A Database Centric View of Semantic Image Annotation and Retrieval

Gustavo Carneiro[†,*]
Department of Computer Science[†]
University of British Columbia
Vancouver, BC, Canada
carneiro@cs.ubc.ca

Nuno Vasconcelos[*]
Department of Electrical and Computer Engineering[*]
University of California, San Diego
San Diego, CA, USA
nuno@ece.ucsd.edu

ABSTRACT

We introduce a new model for semantic annotation and retrieval from image databases. The new model is based on a probabilistic formulation that poses annotation and retrieval as classification problems, and produces solutions that are optimal in the minimum probability of error sense. It is also database centric, by establishing a one-to-one mapping between semantic classes and the groups of database images that share the associated semantic labels. In this work we show that, under the database centric probabilistic model, optimal annotation and retrieval can be implemented with algorithms that are conceptually simple, computationally efficient, and do not require prior semantic segmentation of training images. Due to its simplicity, the annotation and retrieval architecture is also amenable to sophisticated parameter tuning, a property that is exploited to investigate the role of feature selection in the design of optimal annotation and retrieval systems. Finally, we demonstrate the benefits of simply establishing a one-to-one mapping between keywords and the states of the semantic classification problem over the more complex, and currently popular, joint modeling of keyword and visual feature distributions. The database centric probabilistic retrieval model is compared to existing semantic labeling and retrieval methods, and shown to achieve higher accuracy than the previously best published results, at a fraction of their computational cost.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models; I.4.8 [**Scene Analysis**]: Object Recognition; G.3 [**Probability and Statistics**]: Probabilistic Algorithms

General Terms

Algorithms,Measurement,Experimentation

Keywords

Image retrieval, automatic image annotation, supervised learning

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

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 research in the recent past [20]. 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 [17]. These are queries specified through a natural language description of the visual concepts of interest, and require the textual annotation of the images in the database. Since manual image labeling is a costly operation, there is currently great interest in the automatic extraction of semantic descriptors from images. Both semantic retrieval and automatic image annotation are instances of the more general problem of learning a mapping between images and keywords. In this work we show that such a mapping can be learned effectively and efficiently under a *database centric* view of the information retrieval (IR) problem, denoted by *probabilistic retrieval*.

To motivate the advantages of database centric retrieval, it is helpful to consider the limitations of the traditional, *query centric*, view of IR. Under this view, the goal of a retrieval system is to rank database documents according to their relevance/irrelevance to the query. This poses the retrieval problem as one of binary classification (a *relevant* vs an *irrelevant* class). While binary classifiers are a well understood area of machine learning, the retrieval problem introduces a twist that is rarely considered in the learning literature: because the relevance of a document with respect to a query cannot be defined until the latter is known, the relevant/irrelevant classifier can only be designed once the query is available. This introduces two notable constraints on the design of classifiers for retrieval. The first, which derives from the need to answer the query quickly, is a tight upper bound on training complexity. The second, due to the fact that typical queries only contain one example (or a few) from the relevant class, and none from the irrelevant, is scarcity of training data. These constraints are quite difficult to reconcile with the learning requirements, in terms of both training time and training set size, of most state-of-the-art classifiers.

The alternative proposed by the database centric formulation is to ground the classes on the database, rather than on the query. In the simplest case, this is done by defining each entry in the database (e.g. document or image) as an independent class. The retrieval problem is then defined as that of *finding the database class to which the query belongs*. The main advantage is that the classifier is defined by the database, not the query, and can therefore be *learned off-line*. This allows time to train more powerful classifiers, based on sophisticated data representations, e.g., mixtures [23] or hidden Markov models [19]. Furthermore, because classifier train-

ing is based on the *entire database*, rather than just the examples provided in the query, the resulting parameter estimates are significantly more reliable than what is feasible under the query centric formulation. Overall, by eliminating the two major bottlenecks of query centric retrieval, the database centric formulation can lead to significantly better retrieval performance.

Since database centric retrieval can be seen as a nearest neighbor search, under a suitable distance measure (the probability of the query given the database class), it is subjacent to the matching-based retrieval systems that have been popular since the early days of image retrieval [22, 15, 16, 8, 9, 11, 21, 12]. The precise probabilistic formulation was eventually formalized in [4, 26] and appears to have been rediscovered by the IR community at large, through the language modeling work of Ponte and Croft [18], a few years later. While the inherent benefits of longer training times and better model estimates are now fairly well understood, it has one additional advantage (over query centric retrieval) that does not appear to be widely appreciated. To identify this advantage, it is necessary to analyze the retrieval process in light of the causal relationships (from class to observations) that follow from the generative interpretation of any classification problem. While the query centric formulation poses the query as a source, which produces binary observations (database entries) in the relevant and irrelevant classes, the roles are reversed by the database centric formulation, where the query becomes an observation drawn from one of the many classes in the (source) database. This distinction is important when query and database concepts have different "granularity".

This is the case of semantic retrieval, where the query is an image but the database classes are generic semantic concepts such as "sky" or "grass". Due to this unbalance, answering the (query centric) question of whether the database concepts are relevant requires an ability to generalize that retrieval systems rarely possess. For example, when faced with a query consisting of a patch of bright blue sky, query centered retrieval systems need to somehow infer that sky could also be orange (and consequently infer that all images of sunsets are relevant). This problem is easily eliminated under the database centric formulation, by simply *defining the database classes as the semantic concepts of interest*. In terms of implementation, the only difference (with respect to non-semantic retrieval) is that one probability distribution is estimated per concept (using all the images that contain the concept) rather than per image. However, when compared to query centric retrieval, this makes for a substantial difference at retrieval time: while query centric retrieval requires a relevance judgment for *all types of images* in the relevant class from *a single example*, database centric retrieval only requires a similarity judgment for *one image* (the query) from the *probability distribution of the entire class*. From a generalization point of view, the former is an *extrapolation* problem, while the later only requires *interpolation*. In this sense, database centric retrieval is a significantly easier problem.

In this work, we show that the database centric probabilistic retrieval model has various interesting properties for both automatic image annotation and semantic retrieval. In particular it is shown that, under this model, optimal (in a minimum probability of error sense) annotation and retrieval can be implemented with algorithms that are conceptually simple, computationally efficient, and do not require prior semantic segmentation of training images. Due to its simplicity, the annotation and retrieval architecture is also amenable to sophisticated parameter tuning, a property that is exploited to investigate the role of feature selection in the design of optimal annotation and retrieval systems. Finally, we demonstrate the benefits of simply establishing a one-to-one mapping between keywords and the states of the semantic classification problem over the more complex, and currently popular, joint modeling of keyword and visual feature distributions. The database centric probabilistic retrieval model is compared to existing semantic labeling and retrieval methods, and shown to achieve higher accuracy than the previously best published results, at a fraction of their computational cost.

2. SEMANTIC LABELING AND RETRIEVAL

Consider a database $\mathcal{T} = \{\mathcal{I}_1, \ldots, \mathcal{I}_N\}$ of images $\mathcal{I}_i$ and a semantic vocabulary $\mathcal{L} = \{w_1, \ldots, w_L\}$ of semantic labels, or keywords, w_i. The goal of semantic image annotation is to, given an image $\mathcal{I}$, extract the set of keywords, or caption, $\mathbf{w}$ that best describes $\mathcal{I}$. The goal of semantic retrieval is to, given a keyword w, extract the images in the database that contain the associated visual concept. In both cases, 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.

Under database centric probabilistic retrieval, both labeling and retrieval are formulated as classification problems. Mathematically, this requires introducing 1) a random vector $\mathbf{X}$ of visual features, 2) 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 3) the set of corresponding class-conditional densities $P_{\mathbf{X}|W}(\mathbf{x}|i)$, $i \in \{1, \ldots, L\}$ for the distribution of visual features given the semantic class. Using well known results in statistical decision theory [5], it is not difficult to show that both labeling and retrieval can be implemented with minimum probability of error if the posterior probabilities

$$P_{W|\mathbf{X}}(i|\mathbf{x}) = \frac{P_{\mathbf{X}|W}(\mathbf{x}|i)P_W(i)}{P_{\mathbf{X}}(\mathbf{x})} \quad (1)$$

are available, where $P_W(i)$ is a prior probability for the i^{th} semantic class. For annotation, the minimum probability of error rule is to, given a set of query feature vectors $\mathbf{x}$, pick concept

$$i^*(\mathbf{x}) = \arg\max_i P_{W|\mathbf{X}}(i|\mathbf{x}) = \arg\max_i P_{\mathbf{X}|W}(\mathbf{x}|i)P_W(i). \quad (2)$$

For semantic retrieval, given concept w_i, the optimal rule is to select the database image of index

$$j^*(w_i) = \arg\max_j P_{W|\mathbf{X}}(i|\mathbf{x}_j) = \frac{P_{\mathbf{X}|W}(\mathbf{x}_j|i)P_W(i)}{P_{\mathbf{X}}(\mathbf{x}_j)} \quad (3)$$

where $\mathbf{x}_j$ is the set of feature vectors extracted from the j^{th} database image $\mathcal{I}_j$. In both cases, the ordering by decreasing posterior probability is a minimum probability of error ranking for the remaining keywords or images.

3. ESTIMATION OF CLASS DENSITIES

Given the collection of semantic class densities $P_{\mathbf{X}|W}(\mathbf{x}_j|i), \forall i, j$, both annotation and retrieval are relatively trivial operations. They simply consist of the search for the solution of (2) and (3), respectively, where $P_W(i)$ can be estimated by the relative frequencies of the various classes in the database and $P_{\mathbf{X}}(\mathbf{x}) = \sum_i P_{\mathbf{X}|W}(\mathbf{x}|i)$ $P_W(i)$. However, the estimation of the class densities raises two interesting questions. The first is computational complexity: if the database is large, the direct estimation of $P_{\mathbf{X}|W}(\mathbf{x}|i)$ from the set of all feature vectors extracted from all images that contain the concept w_i is usually infeasible. One solution is to discard part of the data, but this is suboptimal in the sense that important training cases may be lost. Section 3.1 discusses more effective alternatives. The second is whether it is possible to learn the densities of semantic concepts in the absence of a semantic segmentation for each image in the database. This is the subject of Section 3.2.

3.1 Density Estimation

One possibility to reduce the complexity of estimating $P_{\mathbf{X}|W}(\mathbf{x}|i)$, which we denote by *model averaging*, is to decompose the estimation in two steps. First, a density estimate is produced for each im-

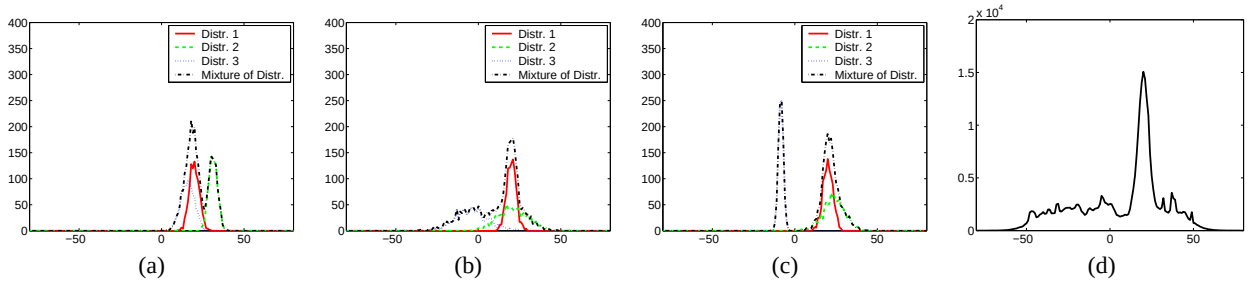


Figure 1: Illustrative example of the process of learning semantic class densities. Graphs (a)-(c) show three examples of the feature distributions of individual images. Graph (d) presents the feature distribution over 1,000 images. Although not necessarily dominant in any image, the concept density (shown in red) dominates over the entire training set.

age, originating a sequence $P_{\mathbf{X}|L,W}(\mathbf{x}|l,i), l \in \{1,\ldots D\}$ where L is a latent variable that indicates the image number. The class density is then obtained by averaging the densities in this sequence

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D}\sum_{l} P_{\mathbf{X}|L,W}(\mathbf{x}|l,i). \quad (4)$$

While the overall amount of data to be processed remains constant, each density estimate only involves the feature vectors extracted from one image. This allows the estimation to be performed in memory and results in significantly faster estimates[1]. Furthermore, images which contribute to various semantic concepts no longer need to be reprocessed for each of them. The fact that this is the case for most images adds another significant layer of computational savings.

While training complexity is substantially decreased, model averaging leads to class-conditional distributions that are too expensive to evaluate. Consider, for example, the case where each estimate is a Gaussian mixture

$$P_{\mathbf{X}|L,W}(\mathbf{x}|l,i) = \sum_{k} \pi_{i,l}^{k}\mathcal{G}(\mathbf{x},\mu_{i,l}^{k},\Sigma_{i,l}^{k}), \quad (5)$$

where $\sum_k \pi_{i,l}^k = 1$ and $\mathcal{G}(\mathbf{x},\mu,\Sigma)$ is a Gaussian of mean μ and covariance Σ. Direct application of (4) leads to

$$P_{\mathbf{X}|W}(\mathbf{x}|i) = \frac{1}{D}\sum_{k,l} \pi_{i,l}^{k}\mathcal{G}(\mathbf{x},\mu_{i,l}^{k},\Sigma_{i,l}^{k}) \quad (6)$$

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, straightforward model averaging can lead to an extremely slow annotation process. One efficient alternative is to adopt the hierarchical density estimation method proposed in [25] for image indexing. This method is based on a mixture hierarchy where children densities consist of different combinations of subsets of their parents components. A formal definition is given in [25], we omit the details for brevity. The important point is that, under this model, it is possible to estimate the parameters of the class mixture directly from those of the mixture resulting from model averaging. Assuming that this has DK components of parameters

$$\{\pi_j^k, \mu_j^k, \Sigma_j^k\}, j = 1,\ldots,D,\ k = 1,\ldots,K. \quad (7)$$

the estimation can be done with an extension of the expectation-maximization (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 these components, the algorithm iterates between the following steps.

E-step: compute

$$h_{jk}^t = \frac{\left[\mathcal{G}(\mu_j^k,\mu_c^t,\mathbf{\Sigma}_c^t)e^{-\frac{1}{2}trace\{(\mathbf{\Sigma}_c^t)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N}\pi_c^t}{\sum_l\left[\mathcal{G}(\mu_j^k,\mu_c^l,\mathbf{\Sigma}_c^l)e^{-\frac{1}{2}trace\{(\mathbf{\Sigma}_c^l)^{-1}\mathbf{\Sigma}_j^k\}}\right]^{\pi_j^k N}\pi_c^l}, \quad (8)$$

where N is a user-defined parameter (see [25] for details).

M-step: set

$$(\pi_c^t)^{new} = \frac{\sum_{jk} h_{jk}^t}{DK} \quad (9)$$

$$(\mu_c^t)^{new} = \sum_{jk} w_{jk}^t \mu_j^k, \text{ where } w_{jk}^t = \frac{h_{jk}^t \pi_j^k}{\sum_{jk} h_{jk}^t \pi_j^k} \quad (10)$$

$$(\mathbf{\Sigma}_c^t)^{new} = \sum_{jk} w_{jk}^t \left[\mathbf{\Sigma}_j^k + (\mu_j^k - \mu_c^t)(\mu_j^k - \mu_c^t)^T\right]. \quad (11)$$

Note 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, i.e., only marginally superior to that of model 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 model 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 (11) 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 by direct learning.

3.2 Modeling classes without segmentation

Many of the concepts of interest for semantic annotation or retrieval only occupy a fraction of the images that contain them. While objects, e.g. "bear" or "flag", are prominent examples of these concepts this property also holds for more generic semantic classes, e.g. "sky" or "grass". Hence, most images are a combination of various concepts and, ideally, the assembly of a training set for each semantic class should be preceded by 1) careful semantic segmentation, and 2) identification of the image regions containing the associated visual feature vectors. In practice, the manual

[1] Because hierarchical mixture density estimation is not the end-goal of this work, the characterization of complexity is rather informal. A precise characterization is available in [25]. We would, however, like to emphasize that the gains are substantial. For example, the experiments described in Section 6 would simply not have been feasible without hierarchical estimates.

segmentation of all database images with respect to all concepts of interest is infeasible. A pressing question is then whether it is possible to estimate the densities of a semantic class without prior semantic segmentation, i.e. from a training set containing a significant percentage of feature vectors from other semantic classes.

This question has been studied in the machine learning literature, where the it is usually referred to as *multiple instance* learning [13]. While the problem is still not completely understood, there is strong empirical evidence that, if enough images containing the concept of interest are available, the best fit to the density of its training set is a good approximation to the concept density. The basic idea is that, while all images will have probability mass on the region of the feature space associated with the concept, the remaining probability mass (due to the appearance of other concepts in the images) is uniformly spread out throughout the space (because the appearance of the remaining concepts is random). Since it has to integrate to one, this uniform component tends to have small amplitude (in particular when the feature space is high dimensional). Hence, while the density of the concept may not be dominant in any individual image, the consistent appearance makes it dominant over the entire training set. This is illustrated in Figure 1 which presents a simulation of this effect, when all classes are Gaussian of mean $\mu \in [-100, 100]$ and variance $\sigma \in [0.1, 10]$ and the ensemble contains of $1,000$ training images with three semantic concepts (the concept of interest, with $\mu = 20$ and $\sigma = 3$, and two others selected at random).

An example on a real image database is provided by Figure 2 which illustrates the quality of the semantic density estimates indirectly, by presenting their performance in a semantic segmentation task. In this example, each training image was broken into 8×8 pixel neighborhoods, and a feature vector extracted from each neighborhood. All densities were modeled as Gaussian mixtures, and the semantic densities were learned over a set of training images derived from the Corel data set (see Section 6 for a detailed discussion of this dataset and the features used). The same feature extraction procedure was then applied to a set of test images, and each feature vector classified into one of the semantic classes present in the image (the semantic classes were obtained from the caption provided with the image). Figure 2 depicts the class indexes that produced the largest posterior probability at each image location, illustrating how each pixel is assigned to each of the classes (class indexes are represented in the color bar on the right image). The class at each image location $\mathbf{x}$ was determined by

$$i^*(\mathbf{x}) = \begin{cases} \arg\max_i P_{W|\mathbf{X}}(i|\mathbf{x}), & \text{if} P_{W|\mathbf{X}}(i|\mathbf{x}) > \tau \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

where $\tau = 0.5$, $P_{W|\mathbf{X}}(i|\mathbf{x})$ was computed as in (1) with

$$P_{\mathbf{X}}(\mathbf{x}) = P_{\mathbf{X}|W}(\mathbf{x}|i)P_W(i) + P_{\mathbf{X}|W}(\mathbf{x}|\neg i)P_W(\neg i),$$

and the training set for "no class i" consisted of all training images that did not contain the class i in their set of semantic labels. In order to facilitate the visualization, the posterior maps shown on the right were obtained by adding a constant, the index of the class associated with the largest posterior, to that posterior. Regions where all posteriors were below threshold are declared "undecided". Finally, the segmentation map was blurred with a Gaussian filter. Note that, overall, this procedure results in very reasonable segmentations, indicating that the density estimates are very reasonable approximations to the true concept densities.

4. MODEL TUNING

One of the important properties of the database centric probabilistic retrieval formulation is that, due to the simplicity of the retrieval model, it enables the implementation of sophisticated parameter optimization procedures. For example, given the interpre-

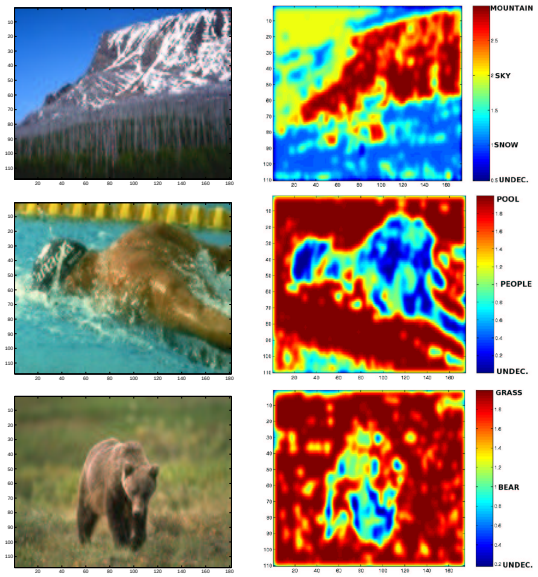


Figure 2: Original images (left column) and their maximum posteriors (right column) at each image neighborhood (Undec. means that no class has a posterior bigger that τ in (12)).

tation of semantic annotation and retrieval as classification problems, it is well known that the selection of good image features is an important requirement for accurate performance. Given a feature space $\mathcal{Z}$, the goal of feature selection is to find the best subset $\mathcal{X} \subseteq \mathcal{Z}$ of features, so as to enable accurate classification with reduced model complexity. Optimal feature selection is usually a problem of significant complexity, which depends on both the size of the training set (image database) and the complexity of the adopted probabilistic models. Since, for large databases, most models are too complex to enable any sophisticated feature selection, there is usually a dilemma of choosing between 1) a more sophisticated model with poor features, or 2) a simpler model with optimal features. The later option often provides better generalization guarantees and ends up achieving the best performance [24].

To illustrate how optimal feature selection can be easily accomplished under the database centric probabilistic retrieval model, we augment the basic architecture discussed so far with the feature selection algorithm proposed in [27]. Given a set of complexity constraints, this algorithm explores known statistical properties of images to find a set of features that is optimal in a discriminant sense, the maximization of mutual information between features and class label, closely related to the minimization of Bayes error rate. While, like most feature selection procedures, it is a greedy algorithm, it is unique in the sense that it sequentially chooses features so as to optimally balance three conflicting goals: 1) that the features must be discriminative, 2) that the features must not be redundant, and 3) that redundancy is acceptable if it is the source of information about the class label. The two latter can be seen as complexity penalties, and enable the computation of the optimal solution without compromise of scalability, making the algorithm viable in the semantic learning context. For brevity, we omit the implementation details, they are available in [27].

5. KEYWORD DISTRIBUTIONS

While the solution of the semantic annotation and retrieval problems with recourse to database centric probabilistic retrieval is a novel contribution of this work, there have been previous attempts to solve these problems through probabilistic modeling (e.g., [1, 2, 3, 6, 7, 10]). One aspect in which the solution now proposed is

Table 1: Performance comparison of automatic annotation on the Corel dataset.

Models	Co-occurrence	Translation	CRM	CRM-rect	MBRM	Mix-Hier
#words with recall > 0	19	49	107	119	122	137
Results on all 260 words						
Mean Per-word Recall	0.02	0.04	0.19	0.23	0.25	0.29
Mean Per-word Precision	0.03	0.06	0.16	0.22	0.24	0.23

fundamentally different from these efforts is the importance given to word distributions: while the previous approaches aim to create joint models for words and visual features (some even aim to provide a *translation* between the two modalities [6]), database centric probabilistic retrieval aims for the much simpler goal of estimating the visual feature distributions associated with each word. This implies that there is no need to introduce very sophisticated word probability models: word probabilities only influence the classification through the class prior $P_W(i)$.

Although this may appear as an over-simplification, we contend that it is more effective than estimating joint models, for three fundamental reasons. First, joint modeling can be quite complex, since the relationship between language and vision are highly nuanced and dependent on context. Second, the two modalities have fundamentally different representations (words are samples from discrete sources of finite alphabet, visual features are samples from continuous sources with open-ended vocabulary) and statistical learning and inference tend to be difficult when that is the case. Finally, the amount of training data is highly unbalanced: while each image may contribute thousands of feature vectors to the estimation of the visual component of the model, it contributes a very small number of observations to the text component. In result of all this, joint models for language and vision tend to be unrealistic simplifications of the underlying stochastic process, and the parameter estimates of the text component can be highly unreliable (a significant number of semantic concepts only appear a few times in the entire database [6]).

To illustrate these problems, we consider the semantic annotation and retrieval model that, to the best of our knowledge, has achieved the best existing results in experimental trials [7, 10]. This model introduces a latent variable L that indexes the image in the database, and assumes conditional independence between image features $\mathbf{X}$ and captions $\mathbf{T}$, i.e.

$$P_{\mathbf{X},\mathbf{T}}(\mathbf{x},\mathbf{t}) = \sum_{l=1}^{D} P_{\mathbf{X}|L}(\mathbf{x}|l) P_{\mathbf{T}|L}(\mathbf{t}|l) P_L(l) \qquad (13)$$

where D is the database size. This enables individual estimation of $P_{\mathbf{X}|L}(\mathbf{x}|l)$ and $P_{\mathbf{T}|L}(\mathbf{t}|l)$, and the overall density estimates are obtained by model averaging. The likelihood of the text component can be seen as weighting the contribution of each image to the overall estimate of the distribution of visual features. The training of the $P_{\mathbf{T}|L}(\mathbf{T}|l)$, $l \in \{1,\ldots,D\}$ is a maximum likelihood estimation based on the annotations associated with the l^{th} training image, and usually reduces to counting [7, 10]. At annotation time, the possible captions for the query $\mathcal{I}$ are ranked by either the joint probability of (13) or the posterior probability

$$P_{\mathbf{T}|\mathbf{X}}(\mathbf{t}|\mathbf{x}) = \frac{P_{\mathbf{X},\mathbf{T}}(\mathbf{x},\mathbf{t})}{P_{\mathbf{X}}(\mathbf{x})}. \qquad (14)$$

While the latter can be interpreted as the Bayesian decision rule for a classification problem with the states of $\mathbf{T}$ as classes, such class structure is not consistent with the generative model of (13) which enforces a causal relationship from L to $\mathbf{T}$. This leads to a very weak dependency between the observation $\mathbf{X}$ and class $\mathbf{T}$ variables, e.g., that they are independent given L. Hence, there is a mismatch between the class structure used for designing the probabilistic models (where the states of the latent variable are the classes) and that used for labeling and retrieval (which assume the states of $\mathbf{T}$ to be the classes). This can lead to decisions that are suboptimal in a minimum probability of error sense.

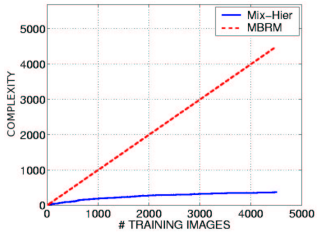


Figure 3: Comparison of the time complexity for the annotation of a test image on the Corel data set.

It is important to note that database centric probabilistic retrieval does not preclude the use of a joint distribution for visual features and text. For example, in (2) and (3), $P_{\mathbf{X}|W}(\mathbf{x}|i)$ can be replaced by

$$\begin{aligned} P_{\mathbf{X},\mathbf{T}|W}(\mathbf{x},\mathbf{t}|i) &= \frac{1}{D_i}\sum_{l=1}^{D_i} P_{\mathbf{X},\mathbf{T}|L,W}(\mathbf{x},\mathbf{t}|l,i) \\ &= \frac{1}{D_i}\sum_{l=1}^{D_i} P_{\mathbf{X}|L,W}(\mathbf{x}|l,i) P_{\mathbf{T}|L,W}(\mathbf{t}|l,i) \end{aligned} \qquad (15)$$

where L is a latent variable that indexes the images containing concept w_i, and we have assumed conditional independence between text and visual features given the semantic class. This model is equivalent to (13) but with the latent variable L restricted to the images of the i^{th} semantic class, enabling consistency with the minimum probability of error goal for annotation and retrieval. Although the probability mass of the text component is highly concentrated on the word associated with semantic class i (for the same arguments as in section 3.2) this component will also capture the co-occurrences with other words. It resembles the translation model [6], with the clusters in the feature space of [6] replaced by the hierarchical estimates of the class densities $P_{\mathbf{X}|W}(\mathbf{x}|i)$ discussed in section 3.1. The text component $P_{\mathbf{T}|W}(\mathbf{t}|i)$ is modeled by a multinomial distribution, as in [10]. The comparison of the performance achieved with this model and the text-free model previously discussed provides insight on the benefits of text modeling. These are discussed in the following section.

6. EXPERIMENTAL RESULTS

To evaluate the performance of semantic annotation and retrieval we relied on the Corel data set used in [6, 10, 7]. The translation model of [6] was the first milestone in the area of semantic annotation, in the sense of demonstrating results of practical interest. After years of research, and various other contributions, the best existing results are, to the best of our knowledge, those of[7]. We

therefore adopt an evaluation strategy identical to that used in this work. In particular, all experiments discussed below are based on the database introduced in [6] [2], which consists of $5,000$ images from 50 Corel Stock Photo CDs, divided into three parts: a training set of $4,000$ images, a validation set of 500 images, and a test set of 500 images. After model parameters are optimized using the validation set, this is merged with the training set to build a new training set of $4,500$ images. Each image has a caption of 1-5 keywords, and there are 371 keywords in the data set. With respect to the visual component, the YBR color space was adopted, each image decomposed into a set of overlapping 8×8 windows, the discrete cosine transform (DCT) applied to each window, and the image represented as a bag of feature vectors containing the first 21 DCT coefficients of each color channel. Note that this feature set is different from the one used in [6, 10, 7] (which consists of color, texture, and shape features).

6.1 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 goal is to automatically generate a caption which is then compared to the annotation produced by a human. Similarly to [10, 7] we define the automatic annotation as the five semantic classes of largest posterior probability. We then compute the recall and precision of every word in the test set. For a semantic descriptor w, assuming that there are $|w_H|$ human annotated images in the test set, and the system annotates $|w_{\text{auto}}|$, of which $|w_C|$ are correct, recall and precision are given by $recall = \frac{|w_C|}{|w_H|}$, $precision = \frac{|w_C|}{|w_{\text{auto}}|}$. As suggested by [10, 7], the values of recall and precision are averaged over the set of 260 words that appear in the test set. Table 1 presents these results for both the approach now proposed (which is denoted by 'Mix-Hier') and various other previously proposed methods (results borrowed from [10, 7]). Specifically, we considered: the co-occurrence model [14], the translation model [6], the continuous-space relevance model (CRM-rect)[10, 7], and the multiple-Bernoulli relevance model (MBRM) [7]. Note that the Mix-Hier results assume a uniform distribution $P_W(i)$ of semantic keywords in (2).

Overall, the method now proposed achieves the best performance. When compared to the previous best results (MBRM) it exhibits a gain of 16% in recall for an equivalent level of precision. Similarly, the number of words with positive recall increases by 15%. Fig. 4 presents some examples of the annotations produced. Note that when the system annotates an image with a descriptor not contained in the human-made caption, this annotation is frequently plausible. 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 the number of visual feature vectors per image. 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 feature vectors R, Figure 3 shows how the annotation time of a test image grows for Mix-Hier and MBRM, as a function of the number of training images, on the Corel dataset.

6.2 Image Retrieval with Single Word Queries

To evaluate the performance of semantic retrieval, precision and recall were computed as follows: when the n top matches 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 [7], evaluating the retrieval performance by the mean average precision. As can be seen from Table 2, 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. Figure 5 illustrates the performance of the system on one word queries for challenging visual concepts. Note the diversity of visual appearance of the returned images, indicating that the method now proposed has good generalization ability.

Table 2: 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

Table 3: Performance comparison between Mix-Hier and Mix-Hier-SDK for the task of automatic annotation.

Models	Mix-Hier	Mix-Hier-SKD
#words with recall > 0	137	86
Results on all 260 words		
Mean Per-word Recall	0.29	0.17
Mean Per-word Precision	0.23	0.20

Table 4: Performance comparison between Mix-Hier and Mix-Hier-SDK for the task of image retrieval.

Mean Average Precision for Corel Dataset		
Models	All 260 words	Words with recall > 0
Mix-Hier	0.31	0.49
Mix-Hier-SKD	0.20	0.27

6.3 Semantic Keyword Distribution

In this section we evaluate the benefits of including semantic keyword distributions in the probabilistic model, i.e. using (15). Tables 3 and 4 show a comparison between this model (denoted by Mix-Hier-SKD) and the text-free Mix-Hier model in the tasks of image annotation and retrieval, respectively (the results of Mix-Hier are repeated to facilitate the comparison). Note that Mix-Hier produces significantly better results in both tasks. We believe that this is due to the factors discussed in Section 5: the difficulty of combining continuous and discrete variables and the unreliability of the estimates of keyword probabilities[3].

6.4 Feature Selection

In this section we briefly discuss the performance improvement of Mix-Hier resulting from the addition of the feature selection method of Section 4. Figure 6 shows annotation results (number of words with recall > 0, mean precision, and mean recall), while the retrieval results (mean precision-recall for all words and for words of recall > 0) are presented in Figure 7. In all plots the performance is shown as a function as the number of features selected (number of subspaces of the feature space where the classifier is defined). Note that, for both annotation and retrieval, the results achieved with the best 32 features are equivalent to those attained on the full 64-dimensional space, but have half complexity. While these results support the argument that feature selection is beneficial, the good performance of the complete feature set is somewhat

[2] We would like to thank Kobus Barnard to make this dataset available for our experiments.

[3] The results reported are the best achieved over a set of trials using different strategies for regularizing the keyword probabilities.

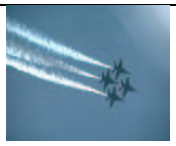
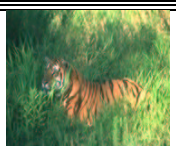
					
Human Annotation	sky jet plane smoke	snow fox arctic	sky buildings street cars	water bridge train railroad	water pool athlete swimmers
Mix-Hier Annotation	plane jet smoke flight prop	arctic snow polar fox ice	street buildings bridge sky arch	sky bridge locomotive water train	swimmers people water pool athlete
					
Human Annotation	grass forest cat tiger	bear polar snow tundra	coral fish ocean reefs	buildings clothes shops street	mountain sky clouds tree
Mix-Hier Annotation	cat tiger plants leaf grass	polar tundra bear snow ice	reefs coral ocean fan fish	buildings street shops people skyline	mountain valley sky clouds tree

Figure 4: . Comparisons of annotations made by our system and annotations made by a Human subject.

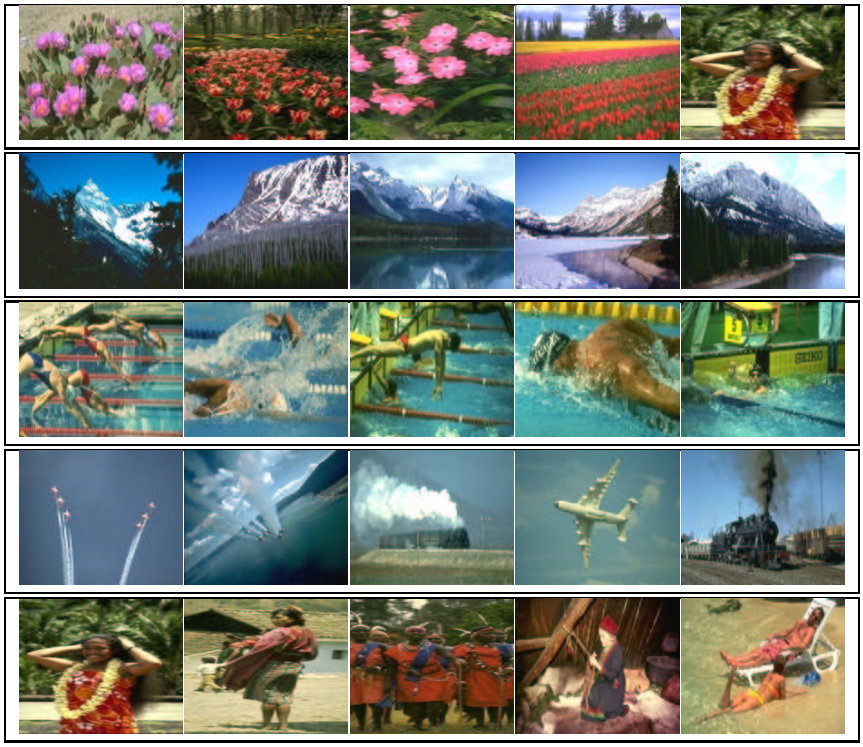


Figure 5: Semantic retrieval results on Corel. Each row shows the top five matches to a semantic query. From top to bottom: five top matches for *'blooms'*, *'mountain'*, *'pool'*, *'smoke'*, and *'woman'* .

surprising: our previous experience with non-semantic retrieval is that performance starts to degrade after 16 to 32 features. We believe that the increased robustness of semantic retrieval is due to the intrinsic data-driven regularization of hierarchical density estimation, as discussed in Section 3.1.

6.5 Generalization

We finish with an evaluation of the generalization ability of the semantic retrieval model. Figure 8 presents the curves of average precision-recall, and associated error bars, obtained over the entire test set, but grouped by the number of available training examples from the class of the query. Once again the results are somewhat surprising, since the performance seems to improve for classes with less training examples. This is likely to be due to the make-up of the Corel dataset, where classes with few examples tend to contain images with similar scenes. Nevertheless, these results suggest that performance starts to stabilize at about 100 examples: adding more examples decreases the precision-recall variance, but does not seem to affect its mean. This is an encouraging result, since it indicates that semantic retrieval is feasible with small training sets. On the other hand, it also indicates that the average curve of precision recall over all queries should be taken with a grain of salt.

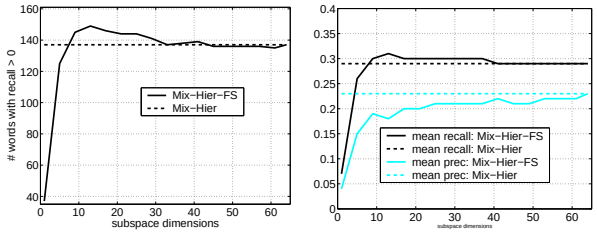


Figure 6: Automatic image annotation results using feature selection (Mix-Hier-FS). The straight dashed-line shows the Mix-Hier result.

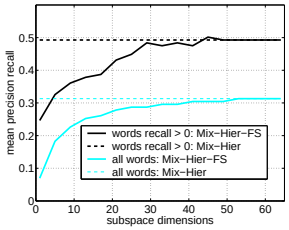


Figure 7: Image retrieval results using feature selection (Mix-Hier-FS). The straight dashed-line shows the Mix-Hier result.

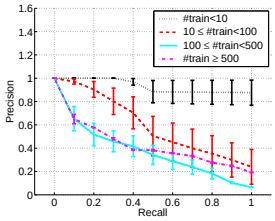


Figure 8: Average curves of precision-recall, and error bars, for semantic retrieval with classes of variable training set size.

More realistic values would likely be obtained by discarding the semantic classes with very few training examples. We have not done so to maintain consistency with the experimental set-up previously adopted in the literature.

7. REFERENCES

[1] K. Barnard and D. Forsyth. Learning the Semantics of Words and Pictures. In *Int. Conf. on Computer Vision*, 2001.

[2] D. Blei and M. Jordan. Modeling Annotated Data. In *Proc. ACM SIGIR*, 2003.

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

[4] I. Cox, M. Miller, S. Omohundro, and P. Yianilos. PicHunter: Bayesian Relevance Feedback for Image Retrieval. In *Int. Conf. on Pattern Recognition*, 1996.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* John Wiley and Sons, 2001.

[6] P. Duygulu, K. Barnard, and D. Forsyth N. Freitas. Object Recognition as Machine Translation: Learning a lexicon for a fixed image vocabulary. In *European Conf. on Computer Vision*, 2002.

[7] S. Feng, R. Manmatha, and V. Lavrenko. Multiple Bernoulli Relevance Models for Image and Video Annotation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[8] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[9] A. Jain and A. Vailaya. Image Retrieval using Color and Shape. *Pattern Recognition Journal*, 29:1233–1244, August 1996.

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

[11] W. Ma and H. Zhang. Benchmarking of Image Features for Content-based Retrieval. In 32^{nd} *Asilomar Conference on Signals, Systems, and Computers, Asilomar, California*, 1998.

[12] B. Manjunath and W. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 18(8):837–842, August 1996.

[13] O. Maron and T. Lozano-Perez. A framework for multiple instance learning. In *NIPS*, 1998.

[14] 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.

[15] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Pektovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content using color, texture, and shape. In *SPIE Storage and Retrieval for Image and Video Databases, San Jose, California*, pages 173–181, 1993.

[16] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. *Int. Journal of Computer Vision*, Vol. 18(3):233–254, June 1996.

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

[18] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proc. ACM SIGIR*, Melbourne, Australia, 1998.

[19] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[20] 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.

[21] J. Smith and S. Chang. VisualSEEk: a fully automated content-based image query system. In *ACM Multimedia, Boston, Massachussetts*, pages 87–98, 1996.

[22] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision*, Vol. 7(1):11–32, 1991.

[23] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions*. John Wiley, 1985.

[24] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

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

[26] N. Vasconcelos and A. Lippman. Library-based Coding: a Representation for Efficient Video Compression and Retrieval. In *Proc. Data Compression Conference, Snowbird, Utah*, 1997.

[27] N. Vasconcelos and M. Vasconcelos. Scalable Discriminant Feature Selection for Image Retrieval and Recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.