Formulating Semantic Image Annotation as a Supervised Learning Problem

Gustavo Carneiro and Nuno Vasconcelos

Statistical Visual Computing Laboratory

SVCL UCSD

SVCL-TR 2004/03

December 2004

Supervised semantic image annotation

Gustavo Carneiro and Nuno Vasconcelos
Statistical Visual Computing Lab
Department of Electrical and Computer Engineering
University of California, San Diego

December 2004

Abstract

We introduce a new method to automatically annotate and retrieve images using a vocabulary of image semantics. Our main contribution resides in the use of a hierarchical description of the density of each of the image classes in the database of classes. This method has been shown to be well suited to problems involving large databases where groups of images can be combined into higher-level groups. Compared to current methods of image annotation and retrieval, ours has a significantly smaller time complexity for a better recognition performance. Specifically, the recognition complexity of our system is O(CxR), where C is the number of classes (or image annotations) and R is the number of image regions, while the best results in the literature are with a system that has complexity O(TxR), where T is the number of training images. Since the number of classes grows substantially slower than the number of training images, our method not only scales better for larger data set, but it also processes a test image faster. We show comparisons in terms of complexity, time, and recognition performance with the state-of-the-art methods proposed in the literature. The results illustrate that our system has a superior performance in terms of recognition accuracy for significantly smaller time complexity.

Author email: gcarneiro@ucsd.edu

SVCL Technical reports are available on the SVCL's web page at
http://www.svcl.ucsd.edu

University of California, San Diego
Statistical Visual Computing Laboratory
9500 Gilman Drive, Mail code 0407
EBU 1, Room 5512
La Jolla, CA 92093-0407

1 Introduction

Content-based image retrieval, the problem of searching large image repositories according to their content, has been the subject of a significant amount of computer vision research in the recent past [15]. While early retrieval architectures were based on the query-by-example paradigm, which formulates image retrieval as the search for the best database match to a user-provided query image, it was quickly realized that the design of fully functional retrieval systems would require support for semantic queries [13]. These are systems where the database of images are annotated with semantic keywords, enabling the user to specify the query through a natural language description of the visual concepts of interest. This realization, combined with the cost of manual image labeling, generated significant interest in the problem of automatically extracting semantic descriptors from images.

The earliest efforts in the area were directed to the reliable extraction of specific semantics, e.g. differentiating indoor from outdoor scenes [16], cities from landscapes [17], and detecting trees [8], horses [6], or buildings [11], among others. These efforts posed the problem of semantics extraction as one of supervised learning: a set of training images with and without the concept of interest was collected and a binary classifier trained to detect the concept of interest. The classifier was then applied to all database of images which were, in this way, annotated with respect to the presence or absence of the concept. Since each classifier is trained in the “one-vs-all” (OVA) mode (concept of interest vs everything else), we refer to this semantic labeling framework as *supervised OVA*.

More recently, there has been an effort to solve the problem in its full generality, by resorting to unsupervised learning [1, 5, 2, 14, 7]. The basic idea is to introduce a set of latent variables that encode hidden states of the world, where each state defines a joint distribution on the space of semantic keywords and image appearance descriptors (in the form of local features computed over image neighborhoods). During training, a set of labels is assigned to each image, the image is segmented into a collection of regions (either through a block-based decomposition [3, 14, 10] or through traditional image segmentation mathods [1, 2, 5, 9]), and an unsupervised learning algorithm is run over the entire database to estimate the joint density of words and visual features. Given a new image to annotate, visual feature vectors are extracted, the joint probability model is instantiated with those feature vectors, state variables are marginalized, and a search for the set of labels that maximize the joint density of text and appearance is carried out. We refer to this labeling framework as *unsupervised*.

Both formulations of the semantic labeling problem have strong advantages and disadvantages. In generic terms, unsupervised labeling leads to significantly more scalable (in database size and number of concepts of interest) training procedures, places much weaker demands on the quality of the manual annotations required to bootstrap learning, and produces a natural ranking of keywords for each new image to annotate. On the other hand, it does not explicitly treat semantics as image classes and, therefore, provides little guarantees that the semantic annotations are optimal in a recognition or retrieval sense. That is, instead of annotations that achieve the smallest probability of retrieval error, it simply produces the ones that have largest joint likelihood under the assumed mixture model.

In this work we show that it is possible to combine the advantages of the two formulations through a slight reformulation of the supervised one. This consists of defining an M-ary classification problem where each of the semantic concepts of interest defines an image class. At annotation time, these classes all directly compete for the image to annotate, which no longer faces a sequence of independent binary tests. This *supervised M-ary* formulation obviously retains the classification and retrieval optimality of supervised OVA, but 1) produces a natural ordering of keywords at annotation time, and 2) eliminates the need to compute a "non-class" model for each of the semantic concepts of interest. In result, it has learning complexity equivalent to that of the unsupervised formulation and, like the latter, places much weaker requirements on the quality of manual labels than supervised OVA.

There are, nevertheless, two important questions regarding the feasibility of the practical implementation of the supervised M-ary formulation. The first, is that of how to learn a probability distribution for a semantic class, from images that are only weakly labeled with respect to that class. That is, images that are labeled as containing the semantic concept of interest, but contain no indication about which image regions are observations of that concept. The second is that of how to learn these distributions in a computationally efficient manner while accounting for all data that is available from each semantic class. We show that both questions can be partially solved by naive model averaging, i.e. by simply averaging all the distributions of images labeled as containing the semantic concept. This is shown to be remarkably similar to the estimation of the joint model underlying the unsupervised labeling formulation. It turns out, however, that model averaging has major inconvenient from the point of view of 1) the quality of the density estimates obtained for the semantic classes, and 2) the computational complexity of the annotation procedure. We note that both problems can be eliminated with recourse to a hierarchical density model proposed in [18] for image indexing purposes. It is shown that, by adopting this probabilistic model, it is feasible to 1) learn semantic class densities with complexity equivalent to that of the unsupervised formulation, 2) obtain semantic density estimates significantly more reliable than those available by model averaging, and 3) achieve significantly greater computational efficiency in what regards to image annotation. The method now proposed is compares to the state-of-the-art methods of [14, 9] using the experimental setup introduced in [5]. The results show that that the approach now proposed has advantages not only in terms of annotation and retrieval accuracy, but also in terms of efficiency.

2 Supervised vs. Unsupervised Semantic Annotation

The goal of semantic image labeling is to, given an image $\mathcal{I}$, extract, from a vocabulary $\mathcal{L}$ of semantic descriptors, the set of keywords, or captions, $\mathbf{w}$ that best describes $\mathcal{I}$. Learning is based on a training set $\mathcal{D} = \{(\mathcal{I}_1, \mathbf{w}_1), \ldots, (\mathcal{I}_D, \mathbf{w}_D)\}$ of image-caption pairs. The training set is said to be weakly labeled if the absence of a keyword from caption $\mathbf{w}_i$ does not necessarily mean that the associated concept is not present in $\mathcal{I}_i$. For example, an image containing "sky" may not be explicitly labeled with that keyword. This is usually the case in practical scenarios, since each image is likely to be annotated with a small caption that only identifies the semantics deemed as most relevant to the labeller.

2.1 Supervised OVA Labeling

Let $\mathcal{L} = \{w_1, \ldots, w_L\}$ be the vocabulary of semantic labels, or keywords, w_i. Under the supervised OVA formulation, labeling is formulated as a collection of L detection problems that determine the presence/absence of the concepts w_i in the image $\mathcal{I}$. Consider the i^{th} such problem and the random variable Y_i such that

$$Y_i = \begin{cases} 1, & \text{if } \mathcal{I} \text{ contains concept } w_i \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Given a collection of image features $\mathbf{X}$ extracted from $\mathcal{I}$, the goal is to infer the state of Y_i with smallest probability of error, for all i. This can be solved by application of standard Bayesian decision theory, namely by declaring the concept as present if

$$P_{\mathbf{X}|Y_i}(\mathbf{x}|1)P_Y(1) \geq P_{\mathbf{X}|Y_i}(\mathbf{x}|0)P_Y(0) \tag{2}$$

where $P_{\mathbf{X}|Y_i}(\mathbf{x}|j)$ is the class-conditional density and $P_Y(i)$ the prior probability for class $j \in \{0, 1\}$.

Training consists of assembling a training set $\mathcal{D}_1$ containing all images labeled with the concept w_i, a training set $\mathcal{D}_0$ containing the remaining images, and using some density estimation procedure to estimate $P_{\mathbf{X}|Y_i}(\mathbf{x}|j)$ from $\mathcal{D}_j$, $j \in \{0, 1\}$. Note that any images containing concept w_i which are not explicitly annotated with this concept are incorrectly assigned to $\mathcal{D}_0$ and can compromise the classification accuracy. In this sense, the supervise OVA formulation is not amenable to weak labeling. Furthermore, the set $\mathcal{D}_0$ is likely to be quite large when the vocabulary size L is large and the training complexity is dominated by the complexity of learning the conditional density for $Y = 0$.

Applying the process of (2) to the query image $\mathcal{I}$, produces a sequence of labels $\hat{w}_i \in \{0, 1\}, i \in \{1, \ldots, L\}$, and a set of posterior probabilities $P_{Y_i|\mathbf{X}}(1|\mathbf{x})$ that can be taken as degrees of confidence on the annotation. Notice, however, that these are posterior probabilities relative to different classification problems and do not establish a natural ordering of importance of the keywords w_i as descriptors of $\mathcal{I}$. Nevertheless, the binary decision regarding whether each concept is present in the image or not is a minimum probability of error decision.

2.2 Unsupervised Labeling

The basic idea underlying the unsupervised learning formulation [1, 5, 2, 14, 7] is to introduce a variable L that encodes hidden states of the world. Each of these states then defines a joint distribution for keywords and image features. The various methods differ in the definition of the states of the hidden variable: some associate a state to each image in the database [14, 9], others associate them with image clusters [1, 5, 2]. The overall model is of the form

$$P_{\mathbf{X},\mathbf{W}}(\mathbf{x}, \mathbf{w}) = \sum_{l=1}^{S} P_{\mathbf{X},\mathbf{W}|L}(\mathbf{x}, \mathbf{w}|l)P_L(l) \tag{3}$$

where S is the number of possible states of L, $\mathbf{X}$ the set of feature vectors extracted from $\mathcal{I}$ and $\mathbf{W}$ the vector of keywords associated with this image. Since this is a mixture model, learning is usually based on the expectation-maximization (EM) [4] algorithm, but the details depend on the particular definition of hidden variable and probabilistic model adopted for $P_{\mathbf{X},\mathbf{W}}(\mathbf{x},\mathbf{w})$.

The simplest model in this family [14, 9], which has also achieved the best results in experimental trials, makes each image in the training database a state of the latent variable, and assumes conditional independence between image features and keywords, i.e.

$$P_{\mathbf{X},\mathbf{W}}(\mathbf{x},\mathbf{w}) = \sum_{l=1}^{D} P_{\mathbf{X}|L}(\mathbf{x}|l)P_{\mathbf{W}|L}(\mathbf{w}|l)P_L(l) \tag{4}$$

where D is the training set size. This enables individual estimation of $P_{\mathbf{X}|L}(\mathbf{x}|l)$ and $P_{\mathbf{W}|L}(\mathbf{w}|l)$, as is common in the probabilistic retrieval literature [15], therefore eliminating the need to iterate the EM algorithm over the entire database (a procedure of large computational complexity). In this way, the training complexity is equivalent to that of learning the conditional densities for $Y_i = 1$ in the supervised OVA formulation. This is significantly smaller than the learning complexity of that formulation (which, as discussed above, is dominate by the much more demanding task of learning the conditionals for $Y_i = 0$). The training of the $P_{\mathbf{W}|L}(\mathbf{w}|l)$, $l \in \{1, \ldots, D\}$ consists of a maximum likelihood estimate based on the annotations associated with the l^{th} training image, and usually reduces to counting [14, 9]. Note that, while the quality of the estimates improves when the image is annotated with all concepts that it includes, it is possible to compensate for missing labels by using standard Bayesian (regularized) estimates [14, 9]. Hence, the impact of weak labeling is not major under this formulation.

At annotation time, the feature vectors extracted from the query $\mathcal{I}$ are used in (3) to obtain a function of $\mathbf{w}$ that provides a natural ordering of the relevance of all possible captions for the query. This function can be the joint density of (3) or the posterior density

$$P_{\mathbf{W}|\mathbf{X}}(\mathbf{w}|\mathbf{x}) = \frac{P_{\mathbf{X},\mathbf{W}}(\mathbf{x},\mathbf{w})}{P_{\mathbf{X}}(\mathbf{x})}. \tag{5}$$

Note that, while this can be interpreted as the Bayesian decision rule for a classification problem with the states of $\mathbf{W}$ as classes, such class structure is not consistent with the generative model of (3) which enforces a causal relationship from L to $\mathbf{W}$. This leads to a very weak dependency between the observation $\mathbf{X}$ and class $\mathbf{W}$ variables, e.g. that they are independent given L in the model of (4). Therefore, in our view, this formulation imposes a mismatch between the class structure used for the purposes of designing the probabilistic models (where the states of the hidden variable are the dominant classes) and that used for labeling (which assume the states of $\mathbf{W}$ to be the real classes). This implies that the annotation decisions are not optimal in a minimum probability of error sense.

3 Supervised M-ary Labeling

The supervised M-ary formulation now proposed addresses this problem by explicitly making the elements of the semantic vocabulary the classes of the M-ary classification problem. That is, by introducing 1) a random variable W, which takes values in $\{1, \ldots, L\}$, so that $W = i$ if and only if $\mathbf{x}$ is a sample from the concept w_i, and 2) a set of class-conditional distributions $P_{\mathbf{X}|W}(\mathbf{x}|i), i \in \{1, \ldots, L\}$ for the distribution visual features given the semantic class. Similarly to supervised OVA, the goal is to infer the state of W with smallest probability of error. Given a set of features $\mathbf{x}$ from a query image $\mathcal{I}$ this is accomplished by application of the Bayes decision rule

$$i^* = \arg\max_i P_{\mathbf{X}|W}(\mathbf{x}|i) P_W(i) \tag{6}$$

where $P_W(i)$ is a prior probability for the i^{th} semantic class. The difference with respect to the OVA formulation is that instead of a sequence of L binary detection problems, we now have a single M-ary problem with L classes.

This has several advantages. First, there is no longer a need to estimate L non-class distributions ($Y_i = 0$ in (1)), an operation which, as discussed above, is the computational bottleneck of the OVA formulation. On the contrary, as will be shown in Section 4, it is possible to estimate all semantic densities $P_{\mathbf{X}|W}(\mathbf{x}|i)$ with computation equivalent to that required to estimate one density per image. Hence, the supervised M-ary formulation has learning complexity equivalent to the simpler of the unsupervised labeling approaches (4).

Second, the i^{th} semantic class density is estimated from a training set $\mathcal{D}_i$ containing all feature vectors extracted from images labeled with concept w_i. While this will be most accuracte if all images that contain the concept include w_i in their captions, images for which this keyword is missing will simply not be considered. If the number of images correctly annotated is large, this is likely not to make any practical difference. If that number is small, missing labeled images can always be compensated for by adopting Bayesian (regularized) estimates. In this sense, the supervised M-ary formulation is equivalent to the unsupervised formulation and, unlike the supervised OVA formulation, not severely affected by weak labeling.

Finally, at annotation time, the supervised M-ary formulation provides a natural ordering of the semantic classes, by the posterior probability $P_{W|\mathbf{X}}(w|\mathbf{x})$. Unlike the OVA case, under the M-ary formulation these posteriors are relative to the same classification problem, a problem where the semantic classes compete to explain the query. This ordering is, in fact, equivalent to that adopted by the unsupervised learning formulation (5), but now leads to a Bayesian decision rule that is matched to the class structure of the underlying generative model. Hence, this concept ordering is optimal in a minimum probability of error sense.

4 Estimation of Semantic Class Distributions

Given the collection of semantic class-conditional densities $P_{W|\mathbf{X}}(w|\mathbf{x})$, supervised M-ary labeling is relatively trivial (it consists of a search for the solution of (6)). Two interesting questions arise, however, in the context of density estimation.

4.1 Modeling Classes Without Segmentation

So far, we have assumed that all samples in the training set $\mathcal{D}_i$ are from concept w_i. In practice, however, this would require careful segmentation and labeling of all training images. While concepts such as "Indoor", "Outdoor", "Coastline", or "Landscape" tend to be holistic (i.e. the entire image is, or is not, in the class), most concepts refer to objects and other items that only cover a part of any image (e.g. "Bear", "Flag", etc.). Hence, most images contain a combination of various concepts. The creation of a training set $\mathcal{D}_i$ of feature vectors exclusively drawn from the i^{th} class would require manual segmentation of all training images, followed by labelling of the individual segments.

Since this is unfeasible, an interesting question is whether it is possible to estimate the class-conditional density from a training set composed of images with a significant percentage of feature vectors drawn from other classes. The answer to this question is afirmative, a fact that is known in the machine learning literature, where it is the basis of so-called *multiple instance* learning.Here we provide an illustrative example of how this principle applies to the process of learning semantic class densities. We assume, for simplicity, that the training set consists of images that contain three semantic concepts, each with probability $1/3$ (i.e. occupying $1/3$ of the image area), and that those concepts are Gaussian distributed, i.e. the distribution of image k is a mixture of three Gaussians. Introducing a hidden variable L for the image number, this distribution can be written as

$$P_{X|L}(x|l) = \frac{1}{3}\sum_{i=1}^{3}\mathcal{G}(x, \mu_i^l, \sigma_i^l).$$

The distribution over the ensemble of D training images is

$$P_X(x) = \sum_l P_{X|L}(x|l)P_L(l) \quad (7)$$

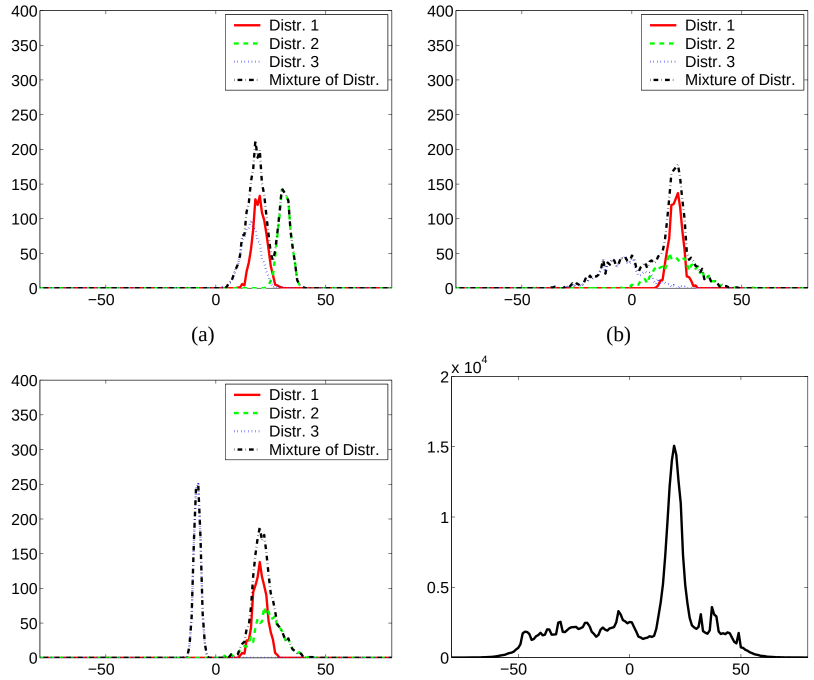
$$= \frac{1}{3D}\sum_{l=1}^{D}\sum_{i=1}^{3}\mathcal{G}(x, \mu_i^l, \sigma_i^l). \quad (8)$$

We next assume that one of the three components (e.g. the first, for simplicity) is always the density of concept w , e.g. $\mu_1^l = 20$ and $\sigma_1^l = 3, \forall l$, while the others are randomly selected from a pool of distributions that can have uniformly distributed mean μ and standard deviation σ. Under this assumption, as $D \to \infty$,

$$P_X(x) = \frac{1}{3}\mathcal{G}(x, 20, 3) + \frac{2}{3}\int \mathcal{G}(x, \mu, \sigma)p_\mu(\mu)p_\sigma(\sigma)d\mu d\sigma.$$

While the first term is the density of w the second term is an average of many Gaussians of different mean and covariance and converges to a uniform distribution that, in order to integrate to one, must have very small amplitude. Hence

$$\lim_{D\to\infty} P_X(x) = \frac{1}{3}\mathcal{G}(x, 20, 3) + \frac{2\kappa}{3}$$



(a) (b)

(c) (d)

Figure 1: Illustrative example of the process of learning semantic class densities. Graphs (a)-(c) show three examples of the feature distribution present in an image. Graph (d) presents the feature distribution over 1,000 images.

with $\kappa \sim 0$. Figure 1 presents a simulation of this effect, when $\mu \in [-100, 100]$ and $\sigma \in [0.1, 10]$ and the ensemble contains $1,000$ training images. The distribution of concept w dominates the probability density $P_X(x)$ when the full training ensemble is taken into account, even though it is never dominant for any individual image. Note that, because the uniform component has to integrate to one, its amplitude decreases exponentially with the dimension of the feature space. Hence, in high-dimensional spaces, the behavior of Figure 1 (d) is observed even when the number of training images is relatively small.

4.2 Density Estimation

Given the training set $\mathcal{D}_i$ of images containing concept w_i, the estimation of the density $P_{\mathbf{X}|W}(\mathbf{x}|i)$ can proceed in four different ways: *direct estimation*, *model averaging*, *naive averaging*, *hierarchical estimation*.

Direct Estimation

Direct estimation consists of estimating the class density from a training set containing all feature vectors from all images in $\mathcal{D}$. The main disadvantage of this strategy is that, for classes with a sizeable number of images, the training set is likely to be quite large. This creates a number of practical problems, e.g. the requirement for large amounts of memory, and makes sophisticated density estimation techniques unfeasible. One solution is to discard part of the data, but this is suboptimal in the sense that important training cases may be lost. We have, so far, not been able to successfully apply this strategy.

Model Averaging

Model averaging exploits (7) to overcome the computational complexity of direct estimation. It performs the estimation of $P_{\mathbf{X}|W}(\mathbf{x}|i)$ in two steps. In the first step, a density estimate is produced for each image, originating a sequence $P_{\mathbf{X}|L,W}(\mathbf{x}|l,i), l \in \{1, \ldots D\}$ where L is a hidden variable that indicates the image number. The class density is then obtained by averaging the densities in this sequence

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D}\sum_{l} P_{\mathbf{X}|L,W}(\mathbf{x}|l,i). \tag{9}$$

Note that this is equivalent to the density estimate obtained under the unsupervised labeling framework, if the text component of the joint density of (3) is marginalized and the hidden states are images (as is the case of (4)). The main difference is that, while under M-ary supervised labeling the averaging is done only over the set of images that belong to the semantic class, under unsupervised labeling it is done over the entire database. This, once again, reflects the lack of classification optimality of the later formulation.

The direct application of (9) is feasible when the densities $P_{\mathbf{X}|L,W}(\mathbf{x}|l,i)$ are defined over a (common) partition of the feature space. For example, if all densities are histograms defined on a partition of the feature space $\mathcal{X}$ into Q cells $\{\mathcal{X}_q\}, q = 1, \cdots, Q$, and $h_{i,j}^q$ the number of feature vectors from class i that land on cell $\mathcal{X}_q$ for image j, then the average class histogram is simply

$$\hat{h}_i^q = \frac{1}{D}\sum_{j} h_{i,j}^q$$

However, when 1) the underlying partition is not the same for all histograms or 2) more sophisticated models (e.g. mixture or non-parametric density estimates) are used model averaging is not as simple.

Naive Averaging

Consider, for example, the Gauss mixture model

$$P_{\mathbf{X}|L,W}(\mathbf{x}|l,i) = \sum_{k} \pi_{i,l}^k \mathcal{G}(\mathbf{x}, \mu_{i,l}^k, \Sigma_{i,l}^k), \tag{10}$$

where $\pi_{i,l}^k$ is a probability mass function such that $\sum_k \pi_{i,l}^k = 1$. Direct application of (9) leads to

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D}\sum_{k,l} \pi_{i,l}^k \mathcal{G}(\mathbf{x}, \mu_{i,l}^k, \Sigma_{i,l}^k) \tag{11}$$

i.e. a D-fold increase in the number of Gaussian components per mixture. Since, at annotation time, this probability has to be evaluated for each semantic class, it is clear that straightforward model averaging will lead to an extremely slow annotation process.

Mixture hierarchies

One efficient alternative to the complexity of model averaging is to adopt a hierachical density estimation method first proposed in [18] for image indexing. This method is based on a mixture hierarchy which, roughly speaking, is a collection of mixtures organized hierarchically. Under this hirarchical organization, children densities consist of different combinations of subsets of the parents components. A formal definition is given in [18], we omit the details for brevity. The important point is that, when the densities conform to the mixture hierarchy model, it is possible to estimate the parameters of the class mixture directly from those available for the individual image mixtures, using a two-stage procedure. The first stage, is the naive averaging of (11). Assuming that each mixture has K components, this leads to an overall mixture with DK components of parameters

$$\{\pi_j^k, \mu_j^k, \Sigma_j^k\}, j = 1, \ldots, D,\ k = 1, \ldots, K. \tag{12}$$

The second is an extension of the EM algorithm, which clusters the Gaussian components into a T-component mixture, where T is the number of components at the class level. Denoting by $\{\pi_c^t, \mu_c^t, \Sigma_c^t\}, t = 1, \ldots, T$ the parameters of the class mixture, this algorithm iterates between the following steps.
E-step: compute

$$h_{jk}^t = \frac{\left[\mathcal{G}(\mu_j^k, \mu_c^t, \mathbf{\Sigma}_c^t) e^{-\frac{1}{2} trace\{(\mathbf{\Sigma}_c^t)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N} \pi_c^t}{\sum_l \left[\mathcal{G}(\mu_j^k, \mu_c^l, \mathbf{\Sigma}_c^l) e^{-\frac{1}{2} trace\{(\mathbf{\Sigma}_c^l)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N} \pi_c^l}, \tag{13}$$

where N is a user-defined parameter (see [18] for details).
M-step: set

$$(\pi_c^t)^{new} = \frac{\sum_{jk} h_{jk}^t}{PK} \tag{14}$$

$$(\mu_c^t)^{new} = \sum_{jk} w_{jk}^t \mu_j^k, \text{where } w_{jk}^t = \frac{h_{jk}^t \pi_j^k}{\sum_{jk} h_{jk}^t \pi_j^k} \tag{15}$$

$$(\mathbf{\Sigma}_c^t)^{new} = \sum_{jk} w_{jk}^t \left[\mathbf{\Sigma}_j^k + (\mu_j^k - \mu_c^t)(\mu_j^k - \mu_c^t)^T\right]. \tag{16}$$

Notice that the number of parameters in each image mixture is orders of magnitude smaller than the number of feature vectors in the image itself. Hence the complexity of estimating the class mixture parameters is negligible when compared to that of estimating the individual mixture parameters for all images in the class. It follows that the overall training complexity is dominated by the latter task, i.e. only marginally superior to that of naive averaging and significantly smaller than that associated with direct estimation of class densities. On the other hand, the complexity of evaluating likelihoods is exactly the same as that achievable with direct estimation, and significantly smaller than that of naive averaging.

One final interesting property of the EM steps above is that they enforce a data-driven form of regularization which improves generalization. This regularization is visible in (16) where the variances on the left hand-size can never be smaller than those on the right-hand side. We have observed that, due to this property, hierarchical class density estimates are much more reliable than those obtained with direct learning.

5 Experimental Results

In this section, we present the experimental results on the Corel data set used in [5, 9, 14]. We focused on this experimental setup since it has been continuously adopted as a standard way to assess annotation and retrieval performance. It is difficult to compare the method now proposed to others that did not adopt this setup since it is impossible to implement all those other methods.

The Translation Model of [5] was the first milestone in the area of semantic annotation, in the sense of demonstrating results of practical interest. After various years of research, and several other contributions, the best existing results are, to our knowledge, those presented in [14]. We therefore adopt an evaluation strategy identical to that used in this work. The data set used in all experiments consists of $5,000$ images from 50 Corel Stock Photo CDs, and was divided into two parts: a training set of $4,500$ images and a test set of 500 images. Each CD includes 100 images of the same topic, and each image is associated with 1-5 keywords. Overall there are 371 keywords in the dataset. In all cases, the YBR color space was adopted, and the image features were coefficients of the 8×8 discrete cosine transform (DCT). Note that this is a feature set different that that used in [5, 9, 14], which consists of color, texture, and shape features.

5.1 Automatic Image Annotation

We start by assessing the performance of our model on the task of automatic image annotation. Given an un-annotated image, the task is to automatically generate a caption which is then compared to the annotation made by a human. Similarly to [9, 14] we define the automatic annotation to consist of the five classes under which the image has largest likelihood. We then compute the recall and precision of every word in the test set. Given a particular semantic descriptor w, if there are $|w_H|$ human annotated images with the descriptor w in the test set, and the system annotates $|w_{\text{auto}}|$ images with that descriptor, where $|w_C|$ are correct, recall and precision are given by $recall = \frac{|w_C|}{|w_H|}, precision = \frac{|w_C|}{|w_{\text{auto}}|}$.

We report results obtained on the complete set of 260 words that appear in the test set in Table 2, where the values of recall and precision are averaged over the set of testing words, as suggested by [9, 14]. Also presented are results (borrowed from [9, 14])

Table 1: Performance comparison of automatic annotation on the Corel dataset. Legend: CO = Co-occurrence, TR = translation, CRMR = CRM-rect, CRMD = CRM-rect-DCT, MH = Mix-Hier

Models	CO	TR	CRM	CRMR	MBRM	CRMD	MH
$\#$words recall > 0	19	49	107	119	122	107	137
Results on all 260 words							
Mn/word Recall	0.02	0.04	0.19	0.23	0.25	0.22	0.29
Mn/word Precision	0.03	0.06	0.16	0.22	0.24	0.21	0.23

Table 2: Performance comparison of automatic annotation on the Corel dataset.Legend: CO = Co-occurrence, TR = translation, CRMR = CRM-rect, CRMD = CRM-rect-DCT, MH = Mix-Hier

Models	CO	TR	CRM	CRMR	MBRM	CRMD	MH
$\#$words recall > 0	19	49	107	119	122	107	137
Results on all 260 words							
Mn/word Recall	0.02	0.04	0.19	0.23	0.25	0.22	0.29
Mn/word Precision	0.03	0.06	0.16	0.22	0.24	0.21	0.23

obtained with various other methods under this same experimental setting. Specifically, we consider: the Co-occurrence Model [12], the Translation Model [5],The Continuous-space Relvance Model (CRM-rect)[9, 14], and the Multiple-Bernoulli Relevance Model (MBRM) [14]. The method now proposed is denoted by 'Mix-Hier'. In order to guarantee a fair comparison we also implemented the CRM-rect using the $8x8$ DCT features. These results are presented in the column denoted as 'CRM-rect-DCT'.

Ovearll, the method now proposed achieves the best performance. When compared to the previous best results (MBRM) it exhibts a gain of 16% in recall for an equivalent level of precision. Similarly, the number of words with positive recall increases by 15%. It is also worth noting that the CRM-rect model with DCT features, performs slightly worse than the original CRM-rect. This indicates that the performance of Mix-Hier may improve with a better set of features. We intent to investigate this in the future.

Another important issue is the complexity of the annotation process. The complexity of CRM-rectangles and MBRM is $O(TR)$, where T is the number of training images and R is the number of image regions. Compared to those methods, Mix-Hier has a significantly smaller time complexity of $O(CR)$, where C is the number of classes (or image annotations). Assuming a fixed number of regions R, Fig. 2 shows how the annotation time of a test image grows for Mix-Hier and MBRM, as a function of the number of training images. In our experiments, over the set of 500 test images, the average annotation time was 268 seconds for Mix-Hier, and 371 seconds for CRM-rect-DCT.

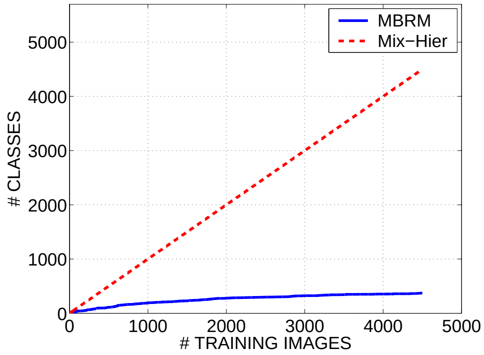


Figure 2: Comparison of the time complexity for the annotation of a test image on the Corel data set.

Table 3: Retrieval results on Corel.

Mean Average Precision for Corel Dataset		
Models	All 260 words	Words with recall > 0
Mix-Hier	0.31	0.49
MBRM	0.30	0.35

5.2 Image Retrieval with Single Word Queries

In this section we analize the performance of semantic retrieval. In this case, the precision and recall measures are computed as follows. If the n most similar images to a query are retrieved, recall is the percentage of all relevant images that are contained in that set and precision the percentage of the n which are relevant (where relevant means that the ground-truth annotation of the image contains the query descriptor). Once again, we adopted the experimental setup of [14]. Under this set-up, retrieval performance is evaluated by the mean average precision. As can be sen from Table 3, for ranked retrieval on Corel, Mix-Hier produces results superior to those of MBRM. In particular, it achieves a gain of 40% mean average precision on the set of words that have positive recall.

5.3 Results: Examples

In this section we present some examples of the annotations produced by our system. Fig. 3 illustrates the fact that, as reported in Table 3, Mix-Hier has a high level of recall. Frequently, when the system annotates an image with a descriptor not contained in the human-made caption, this annotation is not necessarily wrong. Finally, Figure 4 illustrates the performance of the system on one word queries.

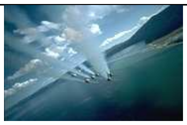
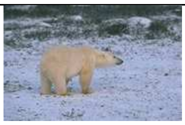
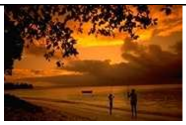
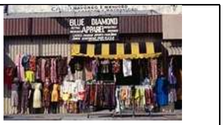
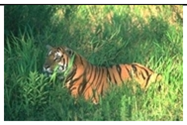
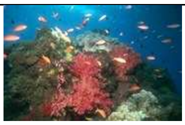
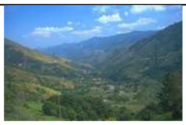
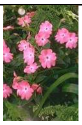
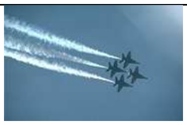
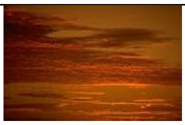
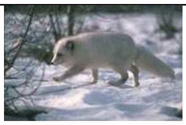
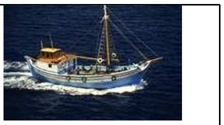
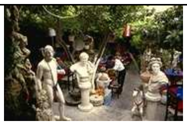
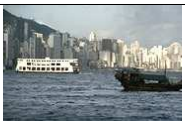
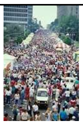
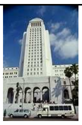
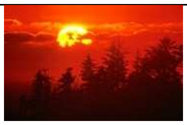
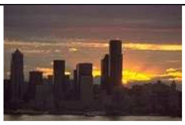
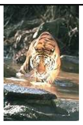
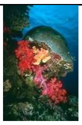
Model				
Human Annotation	sky jet plane smoke	bear polar snow tundra	water beach people sunset	buildings clothes shops street
Automatic Annotation	smoke clouds plane jet flight	polar tundra bear snow ice	sunset sun palm clouds sea	buildings street shops people skyline
Model				
Human Annotation	grass forest cat tiger	coral fish ocean reefs	mountain sky clouds tree	leaf flowers petals stems
Automatic Annotation	cat tiger plants leaf grass	reefs coral ocean fan fish	mountain valley sky clouds tree	petals leaf flowers lily stems
Model				
Human Annotation	sky jet plane smoke	sky clouds formation sunset	snow fox arctic	water boats waves
Automatic Annotation	plane jet smoke flight prop	sea sun sunset waves horizon	arctic snow polar fox ice	coast waves boats water oahu
Model				
Human Annotation	tree restaurant street statue	water boats harbor skyline	people street cars festival	sky buildings street cars
Automatic Annotation	statue street tree buildings castle	skyline boats coast shore water	street cars village buildings people	street buildings bridge sky arch
Model				
Human Annotation	sky sun clouds tree	city sun water	water rocks cat tiger	coral ocean reefs
Automatic Annotation	sun sea sunset clouds horizon	sun sunset city horizon clouds	rocks cat tiger water shore	ocean coral reefs fish fan

Figure 3: Semantic annotations on Corel. Comparison between automatically generated labels and those produced by a human subject.

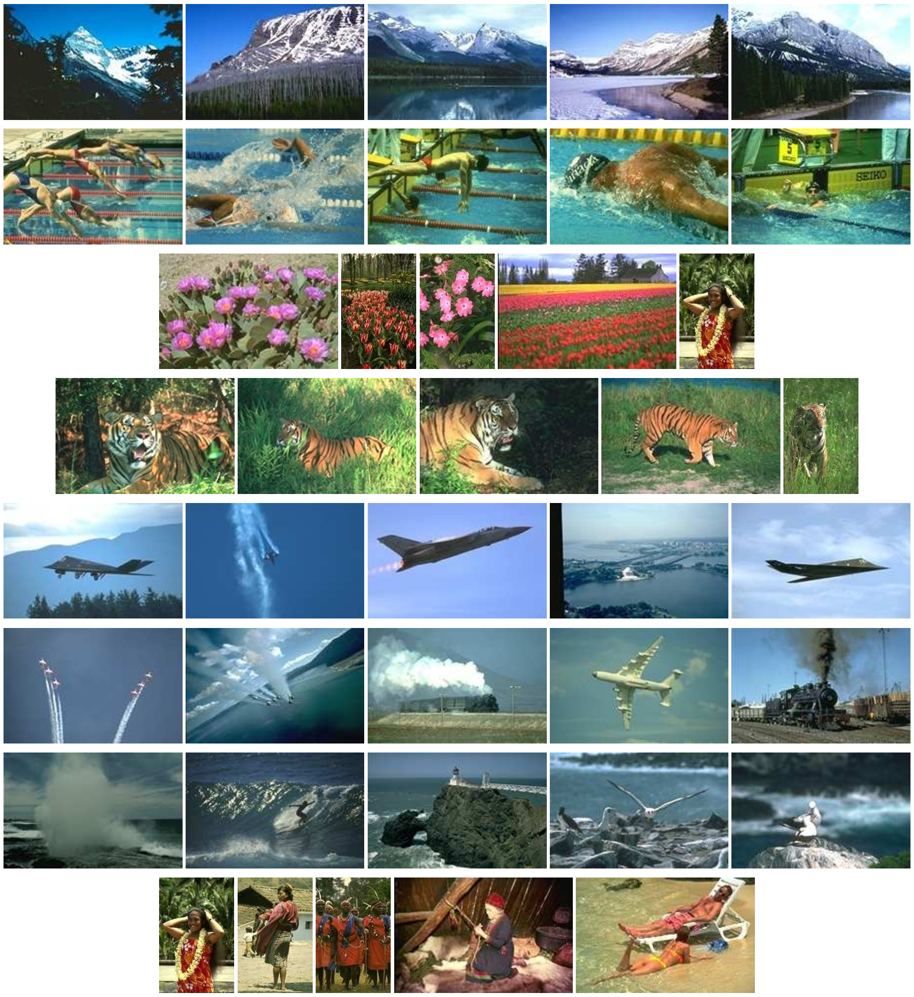


Figure 4: From top to bottom: first five ranked results for the query “mountain” (top), “pool” (row 2), “blooms” (row 3), “tiger” (row 4), “jet” (row 5), “smoke” (row 6), “waves” (row 7), and “woman” (row 8).

References

[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.

[2] D. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the* 26^{th} *Intl. ACM SIGIR Conf.*, pages 127–134, 2003.

[3] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum-likelihood from Incomplete Data via the EM Algorithm. *J. of the Royal Statistical Society*, B-39, 1977.

[5] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, volume 2, pages 97–112, 2002.

[6] D. Forsyth and M. Fleck. Body Plans. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico*, pages 678–683, 1997.

[7] P. Carbonetto H. Kueck and N. Freitas. A Constrained Semi-Supervised Learning Approach to Data Association. In *European Conference on Computer Vision*, Prague, Czech Republic, 2004.

[8] N. Haering, Z. Myles, and N. Lobo. Locating Dedicuous Trees. In *Workshop in Content-based Access to Image and Video Libraries*, pages 18–25, 1997, San Juan, Puerto Rico.

[9] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.

[10] J. Li and J.Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 25(10), 2003.

[11] Y. Li and L. Shapiro. Consistent line clusters for building recognition in CBIR. In *International Conference on Pattern Recognition*, volume 3, pages 952–956, Quebec, Canada, 2002.

[12] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[13] R. Picard. Digital Libraries: Meeting Place for High-Level and Low-Level Vision. In *Proc. Asian Conf. on Computer Vision*, December 1995, Singapore, USA.

[14] S.L.Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *IEEE CVPR*, 2004.

[15] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval: the end of the early years. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[16] Martin Szummer and Rosalind Picard. Indoor-Outdoor Image Classification. In *Workshop in Content-based Access to Image and Video Databases*, 1998, Bombay, India.

[17] A. Vailaya, A. Jain, and H. Zhang. On Image Classification: City vs. Landscape. *Pattern Recognition*, 31:1921–1936, December 1998.

[18] N. Vasconcelos. Image Indexing with Mixture Hierarchies. In *Proc. IEEE Computer Vision and Pattern Recognition Conf.*, Kawai, Hawaii, 2001.

SVCL-TR 2004/03

December 2004

Formulating Semantic Image Annotation as a Supervised Learning Problem

Gustavo Carneiro