluster with the closest centroid

$$C_i^{(t)} = \left\{ \mathbf{x}_j : \|\mathbf{x}_j - \mu_i^{(t)}\| \leq \|\mathbf{x}_j - \mu_h^{(t)}\|, \forall h \in \{1, \dots, k\} \right\}, \tag{4}$$

where $C_i^{(t)}$ contains the data points of the i-th cluster at the iteration t, and $\mu_i^{(t)}$ is the corresponding centroid. In the update step the centroids are recomputed as the means of the data points in the corresponding clusters:

$$\mu_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{\mathbf{x}_j \in C_i^{(t)}} \mathbf{x}_j. \tag{5}$$

The algorithm terminates when the assignments no longer change. To initialize the algorithm we used the Forgy method which randomly chooses k data points as the initial centroids. To reduce the instability introduced by the random initialization, in each experiment we run k-means 100 times with different initializations and we finally selected the clustering result that minimizes the average squared distance of the data points from their respective centroids.

Affinity propagation [51] clustering takes as input a collection of real-valued similarities between data points, where the similarity $s(i, h)$ indicates how well the data point with index i is suited to be the exemplar for data point with index h. The $s(h, h)$ values are referred as "preferences" and data points with larger values of $s(h, h)$ are more likely to be chosen as exemplars for the whole data set. By exchanging numerical messages between data points the exemplars (clusters) for the whole data set are discovered. Two messages are exchanged between data points: the "responsibility" message $r(i, h)$ and the "availability" message $a(i, h)$:

$$\begin{aligned} r(i,h) &\leftarrow s(i,h) - \max_{h' \neq h} \{a(i,h') + s(i,h')\}, & (6) \\ a(h,h) &\leftarrow \sum_{i' \neq h} \max\{0, r(i',h)\}, & (7) \\ a(i,h) &\leftarrow \min\{0, r(h,h) + \sum_{\substack{i' \neq i \\ i' \neq h}} \max\{0, r(i',h)\}\}. & (8) \end{aligned}$$

The first message indicates how strongly each data point favors the candidate exemplar over other candidate exemplars. The second message indicates to what degree each candidate exemplar is available as a cluster center for the data point. Affinity propagation simultaneously considers all data points as potential prototypes and passes soft information around until a subset of data points "win" and become the exemplars. The message-passing procedure is terminated after a fixed number of iterations or after changes in the messages fall below a threshold. For our experiments, in order to partition the data sets into the required number of clusters we used the source code downloaded from Frey's website[2]. The similarity between two feature vectors is computed by using the negation of the Euclidean distance and the algorithm is stopped after 2000 iterations.

Agglomerative hierarchical clustering creates a hierarchy of clusters by grouping similar data points. Clustering starts with a set of singleton clusters, each containing a single point. The two most similar clusters over the entire data set are merged to form a new cluster that covers both. This process is repeated until only one cluster remains. In experimenting different agglomerative criteria, we found that the Ward's linkage [52] produces better results than other linkage strategies. The Ward's linkage exploits the increase in the total within-cluster sum of squares as a result of joining cluster i and cluster j. The within-cluster sum of squares is defined as the sum of the squares of the distances between all data points in the cluster and the centroid of the cluster. The clusters distance is thus defined as:

$$d(i,j) = n_i n_j \frac{\|\mu_i - \mu_j\|^2}{n_i + n_j}, \tag{9}$$

where n_i (n_j) and μ_i (μ_j) are the size and the centroid of the cluster.

The spectral clustering approach is based on viewing the data points as nodes of a connected graph. Clusters are found by partitioning this graph, on the basis of its spectral decomposition [53]. More in detail, we used the variant originally proposed for image segmentation by Shi and Malik [54]. Given a symmetric, non-negative similarity matrix W (where W_{ij} is the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$) and given the number k of clusters, the algorithm proceeds as follows:

- the diagonal matrix D is defined as $D_{ii} = \sum_j W_{ij}$;
- the Laplacian matrix is computed as $L = D - W$;
- the eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_k$, corresponding to the k smallest eigenvalues, are computed by solving the generalized eigenproblem $L\mathbf{v} = \lambda D\mathbf{v}$;
- each data point $\mathbf{x}_i$ is transformed into a vector $\mathbf{y}_i \in \mathbb{R}^k$ by taking the i-th components of the k eigenvectors;
- the transformed points are partitioned into k clusters using the k-means algorithm.

[2] `http://www.psi.toronto.edu/index.php?q=affinity%20propagation`

To build the similarity matrix W we used a Gaussian similarity measure and a fully connected graph:

$$W_{ij} = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \tag{10}$$

For each test in the experiments, the parameter γ has been empirically tuned.

4. Experimental results

To evaluate the effectiveness of supervised features we performed three different types of comparison. In the first, we compared them to several primitive features in the state of the art by using the clustering algorithms described in the previous section. In the second, we assessed the scalability of unsupervised classification by comparing the performance of the features on a large data set. The last comparison is between the combination of supervised features with the k-means algorithm, and other methods specifically designed for unsupervised image categorization and that define their own feature extraction and clustering strategies. The experiments have been conducted on data sets commonly used for supervised image classification. The images of these data sets have been divided by their authors in different semantic categories. This allows us to objectively measure the quality of an unsupervised categorization strategy by comparing the clusters obtained with the original categorization.

The performance metric that we considered is the *classification rate* R (also used in [16, 17, 12, 13, 10]), computed as follows: each cluster found by the algorithm is associated with the ground truth category that accounts for the largest number of images in the cluster (i.e. the dominant category of the cluster); the images labeled with the same category as their own cluster are considered as correctly classified:

$$R = \frac{\sum_j \max_i N_{ij}}{\sum_{i,j} N_{ij}}, \tag{11}$$

where N_{ij} is the number of images of category i which have been placed in cluster j. Typically, the larger the number of categories found, the higher the classification rate. To balance this bias we forced the clustering algorithms to identify the number of clusters equal to the number of categories of the data sets.

Other works adopted the conditional entropy as a performance measure [7]. The main advantage of the conditional entropy is that it depends on the distribution among all the categories in each cluster while the classification rate depends only on the dominant categories. In our experiments there is a substantial agreement between the classification rate and the conditional entropy. For the sake of brevity we decided to report only the classification rate, since its interpretation is more intuitive (for instance, it allows a very rough comparison between supervised and unsupervised categorization). We report in Appendix B the conditional entropies obtained in some of the experiments.

This performance measure works under the assumption that there is a single correct way to divide the images, and that the inaccuracy of a clustering

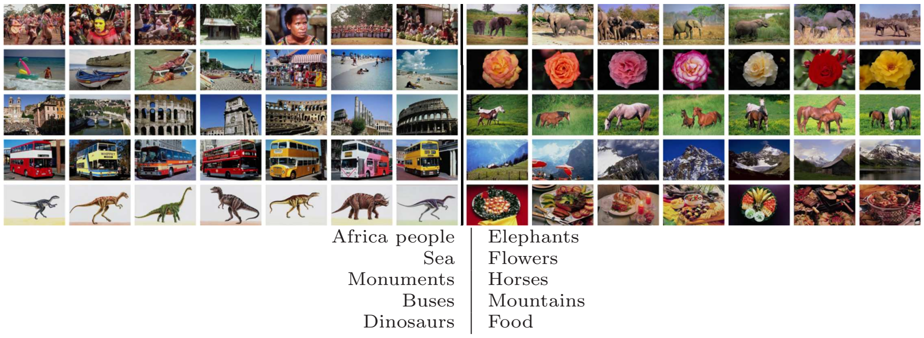

Figure 3: Samples of the ten classes of the Simplicity data set.

can be summarized by its divergence with respect to such a correct subdivision. In practice, given a collection of images, there may be many ways to divide it into semantically meaningful categories. Taking this into account, we evaluated features and algorithms on several data sets characterized by different type of contents and divided according to different criteria: one data set (Simplicity [55]) is composed of classes of visually coherent images; the second is a common benchmark for the problem of supervised scene recognition (MIT eight-scenes [42]); the third is divided by the type of event depicted (eight events classes [56]); a more fine-grained categorization is provided by a subset of the SUN data set [57], which contains more than 100,000 images divided into 397 categories of scenes; we included in the evaluation three different subsets of the Caltech data set which is mostly composed of close-ups of objects taken from an uniform point of view. The images of this data set are divided according to the object they portray [58, 59]. Finally, to verify if supervised features can be used outside their natural domains, we evaluated them on two additional data sets: one composed by textures, the other by aerial images; the results of this last experiment are reported in Appendix A.

4.1. Features comparison

The first experiment has been conducted on the Simplicity data set [55]. It is a subset of the COREL data set, formed by ten image categories each containing 100 images (see Fig. 3). It can be considered an “easy” data set, since the ten categories are clearly distinct, with little or no ambiguity. On the one hand, this restricts the significance of the experimentation, but on the other hand, it makes the results more reliable (since there is only a single, reasonable way of dividing the data in ten meaningful clusters). The results obtained are summarized in Table 2. Depending on the clustering algorithms considered, the best performance is obtained either by classemes or by prosemantic features. In almost all cases, the supervised features outperform the primitive features, although the CEDD features perform in general better than Object Bank and are relatively close to the performance of the loser between classemes and prosemantic. Considering, for instance, the k-means algorithm, the ten clusters

Table 2: Classification rates on the Simplicity data set (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	*k*-means	Spectral	Ward	Affinity Prop.
Classemes	65.0	69.1	**65.1**	**66.8**
Prosemantic	**73.7**	**77.9**	64.9	64.0
Object bank	57.8	56.4	57.0	51.8
CCA-56	62.3	68.8	46.3	52.4
Fisher Vectors	55.1	57.1	58.2	42.9
Gist	33.7	35.0	33.5	29.8
Bag of SIFT	49.0	45.5	48.3	44.6
Spatial Pyramid	47.4	52.3	41.8	45.0
CEDD	62.2	61.0	64.2	60.1
SCD	42.3	42.4	40.5	42.2
CLD	54.1	59.1	53.4	50.8
EHD	50.4	51.5	46.8	47.2

Table 3: Confusion matrix for the Simplicity classes in the clusters found by *k*-means applied to the prosemantic features. For each cluster, the dominant category is reported in bold.

	Cluster									
Class	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Africa people	**67**	1	-	14	-	14	2	-	-	2
Buses	1	**95**	-	-	-	-	-	4	-	-
Dinosaurs	-	-	**100**	-	-	-	-	-	-	-
Elephants	-	-	1	**66**	-	1	7	-	2	23
Flowers	1	-	-	-	**96**	2	1	-	-	-
Food	28	-	4	-	3	**64**	-	-	1	-
Horses	3	-	-	1	-	-	**96**	-	-	-
Monuments	17	13	-	7	1	-	-	**56**	2	4
Mountains	-	-	-	1	-	-	1	4	**81**	13
Sea	6	1	1	2	-	2	-	15	11	**62**

found on the prosemantic features represent a good approximation of the ten categories of the Simplicity data set (see Table 3). In particular, simple categories (buses, dinosaurs, flowers, and horses) have been identified with very few errors. Several images of the Africa people and food categories present clear similarities in terms of composition and color distribution and are misclassified. The clusters found by using classemes (Table 4) are, in comparison, more confused: horses are not separated from elephants, and images of mountains and sea are distributed among multiple clusters. The clustering obtained with object bank features roughly corresponds to the ten ground truth classes (see Table 5).

Table 4: Confusion matrix for the Simplicity classes in the clusters found by k-means applied to classemes. For each cluster, the dominant category is reported in bold.

	Cluster									
Class	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Africa people	**61**	-	-	14	8	10	-	5	1	1
Buses	-	**92**	-	-	-	-	-	6	2	-
Dinosaurs	-	-	**98**	-	2	-	-	-	-	-
Elephants	14	-	-	-	1	42	7	3	33	-
Flowers	5	-	-	**91**	4	-	-	-	-	-
Food	15	-	-	6	**65**	4	-	-	10	-
Horses	4	-	-	-	-	**44**	**43**	4	5	-
Monuments	4	2	1	-	3	5	-	**64**	20	1
Mountains	2	-	1	1	3	31	-	1	**47**	14
Sea	1	-	-	1	5	4	-	3	41	**45**

Table 5: Confusion matrix for the Simplicity classes in the clusters found by k-means applied to object bank features. For each cluster, the dominant category is reported in bold.

	Cluster									
Class	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Africa people	**32**	1	-	9	-	41	8	2	6	1
Buses	-	**97**	-	1	-	-	-	2	-	-
Dinosaurs	-	-	**99**	-	-	-	-	-	-	1
Elephants	1	-	-	**42**	-	1	18	5	14	19
Flowers	6	-	-	-	**58**	36	-	-	-	-
Food	10	-	1	4	2	**56**	-	-	25	2
Horses	2	-	-	5	-	-	**66**	8	18	1
Monuments	12	8	2	13	-	4	2	**39**	13	7
Mountains	4	-	-	37	-	1	3	2	**31**	22
Sea	1	-	-	20	-	2	2	3	14	**58**

However, there are clusters (e.g. #1, #4, # 9) containing a mix of images from several classes. The confusion matrix corresponding to CCA features (Table 6) highlights the limitations of this approach: the linear projection obtained from the SUN data set tend to produce outliers on other data sets. These outliers are often grouped by the clustering algorithms in very small clusters. Note that for CCA-based features we need to chose the number of components. In all the experiments reported here, we selected the number obtaining the highest classification rate (56 in this case). See Section 4.2 for an analysis of the performance obtained with a varying number of components.

For the sake of completeness, we also report here the 84.8% accuracy obtained by the latent space approach proposed by Loeff *et al.* [18]. It should be noted that the data set used is similar in composition and size to the Simplicity

Table 6: Confusion matrix for the Simplicity classes in the clusters found by k-means applied to CCA features (56 components). For each cluster, the dominant category is reported in bold.

	Cluster									
Class	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Africa people	-	-	20	-	-	-	-	78	2	-
Buses	**88**	-	-	-	6	-	-	-	-	6
Dinosaurs	-	**96**	4	-	-	-	-	-	-	-
Elephants	-	-	**94**	-	-	-	-	1	4	1
Flowers	-	1	2	**47**	**31**	**9**	**1**	4	5	-
Food	2	-	7	3	-	-	-	**85**	-	3
Horses	-	-	4	-	-	-	-	1	**95**	-
Monuments	-	2	30	1	-	-	-	14	-	53
Mountains	-	1	13	1	-	-	-	8	-	**77**
Sea	-	-	20	-	-	-	-	3	-	**77**

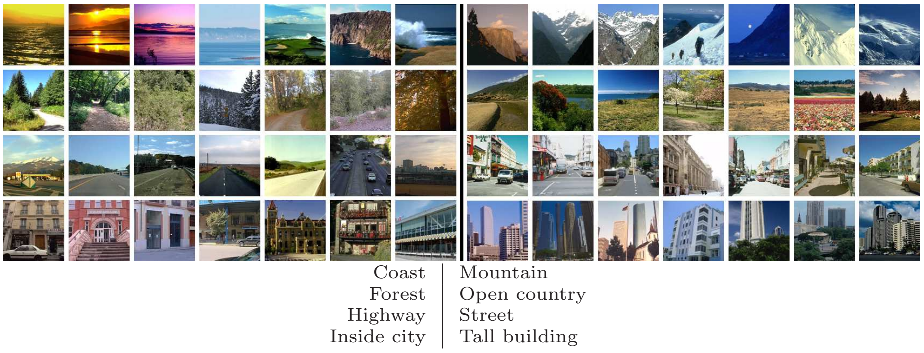

Figure 4: Samples of the eight classes of the scene recognition data set.

one but the image categories are not the same. Specifically, the categories used are sunsets, race cars, flying air planes, African animals, swimming, Egyptian ruins, birds and nests, trains, mountains and snow, and beaches for a total of 1000 images equally distributed in the different groups.

For the second experiment, we considered the scene recognition data set collected by Oliva and Torralba [42] to evaluate features and methods for scene classification (Fig. 4). This data set contains eight outdoor scene categories: coast, mountain, forest, open country, street, inside city, tall buildings and highways, for a total of 2688 images (260–410 images per class). With respect to the Simplicity data set, there is less inter-class variability, and therefore the classes are expected to be harder to separate. Table 7 summarizes the results obtained by the four clustering algorithms. The best results have been obtained by applying spectral clustering to classemes. Classemes identify all the eight ground truth categories (even in the case of k-means, as shown in

Table 7: Classification rates on the scene recognition data set (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	76.4	**80.2**	62.2	64.8
Prosemantic	**78.3**	76.2	**73.9**	**73.6**
Object bank	70.0	63.3	67.8	63.3
CCA-56	69.5	66.4	58.1	65.1
Fisher Vectors	41.7	36.9	44.4	37.3
Gist	57.1	61.4	52.7	49.9
Bag of SIFT	39.1	38.2	39.3	37.4
Spatial Pyramid	43.0	46.9	44.1	44.7
CEDD	38.3	40.3	36.1	35.8
SCD	27.1	29.2	26.8	26.8
CLD	32.1	31.6	32.2	33.4
EHD	59.5	61.1	57.0	53.9

Table 8). Prosemantic features also obtained good results, with all the four clustering algorithms. Interestingly, with prosemantic features two classes (i.e. "street" and "inside city") have been invariably merged into a single cluster (see Table 9). On the other hand, the "coast" class has been split into two clusters. By examining the two clusters we found out that the smallest is mainly composed of sunset images. Worse results have been obtained by using the object bank features for which "coast", "highway", and "open country" images are difficult to separate. With CCA-based features all the eight classes have been identified. However, their overall performance is similar to those of object bank features.

Among primitive features, the best results have been obtained by the MPEG-7 EHD descriptor, and by the Gist features which have been designed in particular for supervised classification on this data set (83.7% accuracy when processed by a support vector machine [42]).

Dai *et al.* considered this data set for the evaluation of their unsupervised classification method [10]. They obtained a classification rate of 63.5%, a value which is better than the performance we obtained with primitive features.

The third data set considered contains images of eight different classes of events [56]. This data set has been collected in order to evaluate event classification methods. It is composed of 1579 images (137–250 images per class) showing people performing various sport activities (rock climbing, rowing, badminton, bocce, croquet, polo, sailing, and snowboarding). Of the three data sets, this is undoubtedly the most challenging, as events can't be classified only at a scene level, but object detection and pose recognition are often required. For instance, the difference between "bocce" and "croquet" images often rests merely on the presence of a mallet, as it can be seen in Fig. 5. Table 10 reports

Table 8: Confusion matrix for the classes of the scene recognition data set in the clusters found by k-means applied to the classemes. For each cluster, the dominant category is reported in bold.

	Cluster							
Class	#1	#2	#3	#4	#5	#6	#7	#8
Coast	**201**	2	92	-	1	62	-	2
Forest	-	**312**	-	-	14	2	-	-
Highway	21	-	**202**	9	5	15	8	-
Inside city	1	-	-	**249**	-	-	51	7
Mountain	4	17	6	-	**274**	71	2	-
Open country	11	19	57	-	46	**276**	-	1
Street	-	-	1	30	-	-	**259**	2
Tall building	-	4	-	26	1	-	45	**280**

Table 9: Confusion matrix for the classes of the scene recognition data set in the clusters found by k-means applied to the prosemantic features. For each cluster, the dominant category is reported in bold.

	Cluster							
Class	#1	#2	#3	#4	#5	#6	#7	#8
Coast	**226**	**111**	2	9	-	6	6	-
Forest	-	-	**289**	-	-	28	7	4
Highway	21	5	-	**198**	10	7	8	11
Inside city	-	-	1	11	**248**	-	-	48
Mountain	29	5	2	2	-	**323**	12	1
Open country	85	7	29	3	-	65	**221**	-
Street	-	-	-	39	221	1	-	31
Tall building	3	5	-	11	42	10	-	**285**

Figure 5: Samples of the eight classes of the event recognition data set.

the results obtained on this data set. As expected the performances are lower than those obtained in the previous experiments. Again, the use of supervised features led to the best results (CCA, classemes and prosemantic features, in

Table 10: Classification rates on the event data set (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	62.0	61.5	**64.2**	**58.8**
Prosemantic	64.9	64.2	63.0	50.1
Object bank	43.1	43.4	45.1	40.5
CCA-56	**65.2**	**65.3**	51.2	48.3
Fisher Vectors	39.5	40.5	37.1	35.2
Gist	46.2	45.3	47.7	42.4
Bag of SIFT	36.6	37.4	32.4	33.9
Spatial Pyramid	36.7	34.1	34.0	29.6
CEDD	40.4	40.0	40.5	38.2
SCD	27.9	27.3	28.0	28.0
CLD	32.1	33.4	30.5	35.9
EHD	49.6	48.3	43.5	43.3

particular obtained about 65% of accuracy).

The difficulty of the data set is also witnessed by the relatively low classification rates obtained in the literature. For instance, the best two methods reported by Li and Fei-Fei [56] on this data set obtained about 73% and 60% of classification accuracy.

As expected, the categories croquet and bocce are often misclassified. In general there is a high degree of confusion between the bocce, croquet, and polo categories.

Concerning the clustering algorithms, they usually obtained similar results, with the exception of k-means and normalized cut when applied to classemes or prosemantic features. For the other features it is not possible to derive a clear conclusion: the best algorithm varies on the feature/data set combination. Globally, the worst algorithm is Affinity Propagation. This may depends on the fact that the original algorithm is not supposed to output a user-specified number of clusters. To make an uniform comparison, we have used the heuristic procedure, suggested by the authors of the algorithm, which forces the creation of the desired number of clusters. Table 11 reports the mean classification rates on the three data sets.

4.2. Dimensionality reduction

A possible explanation of the good results obtained with supervised features is that they perform a form of dimensionality reduction by defining a transformation from the original high-dimensional feature space to a more manageable, low-dimensional semantic space. In particular, prosemantic features reduce the original 2606-dimensional space to just 56 components. To verify how much the dimensionality of the feature space influences the clustering accuracy, we

Table 11: Mean classification rates on the three data sets (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	67.8	70.3	63.8	**63.5**
Prosemantic	**72.3**	**72.8**	**67.3**	62.6
Object bank	57.0	54.4	56.6	51.9
CCA-56	65.7	66.8	51.9	55.3
Fisher Vectors	45.4	44.8	46.6	37.3
Gist	45.7	47.2	44.5	40.7
Bag of SIFT	41.6	40.4	40.0	38.6
Spatial Pyramid	42.4	44.4	40.0	39.8
CEDD	47.0	47.1	46.9	44.7
SCD	32.4	33.0	31.8	32.2
CLD	39.4	41.4	38.7	40.0
EHD	53.2	53.6	49.1	48.1

repeated the experiments on the three data sets by applying a Principal Component Analysis (PCA) as a preprocessing step before k-means clustering.

More in detail, in this experiment we considered three supervised features (classemes, prosemantic and object bank) and the largest primitive features as reported in Table 1 (Fisher Vectors, Gist, Bag of SIFT, and Spatial Pyramid). We gradually reduced the number of components retained by PCA from the original dimensionality of the feature vector, down to five. For CCA-based features, instead, we simply considered different number of components without the final PCA. The classification rates obtained are reported in Figures 6, 7, and 8.

The plots show that, in general, the application of PCA does not affect the classification rates. In all the cases, the performance obtained with the original features is very close to the performance obtained on their ten-dimensional versions. Further reducing the components to five causes in some cases a drop in the classification rate. The only exception to this behavior is represented by classemes on the Simplicity data set: in that case the plot shows some degree of instability of the performance with respect to the dimensionality of the feature space.

In this experiment we also considered the combination of low-level features used to build the prosemantic features (in the figures this descriptor is reported under the name of 'preclassification' features, see Section 2.2.2 for more details). The figures clearly show how prosemantic features represent a better transformation of the low-level feature space with respect to the unsupervised, linear transformation defined by the PCA.

For CCA features we observe that the best number of components is 56 for all the three data set.

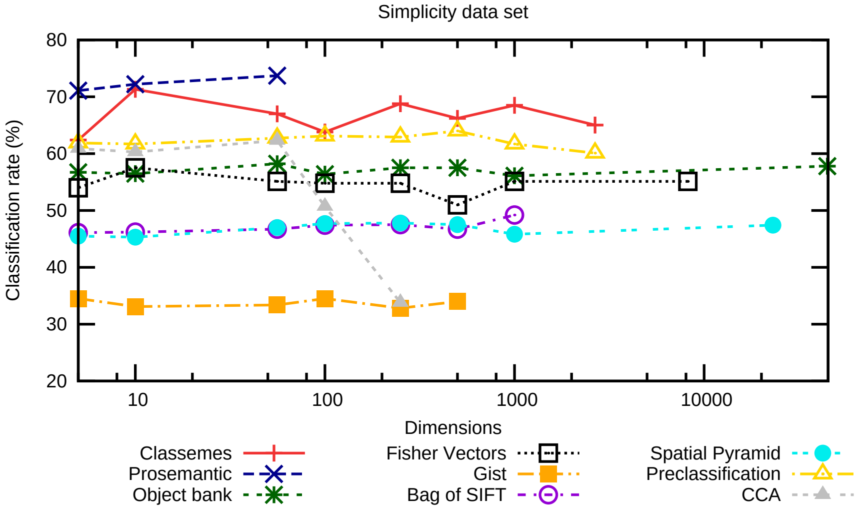

Figure 6: Classification rates obtained on the Simplicity data set by the *k*-means algorithm after dimensionality reduction, as a function of the number of principal components retained.

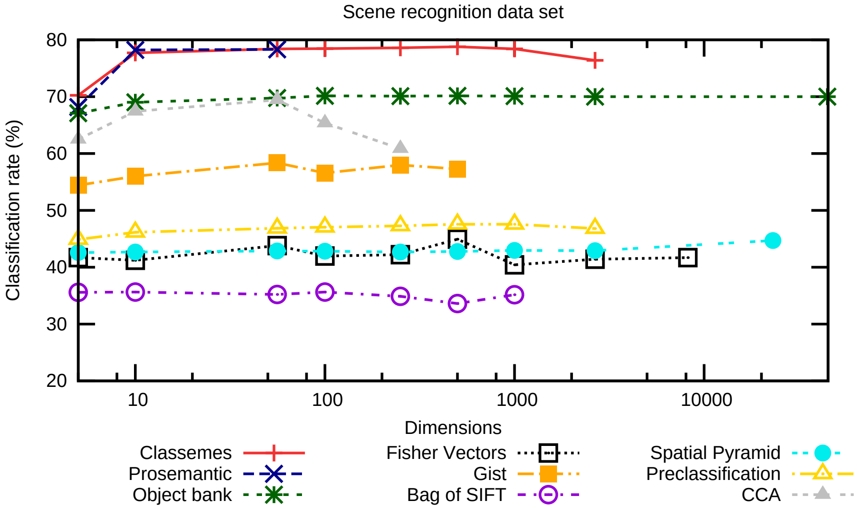

Figure 7: Classification rates obtained on the scene recognition data set by the *k*-means algorithm after dimensionality reduction, as a function of the number of principal components retained.

4.3. Large scale evaluation

The previous experiments shown that supervised features clearly outperform the primitive ones on data sets with 8–10 categories. To verify if this is still true for large number of categories, we performed an experiment on the SUN (Scene

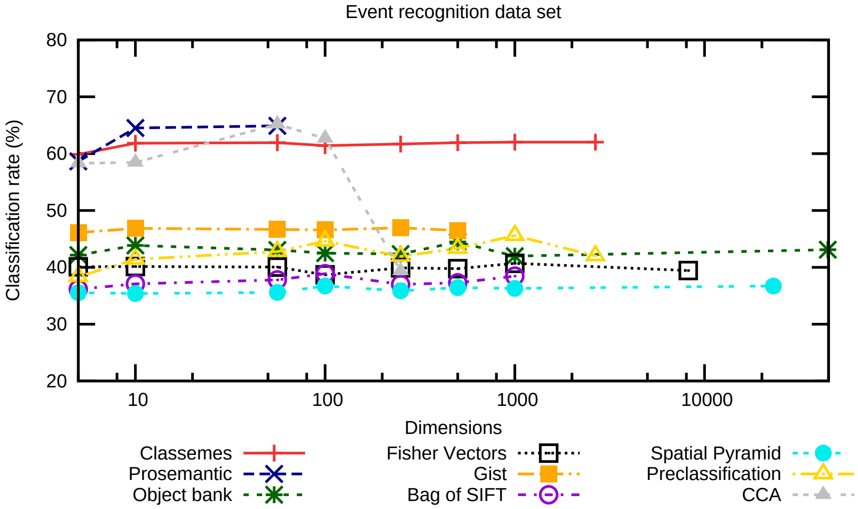

Figure 8: Classification rates obtained on the event recognition data set by the k-means algorithm after dimensionality reduction, as a function of the number of principal components retained.

Understanding) data set [57]. The data set was collected by selecting from the available terms of WordNet [60] those describing concrete scenes, places, and environments. After the removal of synonyms the final set of terms numbered 899 categories. For each term, images were retrieved from the Web by using different search engines obtaining a total of 130,519 images. As suggested by Xiao *et al.*, we considered only those categories containing at least 100 images. The final image data set is thus composed of 108,754 images belonging to 397 categories.

For this experiment we considered three supervised features (classemes, prosemantic and object bank) and five primitive features (Fisher Vectors, Gist, Bag of SIFT, Spatial Pyramid, and the 'preclassification' features used to build the prosemantic ones). Only the k-means algorithm has been considered. CCA-based features have been excluded from this experiment because they have been derived from the labeled images of this data set. Therefore, the use of these features could not be considered as unsupervised.

In addition to the performance obtained by each feature on all the 108,754 images, we also verified how much the classification rate varies as a function of the number of categories. To do so, we randomly selected a set of categories and repeated the experiment by considering only the images belonging to those categories. The numbers of categories considered were 12, 25, 50, 100 and 200. For each of these numbers, we repeated the experiment ten times with a different random selection (the same selections of categories have been used for all the features). The averages over the ten runs of the classification rates are reported in Figure 9.

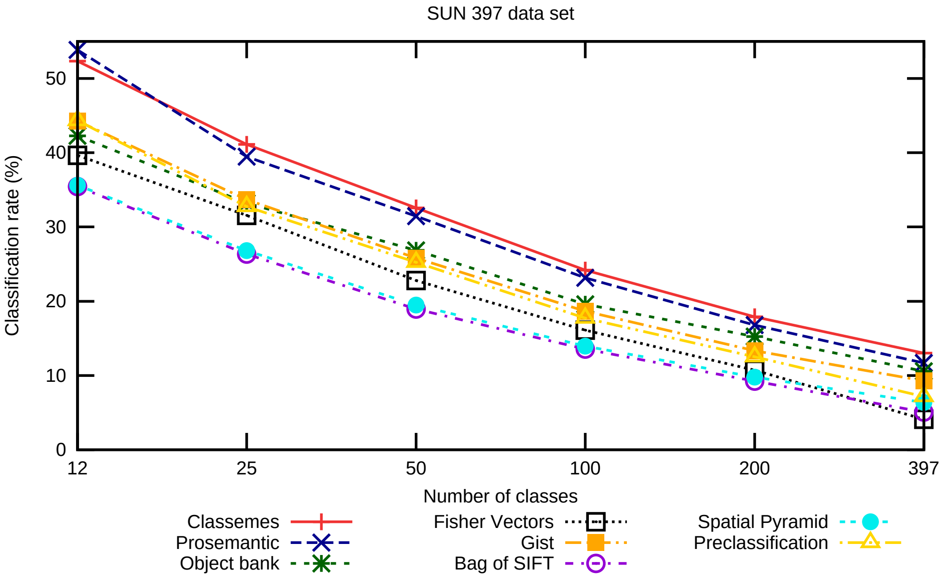

Figure 9: Average classification rates obtained on subsets of the SUN-397 data set by the k-means algorithm as a function of the number of categories. With the exception of the data points at 397 categories, the classification rates are the average over ten runs.

The plot shows how supervised features consistently obtain higher classification rates than the primitive ones. When clustering is performed over 12 categories, prosemantic features obtain the best results. When more categories are considered, the best feature is classemes. As expected, for all the features the classification rate decreases with the number of classes. However, this decrease is less evident for the object bank features that, for the largest number of categories, approaches the performance of the best features. Gist were the best among primitive features.

Since we repeated the clustering multiple times, we can also compute the standard deviations of the classification rates. These are reported in Figure 10. They decreases from about 7% for 12 categories, to less than 1% for 200 categories.

When all the data set is used, the performance of the supervised features is quite similar (about 13.0%, 11.7% and 10.6% for classemes, prosemantic and object bank, respectively). However, the three features induce very different clusterings. This fact can be seen in Figure 11 which reports the distribution of the size of the clusters. With classemes both small and large clusters are found: the largest cluster contains 684 images and there are nine clusters containing a single image. The clusters found with prosemantic features tend to have a more uniform size: the smallest contain eight images and the largest contain 526 images. The use of object bank produces several small clusters (26 with a single image, 47 containing less than seven images). Not considering

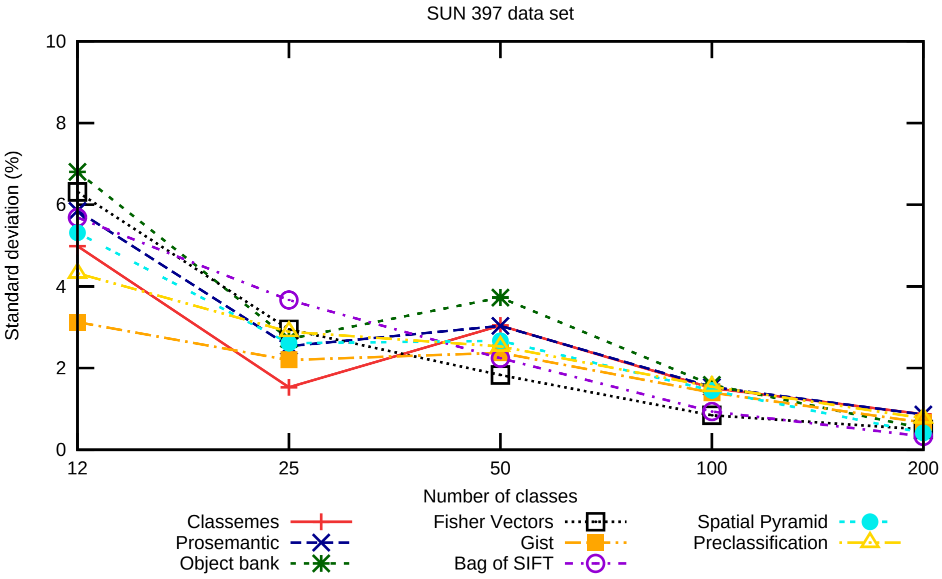

Figure 10: Standard deviation of the classification rates obtained on subsets of the SUN-397 data set by the k-means algorithm as a function of the number of categories.

these small clusters, the distribution of the remaining ones resemble that obtained with prosemantic features. The largest cluster contains 551 images. The standard deviations of the distributions are 155.1, 83.2 and 132.0 for classemes, prosemantic and object bank features. The scatter plots in Figure 12 report the size and the purity of the clusters found with the three features. Purity is defined as the fraction of images belonging to the dominant class in the cluster. The purest clusters tend to small and large clusters are, in general, quite impure.

4.4. Comparison with other strategies

As a final experiment we compared the performance of supervised features against the results reported by Grauman and Darrell [16], Dueck and Frey [17], Lee and Grauman [12], and Faktor and Irani [13]. These methods use quite different clustering paradigms.

Grauman and Darrell used local SIFT descriptors compacted into ten dimensional features via PCA to cluster images with the normalized cut algorithm. They experimented on random subsets of 400 images of the Caltech-4 data set [58] (1,155 images of cars, 800 images of airplanes, 435 images of frontal faces, and 798 images of motorcycles, see Figure 13). Dueck and Frey used affinity propagation to cluster images on the basis of a non metric similarity between SIFT features. They experimented on two subsets of the Caltech-101 data set [59], formed by taking a maximum of 100 images from a selection of

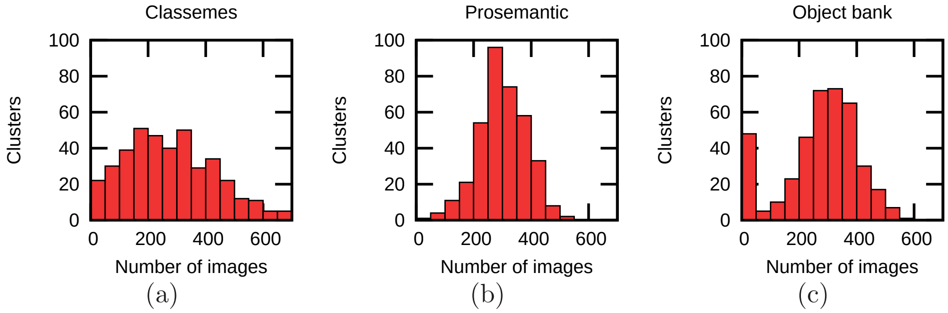

Figure 11: Distribution of the cluster size corresponding to the clusterings obtained on all the images from the SUN-397 data set by the k-means algorithm on (a) classemes, (b) prosemantic, and (c) object bank features.

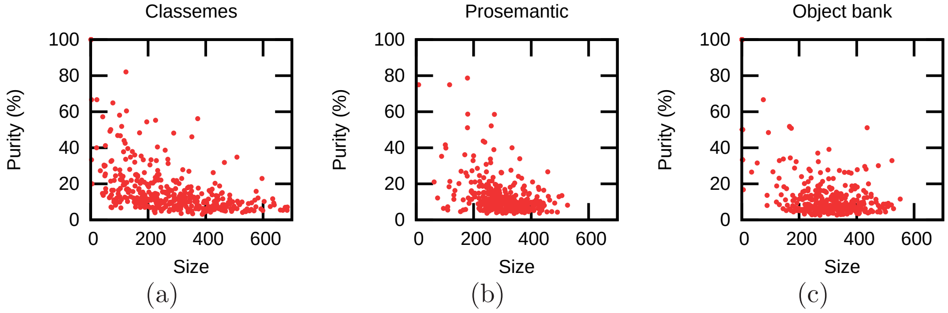

Figure 12: Size vs. purity of the clusters found by k-means on all the images from the SUN-397 data set using (a) classemes, (b) prosemantic, and (c) object bank features.

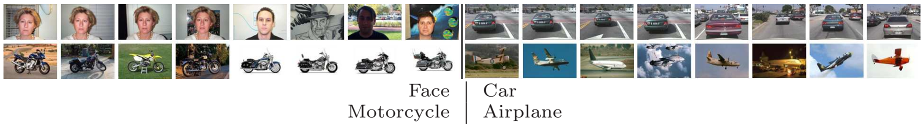

Figure 13: Samples of the four classes of the Caltech-4 data set.

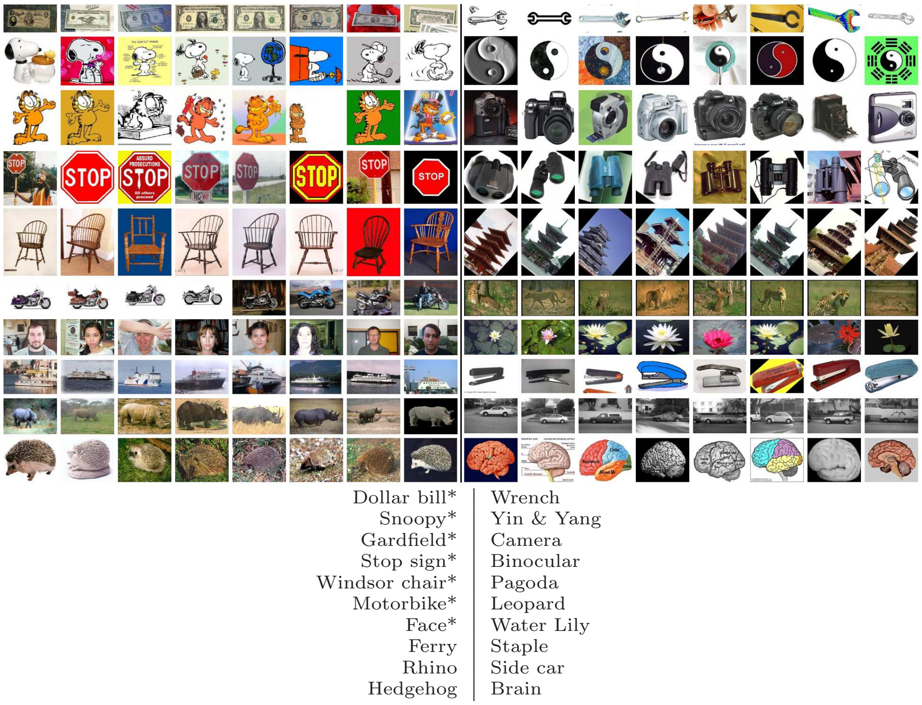

Figure 14: Samples of the Caltech-7 and Caltech-20 data sets. The categories in the Caltech-7 data set are marked with an asterisk.

seven and twenty object categories (we call these subsets Caltech-7 and Caltech-20). More in detail, the Caltech-7 data set includes the classes dollar bill, faces, Garfield, motorbikes, Snoopy, stop sign, and Windsor chair; the Caltech-20 data set includes the Caltech-7 classes, plus: binocular, brain, camera, car side, ferry, hedgehog, leopards, pagoda, rhino, stapler, water lily, wrench, and yin yang (see Figure 14). Finally, Lee and Grauman, and Faktor and Irani, exploited local, low level features (HOG and Local Self Similarity in the first, and SIFT-based in the latter) and perform clustering by identifying and matching local regions of the images. Images are linked via an affinity matrix coding the similarities between the images. Tests are performed on subsets of the Caltech-101 data set.

We evaluated the use of supervised features on the Caltech-4, Caltech-7 and Caltech-20 data sets following the procedure described by the authors. On

Table 12: Classification rates on the object data sets (%). For each data set, the best result is reported in bold.

Method	Caltech-4	Caltech-7	Caltech-20
k-means + Classemes	**98.3**	76.0	72.8
k-means + Prosemantic	89.8	64.9	52.1
k-means + Object bank	96.3	82.5	63.5
k-means + CCA-56	80.1	54.4	46.1
k-means + CCA-100	93.5	50.8	43.8
Grauman and Darrell [16]	85.0	-	-
Dueck and Frey [17]	-	58.9	36.8
Lee and Grauman [12]	91.1	-	65.6
Faktor and Irani [13]	-	**89.9**	**78.9**

the basis of the results obtained in the previous experiments, we used only the k-means algorithm. The number of clusters is four for the Caltech-4 data set. For the Caltech-7 and Caltech-20, Dueck and Frey report the results for a variable number of clusters between ten and 60. Here we consider ten clusters for Caltech-7, and twenty clusters for Caltech-20. Table 12 reports the results obtained.

Among the supervised features, the best results with the k-means clustering have been obtained by classemes and object bank features. With respect to other, more complex clustering techniques, the best results have been obtained by the approach of Faktor and Irani. This method, and like the Lee and Grauman one, uses low level features to determine local regions in the images that are common across images of the same category and group them accordingly. These approaches are quite different and more complex with respect to the ones considered in this paper. They are designed to infer the presence of relevant/foreground elements in the images and to use this information for the clustering. Thus, it is not surprising that the results on the Caltech data set of [13], which uses local information, are higher than the other methods that describe the image in a global way. It should be interesting to investigate if similar performances can be obtained on data sets of more complex scenes. Nevertheless, the combination of k-means with supervised features obtained the best performance on Caltech-4, and the second best on Caltech-7 and Caltech-20.

5. Summary and discussion

The most evident result of these experiments is that supervised features (in the sense in which we have defined them for the purpose of this paper) outperform primitive for unsupervised classification. In no experiment was the best result obtained by a primitive feature and in many cases the performance of supervised features is comparable to that of supervised methods. This seems to suggest that the information provided during the supervised training of the feature extractor is transferred to hitherto unseen categories. That is, supervised learning on a limited number of categories is enough to obtain very good performance on the unsupervised learning of new categories.

In the first three experiments, and across the four clustering algorithms, classemes and prosemantic features perform almost the same, with prosemantic performing slightly better (prosemantic "wins" 7 times, classemes 5). This is *prima facie* surprising, considering the great difference in the number of the base classes for the two cases (2659 for classemes, 14 for prosemantic), and it indicates that supervised features are extremely robust, and that they allow an accurate representation of images with small feature vectors (56 components, in the case of prosemantic).

A large scale experiment on the more than 100,000 images from the SUN data set demonstrated the scalability of the approach based on supervised features. In fact, the use of supervised features consistently allowed to obtain better results than those achieved with primitive features independently on the size of the unsupervised classification problem.

In the last experiment, both classemes and Object Bank features performed very well. In fact they obtained better results than the other methods from the state of the art, with the exception of the method by Faktor and Irani [13]. Prosemantic features have been designed to characterize whole images, while the Caltech data set has been collected for object classification tasks, and will therefore favor an approach based on local descriptors, objects classifiers (classemes), or object detectors (object bank features).

One interesting point is raised by the comparison of Object Bank and prosemantic features. Object Bank features are made of the output of object classifiers, while prosemantic features are composed, essentially, of scene-level descriptors. Quite unsurprisingly, Object Bank performs better than prosemantic on the Caltech database, which contains several rather artificial images depicting isolated objects on a uniform background, while prosemantic outperforms Object Bank on the scene recognition data set (classemes, containing both objects and scene classes, performs well in both cases). However, in data set less easily characterized, such as Simplicity or the event data set, prosemantic features constantly outperform Object Bank. One might envision here the embryo of a design directive for supervised features: a judicious mix of object and scene classes will probably give the best performance (but, quite probably, at the cost of a feature space of rather high dimensionality, as object recognition generalizes poorly from one object to another). If, however, one has to bias the base classes it appears that it is better to bias towards scene features rather than towards object features. The results of the application of a simple dimensionality reduction technique (PCA) demonstrate that the descriptive power of the features do not depends too much on the dimensionality of their spaces. Therefore, at least for unsupervised categorization, it is advisable to select a small number of classes when defining the supervised features. These are, of course, only early qualitative observations, as the experiments were designed simply with performance comparison in mind, and not in order to derive design directives.

Among primitive features, it is worth noticing that modern features (Fisher

Vectors, bag of SIFT, spatial pyramid) don't work significantly better than the standard MPEG-7 features, which have the additional advantage of creating much more compact representations.

6. Conclusion

In this paper we addressed the problem of unsupervised image categorization by using supervised features. The features we considered are derived from multiple image classifiers or object detectors trained to identify a set of semantic categories. Their capability of capturing the semantic content of the images make it possible to use standard clustering algorithm to automatically partition image collections into meaningful categories. This is demonstrated by our experiments where these features allowed to identify the ground truth categories in several data sets of variable difficulty.

On the basis of the results obtained we can conclude that supervised features are able to dynamically characterize new, unseen, categories which are quite different from those used to build them. So far, supervised features have been heuristically defined. In our future work we will take advantage of the insights provided by the results obtained here in order to address some open issues. In particular, we will investigate how to identify the categories that should be used for a specific task; how much the performance depends on the similarity between these and the target categories; and how many categories are required to obtain good results.

Appendix A. Additional data sets

In this appendix we report the classification rates obtained on two additional data sets. These data sets have been chosen to verify the limitation of supervised features. Since these features have been defined on labeled images depicting objects and/or scenes, it is possible that their descriptive power is severely reduced for other kinds of images. In particular we considered a data set of textures and one of aerial images.

Appendix A.1. KTH-TIPS2

This data set consists of 4752 images collected by capturing 44 samples of 11 different categories (see Figure A.15). Each sample has been captured 108 times at different scales and under different illumination conditions [61]. Table A.13 reports the classification rates obtained on this data set. It is not surprising that on this data set the gap between the performance of supervised and primitive features is quite small. Still, supervised features obtained the best results.

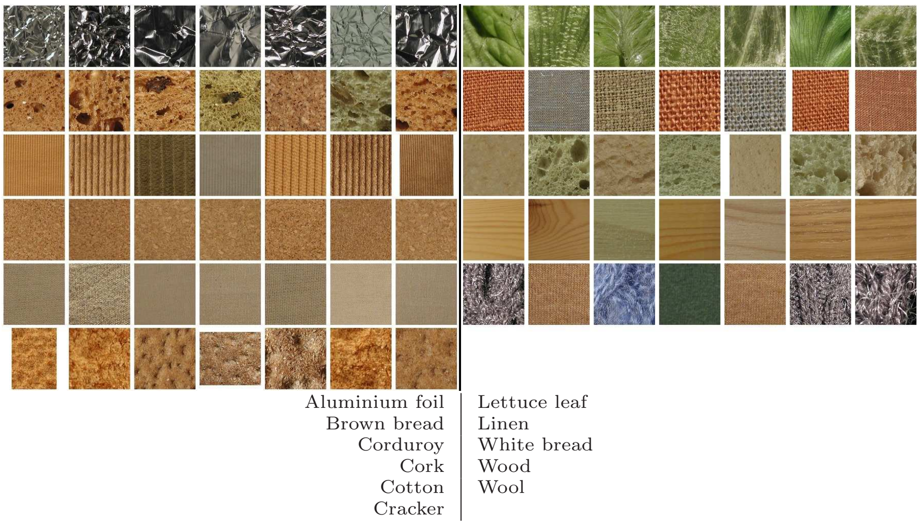

Figure A.15: Samples of the 11 classes of the KTH-TIPS2 data set.

Table A.13: Classification rates on the KTH-TIPS2 data set (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	52.7	58.6	49.3	48.2
Prosemantic	**53.9**	55.7	**56.4**	**54.6**
Object bank	46.4	44.0	42.7	39.3
CCA-56	49.4	**59.8**	44.7	43.5
Fisher Vectors	51.2	48.0	51.2	40.7
Gist	50.0	53.3	50.3	49.6
Bag of SIFT	22.1	51.8	21.1	25.7
Sp. Pyramid	33.1	36.6	30.7	31.5
CEDD	46.4	47.7	48.7	48.0
SCD	32.3	45.6	30.6	9.1
CLD	42.8	33.8	40.8	40.4
EHD	41.5	34.1	40.9	36.4

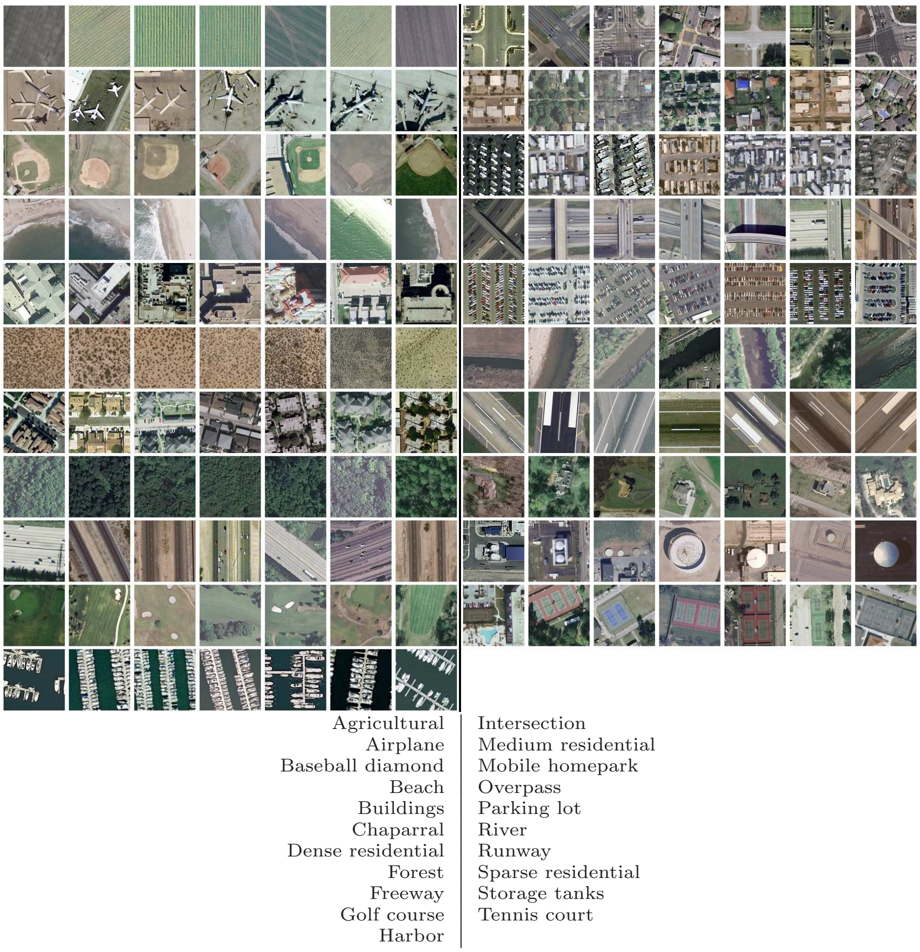

Figure A.16: Samples of the 21 classes of the Landuse data set.

Appendix A.2. UC Merced Land Use Dataset

This is a 21 class land use image data set [62]. There are 100 images for each of the following classes: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile homepark, overpass, parking lot, river, runway, sparse residential, storage tanks, tennis court (see Figure A.16). Table A.14 reports the classification rates obtained on this data set. On this data set the best performance have been obtained by using the Fisher Vectors. The other primitive features obtained classification rates slightly below those of the supervised ones.

Table A.14: Classification rates on the Landuse data set (%). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	36.8	36.7	38.8	35.0
Prosemantic	40.0	41.7	41.3	38.4
Object bank	37.1	34.4	38.2	31.2
CCA-56	40.0	40.7	34.6	32.3
Fisher Vectors	**54.1**	**51.6**	**52.2**	**42.0**
Gist	35.1	34.5	36.4	29.4
Bag of SIFT	26.5	39.1	26.0	24.8
Sp. Pyramid	21.62	25.90	19.57	21.6
CEDD	31.9	32.4	32.8	32.6
SCD	20.5	24.3	21.2	20.3
CLD	27.5	26.1	27.1	26.1
EHD	31.1	31.9	30.9	28.0

Appendix B. Conditional entropies

The conditional entropy is an alternative to the classification rate for the evaluation of unsupervised categorization [7]. For a joint distribution P over $X \times Y$ it is defined as:

$$H(Y|X) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log_2 \frac{1}{P(y|x)}. \tag{B.1}$$

Let N_{ij} be the number of images of category i that have been placed in the cluster j, then the conditional entropy can be estimated as follows:

$$H = \frac{1}{\sum_i \sum_j N_{ij}} \sum_i \sum_j N_{ij} \log_2 \frac{\sum_k N_{kj}}{N_{ij}}. \tag{B.2}$$

The conditional entropy is measured in bits. Tables B.15, B.16, and B.17 reports the conditional entropies obtained on the Simplicity, scene recognition, and event data sets.

References

[1] Y. Jing, M. Covell, H. Rowley, Comparison of clustering approaches for summarizing large populations of images, in: Proc. IEEE Int'l Conf. Multimedia and Expo, 2010, pp. 1523–1527.

[2] T. Liu, C. Rosenberg, H. Rowley, Clustering billions of images with large scale nearest neighbor search, in: Proc. IEEE Workshop Applications of Computer Vision, 2007, pp. 28–33.

Table B.15: Conditional entropies on the Simplicity data set (bits). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	*k*-means	Spectral	Ward	Affinity Prop.
Classemes	1.41	1.32	1.31	1.37
Prosemantic	**1.06**	**1.04**	**1.18**	**1.31**
Object bank	1.62	1.64	1.59	1.79
CCA-56	1.36	1.30	1.88	1.97
Fisher Vectors	1.81	1.81	1.69	2.35
Gist	2.52	2.46	2.54	2.62
Bag of SIFT	2.02	2.10	1.98	2.18
Spatial Pyramid	2.02	1.99	2.12	2.10
CEDD	1.53	1.52	1.46	1.64
SCD	2.32	2.37	2.42	2.34
CLD	1.71	1.70	1.75	1.84
EHD	2.19	2.32	2.06	2.44

Table B.16: Conditional entropies on the scene recognition data set (bits). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	*k*-means	Spectral	Ward	Affinity Prop.
Classemes	**1.05**	**0.93**	1.26	1.31
Prosemantic	1.08	1.13	**1.19**	**1.25**
Object bank	1.26	1.38	1.30	1.42
CCA-56	1.29	1.37	1.51	1.55
Fisher Vectors	2.14	2.29	2.15	2.42
Gist	1.68	1.61	1.70	1.90
Bag of SIFT	2.20	2.20	2.16	2.24
Spatial Pyramid	2.11	2.05	2.10	2.13
CEDD	2.20	2.18	2.31	2.33
SCD	2.70	2.64	2.66	2.68
CLD	2.54	2.53	2.52	2.57
EHD	1.58	1.55	1.62	1.82

[3] L. Torresani, M. Szummer, A. Fitzgibbon, Efficient object category recognition using classemes, in: Proc. European Conf. Computer Vision, 2010, pp. 776–789.

[4] G. Ciocca, C. Cusano, S. Santini, R. Schettini, Prosemantic features for content-based image retrieval, in: Adaptive Multimedia Retrieval 2009. Understanding Media and Adapting to the User, Vol. 6535 of Lecture Notes

Table B.17: Conditional entropies on the event data set (bits). For each algorithm, the best result is reported in bold.

	Clustering algorithm			
Features	k-means	Spectral	Ward	Affinity Prop.
Classemes	1.66	1.60	**1.50**	**1.75**
Prosemantic	1.55	1.55	1.62	1.86
Object bank	2.23	2.21	2.19	2.33
CCA-56	**1.45**	**1.46**	1.79	2.16
Fisher Vectors	2.40	2.32	2.37	2.51
Gist	2.17	2.15	2.06	2.22
Bag of SIFT	2.42	2.37	2.47	2.49
Spatial Pyramid	2.46	2.48	2.49	2.56
CEDD	2.18	2.18	2.15	2.24
SCD	2.67	2.67	2.69	2.67
CLD	2.40	2.35	2.46	2.36
EHD	2.13	2.14	2.23	2.25

in Computer Science, 2011, pp. 87–100.

[5] L. Li, H. Su, E. Xing, L. Fei-Fei, Object bank: A high-level image representation for scene classification and semantic feature sparsification, in: Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.

[6] A. Gordoa, J. A. Rodríguez-Serrano, F. Perronnin, E. Valveny, Leveraging category-level labels for instance-level image retrieval, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012, pp. 3045–3052.

[7] T. Tuytelaars, C. Lampert, M. Blaschko, W. Buntine, Unsupervised object discovery: A comparison, Int'l J. Computer Vision 88 (2) (2010) 284–302.

[8] J. Sivic, B. Russell, A. Efros, A. Zisserman, W. Freeman, Discovering objects and their location in images, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 370–377.

[9] J. Sivic, B. Russell, A. Zisserman, W. Freeman, A. Efros, Unsupervised discovery of visual object class hierarchies, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[10] D. Dai, T. Wut, S.-C. Zhu, Discovering scene categories by information projection and cluster sampling, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010, pp. 3455–3462.

[11] D. Dai, M. Prasad, C. Leistner, L. Van Gool, Ensemble partitioning for unsupervised image categorization, in: Proc. European Conf. on Computer Vision, 2012, pp. 483–496.

[12] Y. J. Lee, K. Grauman, Foreground focus: Unsupervised learning from partially matching images, Int'l J. Computer Vision 85 (2) (2009) 143–166.

[13] A. Faktor, M. Irani, Clustering by composition — unsupervised discovery of image categories, in: Proc. European Conf. on Computer Vision, 2012, pp. 474–487.

[14] T. Käster, V. Wendt, G. Sagerer, Comparing clustering methods for database categorization in image retrieval, in: Pattern Recognition, Vol. 2781 of LNCS, 2003, pp. 228–235.

[15] X. Zheng, D. Cai, X. He, W.-Y. Ma, X. Lin, Locality preserving clustering for image database, in: Proc. ACM Int'l Conf. Multimedia, 2004, pp. 885–891.

[16] K. Grauman, T. Darrell, Unsupervised learning of categories from sets of partially matching image features, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 1, 2006, pp. 19–25.

[17] D. Dueck, B. Frey, Non-metric affinity propagation for unsupervised image categorization, in: Proc. IEEE Int'l Conf. Computer Vision, 2007, pp. 1–8.

[18] N. Loeff, A. Farhadi, Scene discovery by matrix factorization, in: Proc. European Conf. on Computer Vision, 2008, pp. 451–464.

[19] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, Pattern Recognition 40 (1) (2007) 262–282.

[20] G. Wang, D. Hoiem, D. Forsyth, Learning image similarity from flickr groups using stochastic intersection kernel machines, in: Proc. IEEE Int'l Conf. Computer Vision, 2009, pp. 428–435.

[21] G. Ciocca, C. Cusano, S. Santini, R. Schettini, Halfway through the semantic gap: prosemantic features for image retrieval, Information Sciences 181 (22) (2011) 4943–4958.

[22] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. society for information science 41 (6) (1990) 391–407.

[23] M. Ranzato, F. Huang, Y. Boureau, Y. LeCun, Unsupervised learning of invariant feature hierarchies with applications to object recognition, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[24] C. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009, pp. 951–958.

[25] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009, pp. 1778–1785.

[26] H. Zhang, Z.-J. Zha, J. Yan, S.and Bian, T.-S. Chua, Attribute feedback, in: Proc. ACM Int'l Conf. Multimedia, 2012, pp. 79–88.

[27] F. Yu, R. Ji, M.-H. Tsai, G. Ye, S.-F. Chang, Weak attributes for large-scale image retrieval, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2012, pp. 2949–2956.

[28] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, Y. Gao, T.-S. Chua, Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval, in: Proc. ACM Int'l Conf. Multimedia, 2013, pp. 33–42.

[29] Z. M., Y. Y., Z. X., S. Y., N. Sebe, A. Hauptmann, Complex event detection via multi-source video attributes, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2013, pp. 2627–2633.

[30] J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval, Int'l J. Computer Vision 72 (2) (2007) 133–157.

[31] N. Rasiwasia, N. Vasconcelos, Scene classification with low-dimensional semantic spaces and weak supervision, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2008, pp. 1–6.

[32] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, J. Curtis, Large-scale concept ontology for multimedia, IEEE Multimedia 13 (3) (2006) 86–91.

[33] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: IEEE Int'l Conf. Computer Vision, 2009, pp. 221–228.

[34] A. Oliva, A. Torralba, Building the gist of a scene: The role of global image features in recognition, Progress in brain research 155 (2006) 23–36.

[35] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.

[36] A. Bosch, A. Zisserman, X. Munoz, Image classification using rois and multiple kernel learning, Int'l J. Computer Vision 2008 (2008) 1–25.

[37] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[38] D. Hoiem, A. Efros, M. Hebert, Automatic photo pop-up, ACM Trans. on Graphics 24 (3) (2005) 577–584.

[39] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3/4) (1936) 321–377.

[40] T. Sikora, The MPEG-7 visual standard for content description-an overview, IEEE Trans. Circuits and Systems for Video Technology 11 (6) (2001) 696–702.

[41] S. Chatzichristofis, Y. Boutalis, CEDD: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval, in: Computer Vision Systems, Vol. 5008 of LNCS, 2008, pp. 312–322.

[42] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, Int'l J. Computer Vision 42 (3) (2001) 145–175.

[43] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, Int'l J. Computer Vision 73 (2) (2007) 213–238.

[44] C. Wallraven, B. Caputo, A. Graf, Recognition with local features: the kernel recipe, in: Proc. IEEE Int'l Conf. Computer Vision, Vol. 1, 2003, pp. 257–264.

[45] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2161–2168.

[46] D. Lowe, Distinctive image features from scale-invariant keypoints, Int'l J. Computer Vision 60 (2) (2004) 91–110.

[47] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proc. European Conf. on Computer Vision, 2010, pp. 143–156.

[48] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 2169–2178.

[49] V. Pestov, On the geometry of similarity search: dimensionality curse and concentration of measure, Information Processing Letters 73 (1-2) (2000) 47–51.

[50] G. Hamerly, C. Elkan, Alternatives to the k-means algorithm that find better clusterings, in: Proc. Int'l Conf. on Information and Knowledge Management, 2002, pp. 600–607.

[51] B. Frey, D. Dueck, Clustering by passing messages between data points, Science 315 (5814) (2007) 972–976.

[52] J. J. Ward, Hierarchical grouping to optimize an objective function, J. the Am. Statistical Assoc. 58 (301) (1963) 236–244.

[53] U. Von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[54] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[55] J. Wang, J. Li, G. Wiederhold, Simplicity: Semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Analysis and Machine Intelligence 23 (9) (2001) 947–963.

[56] L. Li, L. Fei-Fei, What, where and who? classifying events by scene and object recognition, in: Proc. IEEE Int'l Conf. Computer Vision, 2007, pp. 1–8.

[57] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.

[58] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, Proc. IEEE Conf. Computer Vision and Pattern Recognition 2 (2003) 264.

[59] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, Computer Vision and Image Understanding 106 (1) (2007) 59–70.

[60] C. Fellbaum, Wordnet: An Electronic Lexical Database, Bradford Books, 1998.

[61] B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, in: Proc. IEEE Int'l Conf. on Computer Vision, Vol. 2, 2005, pp. 1597–1604.

[62] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: Proc. Int'l Conf. Advances in Geographic Information Systems, 2010, pp. 270–279.