

Toward Bridging the Annotation-Retrieval Gap in Image Search by a Generative Modeling Approach

Ritendra Datta¹, Weina Ge¹, Jia Li^{2,1}, and James Z. Wang^{3,1}

¹ Computer Science and Engineering, ² Statistics, and ³ Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802, USA

{datta, wnge, jiali, jwang}@psu.edu

ABSTRACT

While automatic image annotation remains an actively pursued research topic, enhancement of image search through its use has not been extensively explored. We propose an annotation-driven image retrieval approach and argue that under a number of different scenarios, this is very effective for semantically meaningful image search. In particular, our system is demonstrated to effectively handle cases of partially tagged and completely untagged image databases, multiple keyword queries, and example based queries with or without tags, all in near-realtime. Because our approach utilizes extra knowledge from a training dataset, it outperforms state-of-the-art visual similarity based retrieval techniques. For this purpose, a novel structure-composition model constructed from Beta distributions is developed to capture the spatial relationship among segmented regions of images. This model combined with the Gaussian mixture model produces scalable categorization of generic images. The categorization results are found to surpass previously reported results in speed and accuracy. Our novel annotation framework utilizes the categorization results to select tags based on term frequency, term saliency, and a WordNet-based measure of congruity, to boost salient tags while penalizing potentially unrelated ones. A bag of words distance measure based on WordNet is used to compute semantic similarity. The effectiveness of our approach is shown through extensive experiments.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*Indexing Methods*; I.4.9 [Image Processing and Computer Vision]: Applications

General Terms

Algorithm, Design, Experimentation, Performance.

Keywords

Image Search, Generative Models, Automatic Annotation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'06, October 23–27, 2006, Santa Barbara, California, USA.
Copyright 2006 ACM 1-59593-447-2/06/0010 ...\$5.00.

1. INTRODUCTION

The volume of digital imagery, acquired directly through imaging devices or indirectly by digitization, is expanding rapidly. In this scenario, the ability to automatically annotate large volumes of images and make them available for semantically meaningful image retrieval can come in very handy. For example, our interactions with the museum community revealed that due to shortage of manpower, there is an immediate need for a system that could automatically annotate their large picture archives and provide a searchable interface internally and for public usage. To this end, a number of attempts have been made at automated image annotation [1, 3, 4, 8, 12, 13, 17, 22]. While many interesting ideas have emerged and promising results reported, the direct use of automated annotation for image search has received less attention. The usual assumption is that good annotation automatically leads to a good image search experience. Moreover, most proposed annotation systems either do not scale well or do not explicitly report annotation speed. These factors make their potential usefulness in real-world image search systems susceptible to doubt.

Ideally, if all images in a database were reliably tagged, keyword based querying would be synonymous with text search. The need for visual content based searching arises primarily because (1) large image databases are seldom fully tagged (e.g., the Yahoo! Flickr image database), and (2) the tags are often unreliable or inconsistent (e.g., surrounding text treated as tags, for Web images). In this work, we propose a scalable image search system built atop an automatic annotation framework, and demonstrate its effectiveness from a retrieval perspective. In particular, we consider three scenarios: (1) the database is partially tagged, and keyword-based or tagged image queries are made on the untagged portion, (2) the database is partially tagged, and example-based untagged image queries are made on the tagged portion, and (3) the database is untagged, and example-based image queries are made, given that a visually coherent tagged image database (i.e., a knowledge base) is available at one's disposal. To achieve fast, semantically meaningful retrieval of images under these scenarios, we propose a novel image categorization method using Beta distributions and the Gaussian mixture model, which forms the basis for automatic annotation. The annotation process involves a novel WordNet-based [23] tag selection strategy which implicitly performs generative model combination as well. Extensive image search experiments are performed and

compared with alternative retrieval strategies involving a state-of-the-art image retrieval system. We summarize our main contributions as follows:

- To the best of our knowledge, this is the first work demonstrating through extensive experiments how near-realtime annotation can actually help in image search. The use of annotation for retrieval under circumstances of untagged databases, multiple keyword queries, and untagged image-based queries is demonstrated and contrasted with alternative retrieval strategies in each case. Our method is found to significantly outperform competing strategies.
- A novel structure-composition (S-C) model based on Beta distributions is proposed for capturing the spatial structure and composition of generic image categories. This model is found to be useful in capturing visual composition of challenging picture categories. An efficient single-pass algorithm for extracting the S-C model features from images is presented.
- The S-C model, together with a Gaussian mixture model for capturing representative color and texture, is shown to produce near-realtime categorization with as many as 600 different categories, and outperform best reported image categorization results.
- A novel annotation strategy is proposed which takes into consideration the evidence provided by the categorization results for potential tags, the chance occurrence of such words, and the congruity of a word among the pool of candidate words, as evidenced by the WordNet ontology.

Our experiments are performed on 54,000 images from the commonly used [1, 3, 4, 8, 12, 13, 17, 22, 20, 21, 14, 19] Corel Stock Photo CDs, and 1,000 publicly annotated images obtained from Yahoo! Flickr.

1.1 Related Work

One way to group past attempts at automatic image annotation is by whether prior image categorization forms the basis for annotation. Some approaches [1, 8, 12, 13, 22, 20] deal directly with the problem of annotation, providing labels to either each region or the whole image. Others [4, 17] treat the problem of annotation in two independent stages, first categorizing the images and then associating labels to them using the top ranked categories. In the latter case, image categorization becomes a critical step and needs to be reliable in order to generate useful annotation. Generic image categorization has been attempted previously, using techniques including Bayes point machines [4], multi-resolution hidden Markov models [17], multiple instance learning [5], generative/discriminative modeling [18], random sub-windows [21], and Bayesian Modeling [7].

Semantically meaningful content-based image retrieval (CBIR) [24] incorporating user feedback have been explored [9, 10, 19]. Performance of automatic annotation given single word queries have been reported [3, 8]. Enhanced image retrieval given partially annotated image databases have been proposed [16]. In [27], the utilization of textual annotations for Web image retrieval has been reported. The use of WordNet for pruning irrelevant keywords from automatically annotated sets has been shown

effective [14]. While public domain systems for human annotation of pictures (e.g., *Yahoo! Flickr*) remain popular, we also witness developments from within the research community (e.g., the CMU *ESP Game*, the IBM *EVA* system [25]).

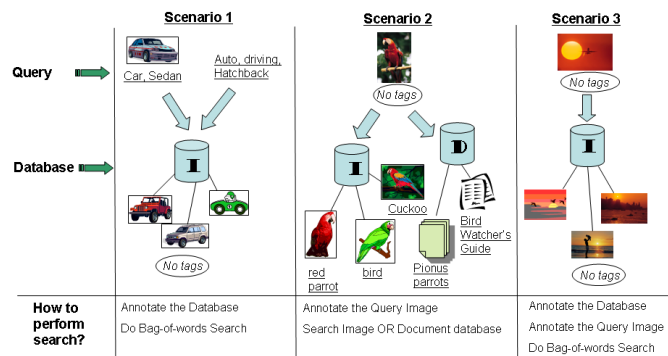


Figure 1: Three common scenarios for real-world image retrieval.

1.2 Bridging the Gap

Our motivation to ‘bridge’ the annotation-retrieval gap is driven by a desire to effectively handle challenging cases of image search in a unified manner. These cases are schematically presented in Figure 1, and elucidated below.

- **Scenario 1:** Either a tagged image or a set of keywords is used as query, the latter being a popular modality in public domain search engines such as Google Images and Yahoo! Images. Problem arises when part or whole of the image database (e.g., Web images) is not tagged, making this portion inaccessible through text queries. We study the effectiveness of our annotation algorithm in tagging the database and subsequently performing multiple keyword retrieval. Results are compared with those obtained by an intuitive CBIR-based retrieval strategy.
- **Scenario 2:** An untagged image is used as query, with the desire to find either semantically related images or documents from a database or the Web. We study the effectiveness of our annotation algorithm in tagging the query image and subsequently performing text-based retrieval. Results are compared with those obtained by a CBIR-based retrieval strategy.
- **Scenario 3:** The query image and part/whole of the image database are untagged. This is the case that best motivates CBIR, since the only available information is visual content. We study the effectiveness of our annotation algorithm in tagging the query image and the image database, and subsequently performing text-based retrieval. Results are compared with those obtained by a CBIR system.

Additional goals include the ability to generate precise annotations of pictures in real time. It is noted that most proposed annotation systems assess performance based on the quality of annotation alone. In our case, this is only part of the goal. Our prime challenge is to have the annotations help generate semantically meaningful and

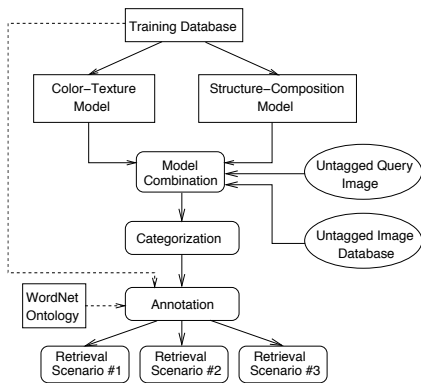


Figure 2: Our annotation-driven retrieval approach.

precise retrieval. To this end, we develop our approach as follows. We first aim to build a near-realtime categorization algorithm capable of producing accurate results. Here, the term *near-realtime* refers to a performance between one and ten seconds per image. We regard our annotation system as near-realtime because it is within this range while other annotation systems reported in literature are not. We then proceed to generate annotation on the basis of categorization, ensuring high precision and recall. With this annotation system in place, we assess its performance as a means of image retrieval under the scenarios described above. In each case, we contrast performance with alternative CBIR-based approaches, in the absence of supervised categorization/annotation but with access to a CBIR system. An overview of our approach is presented in Figure 2.

2. MODEL-BASED CATEGORIZATION

We employ a generative modeling based approach for accurate, near-realtime categorization of generic images. Generative model based categorization implies training independent statistical models for each image category given a small set of training images. Assignment of category labels to unseen images is then a process of interpreting the set of likelihoods of the image features being generated by each trained category model. In our system, two heterogeneous generative models (per image category) provide evidence for categorization from two different aspects of generic images. We generate final categorization of images by combining the evidence from these models. Heterogeneity in this context implies that the two models make independent categorization errors, making the evidence combination advantageous. This idea will be elaborated upon later.

Formally, let there be a feature extraction process or function \mathfrak{S} that takes in an image I and returns a collection of D feature vectors, each of dimension V , i.e., $\mathfrak{S}(I)$ has dimension $D \times V$, D varying with each image. Given C categories and N training images per category, each of C models $M_k, k = 1, \dots, C$ with parameters θ_k are built using training images $I_i^k, i = 1, \dots, N$, by some parameter estimation technique. Suppose the collection of feature vectors, when treated as random variables $\{X_1, \dots, X_D\}$, can be assumed conditionally independent given model parameters θ_k . For a test image I , given that $\mathfrak{S}(I) = \{x_1, \dots, x_D\}$ is extracted, the log-likelihood of I being

generated by model M_k is

$$\ell_1(I|M_k) = \log p(x_1, \dots, x_D|\theta_k) = \sum_{d=1}^D \log p(x_d|\theta_k). \quad (1)$$

Assuming equal category priors, a straightforward way to assign a category label y to I would be to have $y(I) = \arg \max_k \ell_1(I|M_k)$. Now consider that we have another set of C generative models trained on a different set of image features and with a different underlying statistical distribution. Suppose the log-likelihoods generated by these models for the same image I are $\{\ell_2(I|M_1), \dots, \ell_2(I|M_C)\}$. Each category of generic images is typically described by multiple tags (e.g., tiger, forest, and animal for a tiger category). Given a large number of categories, many of them having semantic/visual overlaps (e.g., night and sky, or people and parade), the top ranked category alone from either model may not be accurate. One way to utilize both models in the categorization process is to treat them as two experts independently examining the images from two different perspectives, and reporting their findings. The findings are not limited to the two most likely categories for each model, but rather the entire set of likelihoods for each category, given the image. Hence, an appropriate model combination strategy $\rho(\cdot)$ may be used to predict the image categories in a more general manner:

$$y(I) = \rho\left(\ell_1(I|M_1), \dots, \ell_1(I|M_C), \ell_2(I|M_1), \dots, \ell_2(I|M_C)\right). \quad (2)$$

For a large number of generic image categories, building a robust classifier is an uphill task. Feature extraction is extremely critical here, since it must have the discriminative power to distinguish between a broad range of image categories, no matter what machine learning technique is used. We base our models on the following intuitions: (1) For certain categories such as sky, marketplace, ocean, forests, Hawaii, or those with dominant background colors such as paintings, color and texture features may be sufficient to characterize them. In fact, a structure or composition for these categories may be too hard to generalize. (2) On the other hand, categories such as fruits, waterfall, mountains, lions, and birds may not have dominating color or texture but often have an overall structure or composition which helps us identify them despite heavily varying color distributions. In [17], the authors use 2-D multi-resolution hidden Markov models (2-D MHMMs) to capture the inter-scale and intra-scale dependence of block-based color and texture based features, thus characterizing the composition/structure of image categories. Problems with this approach are that the dependence modeling is over relatively local image regions, the parameter estimation algorithm involves numerical approximation, and the overall categorization process is slow. While our work is inspired by similar motivations, we aim at near-realtime and more accurate categorization. We thus build two models to capture different visual aspects, (1) a structure-composition model that uses Beta distributions to capture color interactions in a very flexible but principled manner, and (2) a Gaussian mixture model in the joint color-texture feature space. We now elaborate on each model.

2.1 Structure-Composition (S-C) Models

The idea of building such a feature arose from a desire to represent how the colors interact with each other in certain

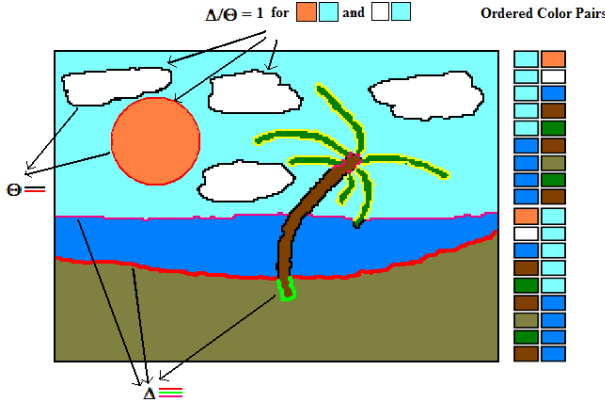


Figure 3: The idea behind the S-C model is illustrated. The perimeters of the segments are denoted Θ , while the border lengths between pairs of segments are denoted Δ . Intuitively, Δ/Θ ratios for the *orange*, *light-blue* (sun and sky) and *white*, *light-blue* (clouds and sky) pairs equal 1 since sun and cloud perimeters coincide with their borders shared with sky. The ratio is typically close to zero when segments are barely touching, and near 1 when a segment is completely contained within another. We model these ratios for all pairs of quantized colors in order to capture the structure or composition of image categories.

picture categories. The average beach picture could be described by a set of relationships between different colored regions, e.g., orange (sun) completely inside light-blue (sky), light-blue sharing a long border with dark-blue (ocean), dark-blue sharing a long border with brown (sand) etc. For tiger images, this description could be that of yellow and black regions sharing very similar borders with each other (stripes) and rest of the colors interacting without much pattern or motif. Very coarse texture patterns such as pictures of beads of different colors (not captured well by color distribution or localized texture features such as wavelets) could be described as any color (bead) surrounding any other color (bead), some color (background) completely containing most colors (beads), and so on. This idea led to a principled statistical formulation of rotational and scale invariant structure-composition (S-C) models.

Given the set of all training images across categories, we take every pixel from each image, converted to the perceptually uniform *LUV* color space. We thus have a very large population of *LUV* vectors in the \mathbb{R}^3 space representing the color distribution within the entire training set. The *K*-means geometric clustering with uniform initialization is performed on a manageable random sub-sample to obtain a set of *S* cluster centroids $\{T_1, \dots, T_S\}$, e.g., shades of red, yellow etc. We then perform a nearest-neighbor based segmentation on each training image *I* by assigning a cluster label to each pixel (x, y) as follows:

$$J(x, y) = \arg \min_i |I_{luv}(x, y) - T_i|. \quad (3)$$

In essence, we have quantized the color space for the

entire set of training images to obtain a small set of representative colors. This helps to build a uniform model representation for all image categories. To uniquely identify each segment in the image, we perform a two-pass 8-*connected component labeling* on *J*. The image *J* now has *P* connected components or segments $\{s_1, \dots, s_P\}$. The many-to-one mapping from a segment s_i to a color T_j is stored and denoted by the function $G(s_i)$. Let χ_i be the set of neighboring segments to segment s_i . Neighborhood in this sense implies that for two segments s_i and s_j , there is at least one pixel in each of s_i and s_j that is 8-connected. We wish to characterize the interaction of colors by modeling how each color shares (if at all) boundaries with every other color. For example, a *red-orange* interaction (in the quantized color space) for a given image category will be modeled by how the boundaries are shared between every *red* segment with every other *orange* segment for each training image, and vice-versa (Figure 3). More formally, let $(x, y) \in B$ indicate that pixel (x, y) in *J* is 8-connected to segment *B*, and let $\mathbb{N}(x, y)$ denote the set of its 8 neighboring points (not segments). Now we define a function $\Delta(s_i, s_j)$ which denotes the length of the shared border between a segment s_i and its neighboring segment s_j , and a function $\Theta(s_i)$ which defines the total length of the perimeter of segment s_i ,

$$\Delta(s_i, s_j) = \sum_{(x, y) \in s_i} In((x, y) \in s_j), s_j \in \chi_i, \text{ and} \quad (4)$$

$$\Theta(s_i) = \sum_{(x, y) \in s_i} In(\mathbb{N}(x, y) \not\subset s_i), \quad (5)$$

where $In(\cdot)$ is the indicator function. By this definition of \mathbb{N} , inner borders (e.g., holes in donut shapes) and image boundaries are considered part of segment perimeters. We want to model the Δ/Θ ratios for each color pair by some statistical distribution. For random variables bounded in the $[0, 1]$ range, the *Beta distribution* is a flexible continuous distribution defined in the same range, with *shape* parameters (α, β) . The Beta density function is defined as

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}, \text{ given} \quad (6)$$

$$B(\alpha, \beta) = \int_0^1 v^{\alpha-1} (1-v)^{\beta-1} dv = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \quad (7)$$

where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ is the well-known *Gamma function*. Our goal is to build models for each category such that they consist of a set of Beta distributions for every color pair. For each category, and for every color pair, we find each instance in the *N* training images in which segments of that color pair share a common border. Let the number of such instances be η . We then compute the corresponding set of Δ/Θ ratios and estimate a Beta distribution (i.e., parameters α and β) using these values for that color pair. The overall structure-composition model for a given category *k* thus has the following form:

k	1	2	...	S
1	n/a	α, β, η	...	α, β, η
2	α, β, η	n/a
...	α, β, η
S	α, β, η	...	α, β, η	n/a

Note that it is not possible to have segments with the same color as neighbors. Thus parameters of the form $\alpha(i, i), \beta(i, i)$ or $\eta(i, i)$ do not exist, i.e., same color pair entries in the model are ignored, denoted by ‘n/a’. Note also that the matrix is *not* symmetric, which means the color pairs are ordered, i.e., we treat yellow-orange and orange-yellow color interactions differentially, for example. Further, the number of samples η used to estimate the α and β are also stored with the corresponding entries as part of the model. The reason for doing so will be evident shortly.

For the estimation of α and β , a moment matching method is employed for its computational efficiency. Given a set of $\eta(i, j)$ Δ/Θ samples for a given color pair (i, j) , having values $\{x_1, \dots, x_{\eta(i, j)}\}$, the parameters are estimated as follows:

$$\alpha(i, j) = \bar{x} \left(\left(\frac{\bar{x}(1-\bar{x})}{s^2} \right) - 1 \right)$$

$$\beta(i, j) = (1 - \bar{x}) \left(\left(\frac{\bar{x}(1-\bar{x})}{s^2} \right) - 1 \right)$$

Here $\bar{x} = \frac{1}{\eta(i, j)} \sum_{k=1}^{\eta(i, j)} x_k$, $s^2 = \frac{1}{\eta(i, j)} \sum_{k=1}^{\eta(i, j)} (x_k - \bar{x})^2$. There are two issues with estimation in this manner, (1) the estimates are not defined for $\eta \leq 1$, and (2) for low values of η , estimation is poor. Yet, it is realistic for some categories to have few or no training samples for a given color pair, where estimation will be either poor or impossible respectively. But, low occurrence of neighboring segments of certain color pairs in the training set may or may not mean they will not occur in test images. To be safe, instead of penalizing the occurrence such color pairs in test images, we treat them as “unknown”. To achieve this, we estimate parameters α'_k and β'_k for the distribution of all Δ/Θ ratios across all color pairs within a given category k of training images, and store them in the models as prior distributions.

During categorization, we segment a test image in exactly the same way we performed the training. With the segmented image, we obtain the set of color interactions characterized by Δ/Θ values for each segment boundary. For a given sample $x = \Delta/\Theta$ coming from color pair (i, j) in the test image, we compute its probability of belonging to a category k . Denoting the stored parameters for the color pair (i, j) for model k as α, β and η , we have

$$P_{sc}(x|k) = \begin{cases} f(x|\alpha'_k, \beta'_k), & \eta \leq 1 \\ \frac{\eta}{\eta+1} f(x|\alpha, \beta) + \frac{1}{\eta+1} f(x|\alpha'_k, \beta'_k), & \eta > 1 \end{cases}$$

where P_{sc} is the conditional p.d.f. for the S-C model. This is typically done in statistics when the amount of confidence in some estimate is low. A weighted probability is computed instead of the original one, weights varying with the number of samples used for estimation. When η is large, $\eta/(\eta+1) \rightarrow 1$ and hence the distribution for that specific color pair exclusively determines the probability. When η is small, $1/(\eta+1) > 0$ in which case the probability from the prior distribution is given considerable importance. This somewhat solves both the problems of undefined and poor parameter estimates. It also justifies the need for storing the number of samples η as part of the models.

The S-C model is estimated for each training category $k \in \{1 \dots C\}$. Each model consists of $3S(S-1)$ parameters $\{\alpha_k(i, j), \beta_k(i, j), \eta_k(i, j), i \in \{1 \dots S\}, j \in \{1 \dots S\}, i \neq j\}$, and parameters for the prior distribution, α'_k and β'_k as explained. This set of parameters constitute θ_k , the parameter set for category k . The feature extraction process

$\mathfrak{S}(I)$ generates the Δ/Θ ratios and the corresponding color-pairs for a given image I . We thus obtain a collection of D (varying with each image) feature vectors $\{x_1, \dots, x_D\}$, where each $x_d = \{\Delta_d/\Theta_d, i_d, j_d\}$. We assume conditional independence of each x_d . Hence, using equation (1), we have

$$\ell_{sc}(I|M_k) = \sum_{d=1}^D \log P_{sc}(\Delta_d/\Theta_d | \theta_k(i_d, j_d)). \quad (8)$$

2.1.1 Fast Computation of S-C model Features

We wish to have a low complexity algorithm to compute the Δ/Θ ratios for a given image (training or testing). As discussed, these ratios can be computed in a naive manner as follows: (1) Segment the image by nearest neighbor assignments followed by connected component labeling. (2) For each segment, compute its perimeter (Θ), and length of border (Δ) shared with each neighboring segment. (3) Compute the Δ/Θ ratios and return them (along with the corresponding color pairs) for modeling or testing, whichever the case. This algorithm can be sped as follows. Denote the segment identity associated with each pixel (x, y) by $s(x, y)$. Each (x, y) is either (1) an interior pixel, not bordering any segment or the image boundary, (2) a pixel that is either bordering two or more segments, or is part of the image boundary, or (3) a pixel that has no neighboring segments but is part of the image boundary. Pixels in (1) do not contribute to the computation of Δ or Θ and hence can be ignored. Pixels in (2) are both part of the perimeter of segment $s(x, y)$ and the borders between $s(x, y)$ and each neighboring segment s_k (i.e., $(i, j) \oplus s_k$). Pixels in (3) are only part of the perimeter of $s(i, j)$. Based on this, a *single-pass algorithm* for computing the S-C feature vector $\{x_1, \dots, x_D\}$ of an image I is presented in Figure 4.

```

Pair(1...P, 1...P) ← 0 [P = No. of segments]
Perim(1...P) ← 0
for each pixel (x, y) in I
  k ← 0; Z ← ∅
  for each 8-neighbor (x', y') ∈ D(x, y)
    if (x', y') is inside image boundary
      if s(x', y') ≠ s(x, y) and s(x', y') is unique
        Z ← Z ∪ s(x', y')
      k ← 1
    else
      k ← 1
  for each s' ∈ Z
    Pair(s(x, y), s') ← Pair(s(x, y), s') + 1
  if k = 1
    Perim(s(x, y)) ← Perim(s(x, y)) + 1
  [Now Generate Δ/Θ ratios : ℑ(I) = {x1, ..., xD}]
d ← 0
for i ← 1 to P
  for j ← 1 to P
    if Pair(i, j) > 0 [[i, j] segments shared border]
      d ← d + 1
      Δd ← Pair(i, j); Θd ← Perim(i)
      xd ← Δd/Θd
      return [xd, G(i), G(j)]
[G(·) - maps segment to color]

```

Figure 4: Single-pass algorithm for computing S-C model features.

The set of ordered triplets $[x_d, G(i), G(j)]$ can now be used to build Beta distributions with parameters

$\alpha(G(i), G(j))$ and $\beta(G(i), G(j))$, provided the number of samples $\eta(G(i), G(j)) > 1$. Besides the two-pass connected component labeling, only a single scanning of the image is required to compute these features. It is not hard to see that this algorithm can be embedded into the two-pass connected component labeling algorithm to further improve speed. Note that though the asymptotic order of complexity remains the same, the improved computational efficiency becomes significant as the image database size increases.

2.2 Color-Texture (C-T) Models

Many image categories, especially those that do not contain specific objects, can be best described by their color and texture distributions. There may not even exist a well-defined structure per se, for high-level categories such as China and Europe, but the overall ambience formed the colors seen in these images often help identify them. A mixture of multivariate Gaussians is used to model the joint color-texture feature space for a given category. The motivation is simple; in many cases, two or more representative regions in the color/texture feature space can represent the image category best. For example, beach pictures typically have one or more yellow areas (sand), a blue non-textured area (sky), and a blue textured region (sea). Gaussian mixture models (GMMs) are well-studied, with many tractable properties in statistics. Yet, these simple models have not been widely exploited in generic image categorization. Recently, GMMs have been used effectively for outdoor scene classification and annotation [18]. After model estimation, likelihood computation at testing is typically very fast.

Let a Gaussian mixture model have λ components, each of which is parameterized by $\theta_k = \{a_k, \mu_k, \Sigma_k\}$, $k = 1 \dots \lambda$, where a is the component prior, μ is the component mean, and Σ is the component covariance matrix. Given a feature vector $x \in \mathbb{R}^m$, the joint probability density function of component k is defined as

$$f(x|\theta_k) = \frac{1}{\zeta} \exp\left(\frac{-(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)}{2}\right)$$

where $\zeta = \sqrt{(2\pi)^m \|\Sigma_k\|}$. Hence the mixture density is $f(x) = \sum_{k=1}^{\lambda} a_k f(x|\theta_k)$. The feature vectors in the C-T model are the same as those used in [17], where a detailed description can be found. Each training image is divided into 4×4 non-overlapping blocks, and a 6-dimensional feature vector x is extracted from each block. Three components are the mean *LUV* color values within the block, and the other three are moments of Daubechies-4 wavelet based texture coefficients. Our feature extraction process \mathfrak{S} for the color-texture model thus takes in an image I and computes $\mathfrak{S}(I) = \{x_1, \dots, x_D\}$, $x_i \in \mathbb{R}^6$, D depending on the image dimensions.

The parameters of GMMs are usually estimated iteratively using the Expectation-Maximization (EM) algorithm, since there is no closed form solution to its maximum likelihood based estimate. Here, for each category c , the feature vectors $\mathfrak{S}(I_i^c)$ (or a subset) obtained from each training image I_i^c , $i = 1 \dots N$ are used for building model M_c . We use Bouman's 'cluster' package [2] to do the modeling. This package allows λ to be specified, and then adaptively chooses the number of clusters less than or equal to λ , using Rissanen's minimum description length (MDL) criteria. Thus we use the feature set $\{\mathfrak{S}(I_1^c), \dots, \mathfrak{S}(I_N^c)\}$ and

λ to generate C models M_c , $c = 1 \dots C$. A test image I is thus represented by a collection of feature vectors $\mathfrak{S}(I) = \{x_1, \dots, x_D\}$, $x_d \in \mathbb{R}^6$. Here, our conditional independence assumption given model M_c is based on ignoring spatial dependence of the block features. However, spatial dependence is expected to be captured by the S-C model. Thus, based on Eq. 1, the log-likelihood of M_c generating I is

$$\ell_{ct}(I|M_c) = \sum_{d=1}^D \log\left(\sum_{k=1}^{\lambda} a_k^c f(x_d|\mu_k^c, \Sigma_k^c)\right). \quad (9)$$

For both models, the predicted sets of categories for a given image I are obtained in rank order by sorting them according to the likelihood scores $\ell_{sc}(I|\cdot)$ and $\ell_{ct}(I|\cdot)$ respectively.

3. ANNOTATION AND RETRIEVAL

The categorization results are utilized to perform image annotation. Tagging an image with any given word entails three considerations, namely (1) frequency of occurrence of the word among the evidence provided by categorization, (2) saliency of the given words, i.e., as is traditional in the text retrieval community, a frequently occurring word is more likely than a rare word to appear in the evidence by chance, and (3) the congruity (or fitness) of the word with respect to the entire set of words under consideration. Suppose we have a 600 category training image dataset (the setting for all our retrieval experiments), each category annotated by 3 to 5 tags, e.g., [sail, boat, ocean] and [sea, fish, ocean], with many tags shared among categories. Initially, all the tags from each category are pooled together. Tag saliency is measured in a way similar to computing inverse document frequency (IDF) in the document retrieval domain. The total number of categories in the database is C . We count the number of categories which contain each unique tag t , and denote it by $F(t)$. For a given test image I , the S-C models and the C-T models independently generate ranked lists of predicted categories. We choose the top 10 categories predicted by each model and pool them together for annotation. We denote the union of all unique words from both models by $U(I)$, which forms the set of *candidate tags*. Let the frequency of occurrence of each unique tag t among the top 10 model predictions be $f_{sc}(t|I)$ and $f_{ct}(t|I)$ respectively.

WordNet [23] is a semantic lexicon which groups English words into sets of synonyms and records the semantic relations among the synonym sets. Based on this ontology, a number of measures of semantic relatedness among words have been proposed. A measure that we empirically observe to produce reasonable relatedness scores among common nouns is the Leacock and Chowdrow (LCH) measure [6], which we use in our experiments. We convert the relatedness measure r_{LCH} from a [0.365, 3.584] range to a distance measure d_{LCH} in the [0, 24] range using the mapping $d_{LCH}(t_1, t_2) = \exp(-r_{LCH}(t_1, t_2) + 3.584) - 1$ for a pair of tags t_1 and t_2 . Inspired by the idea proposed in [14], we measure congruity for a candidate tag t by

$$G(t|I) = \frac{d_{tot}(I)}{d_{tot}(I) + |U(I)| \sum_{x \in U(I)} d_{LCH}(x, t)} \quad (10)$$

where $d_{tot}(I) = \sum_{x \in U(I)} \sum_{y \in U(I)} d_{LCH}(x, y)$ measures the all-pairwise semantic distance among candidate tags,

generating scores in the $[0, 1]$ range. Essentially, a tag that is semantically distinct from the rest of the words predicted will have a low congruity score, while a closely related one will have a high score. The measure can potentially remove noisy and unrelated tags from consideration. Having computed the three measures, for each of which higher scores indicate greater support for inclusion, the overall score for a candidate tag is given by a linear combination as follows:

$$R(t|I) = a_1 f(t|I) + \frac{a_2}{\log C} \log \left(\frac{C}{1 + F(t)} \right) + a_3 G(t|I) \quad (11)$$

Here, $a_1 + a_2 + a_3 = 1$, and $f(t|I) = b f_{sc}(t|I) + (1-b) f_{ct}(t|I)$ is the key model combination step for the annotation process, linearly combining the evidence generated by each model toward tag t . Experiments show that combination of the models helps in annotation significantly over either model. The value of b is a measure of relative confidence in the S-C model. A tag t is chosen for annotation only when its score is within the top ε percentile among the candidate tags, where ε intrinsically controls the number of annotations generated per image. Hence, in the annotation process, we are required to specify values of four parameters, namely $(a_1, a_2, b, \varepsilon)$. We perform annotation on a validation set of 1000 images and arrive at desirable values of precision/recall for $a_1 = 0.4$, $a_2 = 0.2$, $b = 0.3$, and $\varepsilon = 0.6$.

3.1 Performing Annotation-driven Search

We retrieve images using automatic annotation and the WordNet-based bag of words distances. Whenever tags are missing in either the query image or the database, automatic annotation is performed, and bag of words distance between query image tags and the database tags are computed. The images in the database are ranked by relevance based on this distance. We briefly describe the bag of words distance used in our experiments, inspired by the *average aggregated minimum* (AAM) distance proposed in [16]. The WordNet-based LCH distance $d_{LCH}(\cdot, \cdot)$ is again used to compute semantic distances between bags of words in a robust manner. Given two bags of words, $W_i = \{w_{i,1}, \dots, w_{i,m_i}\}$ and $W_j = \{w_{j,1}, \dots, w_{j,m_j}\}$, we have the distance between them

$$\widehat{d}(W_i, W_j) = \frac{1}{2m_i} \sum_{k=1}^{m_i} \bar{d}(w_{i,k}, W_j) + \frac{1}{2m_j} \sum_{k=1}^{m_j} \bar{d}(w_{j,k}, W_i) \quad (12)$$

where $\bar{d}(w_{i,k}, W_j) = \min_{w_{j,l} \in W_j} d_{LCH}(w_{i,k}, w_{j,l})$. Naturally, $\widehat{d}(W_i, W_i)$ is equal to zero. In summary, the approach attempts to match each word in one bag to the closest word in the other bag and compute the average semantic distance over all such closest matches.

4. EXPERIMENTAL RESULTS

Experiments are performed at all three stages, (1) image categorization, (2) categorization-based annotation, and (3) annotation-driven image retrieval under the scenarios described in Section 1.2. A set of 54,000 Corel Stock photos encompassing 600 image categories, and a 1000-image database from Yahoo! Flickr form the datasets for our experiments. The S-C and C-T models for the 600 categories are built on a training set of 24,000 images, 40 images per category. Each category is tagged with 3 – 5 words, identical to the tagging in [17]. For the S-C models, $S = 20$ color clusters are used (refer to Section 2.1).

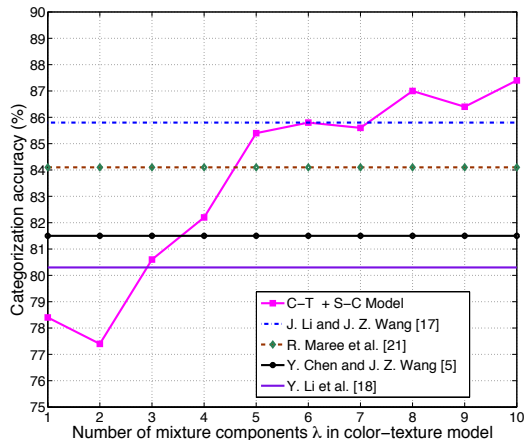


Figure 5: Categorization accuracies for the 10-class experiment are shown. Performance of our combined S-C + C-T model is shown with varying number of mixture components in the C-T model. Previously reported best results shown for comparison.

4.1 Categorization Performance

Typically, fusion of discriminative models can be performed using a number of different ways, such as those reported in [15]. For generative models, one way to perform this kind of fusion is to use *fisher kernels* to obtain a discriminative framework from these generative models, and combine classifiers thereof. Exploration of these fusion techniques is part of our future work. Instead, we employ a simple combination strategy [11] that is capable of producing impressive performance. For an image I , we rank each category $k = 1..C$ based on likelihoods from both models to get ranks $\pi_{sc}(k)$ and $\pi_{ct}(k)$. A linearly combined score is then obtained for each category, $\pi(k) = \sigma \pi_{sc}(k) + (1-\sigma) \pi_{ct}(k)$, where σ is chosen through validation. Finally, the categories are ranked based on these scores, and the top category is predicted for I .

Categorization performance is assessed based on two datasets. The first of them is a standard 10-class image dataset used to assess categorization performance in [5, 18, 21, 17]. Training is performed with $N = 40$ images per category, and 10 images per category form the validation set for choosing a value of σ . A set of 50 images per category outside of the training/validation set are chosen for testing. With a chosen value of $\sigma = 0.2$, the accuracies are computed while varying the number of mixture components λ in the C-T model. These results, along with previously reported accuracies, are shown in Figure 5. Our combined model is found to outperform previously reported results for $\lambda \geq 6$. The best performance is achieved for $\lambda = 10$ at 87.4%, which outperforms previous results for this dataset. Not surprisingly, as λ increases, the mixture models characterizing each category become increasingly precise.

Our second dataset is the same set of 600 category Corel images used in the ALIP system [17]. Each of the categories are manually annotated with 3 – 5 labels using 417 unique words. A C-T model and an S-C model is built for each category using 40 training images, as done in ALIP. We

Table 1: Categorization results on 27,000 images for the 600 category dataset.

Method	Top 1	Top 2	Top 3	Top 4	Top 5
S-C + C-T	14.38%	19.25%	22.69%	25.25%	27.25%
ALIP [17]	11.88%	17.06%	20.76%	23.24%	26.05%

compute combined classification accuracies ($\sigma = 0.2, \lambda = 10$) on 27,000 test images (45 images per category), and accuracies are computed for top 1–5 matches, as performed in ALIP. Here, a top r match means that the original category is within the top r ranked categories. Classification accuracies on these experiments are summarized in Table 1. Note that the performance of say the top 5 matches become important when they are all utilized for image annotation.

Our combined model improves upon the categorization accuracies reported in [17]. Note that a 2% improvement amounts to correct categorization of 540 more images. We now compare the categorization speed. Our system takes about 26 seconds to build an S-C and 106 seconds to build a C-T model. To predict the top 5 ranked categories for a given test image, our system takes 11 seconds. Our system is orders of magnitude faster than the ALIP system which takes about 30 minutes to build a model, and 20 minutes per test image on comparable machines. Other systems do not report exact computation times; however, the CBSA system [4] involves estimating Bayes point machines which is computationally intensive; systems such as [1, 12, 13] that depend on computationally intensive segmentation algorithms are bounded by the segmentation speeds both in modeling and testing.

4.2 Annotation Performance

We generate annotations on both the Corel dataset and the Yahoo! Flickr dataset using the models trained with Corel images. The speed of annotation is dependent upon categorization speed, and hence annotation is near-real-time as well. With the parameter set specified in Section 3, the average number of tags generated per image over the 10,000 randomly chosen Corel images is 7.16. We define *annotation precision* and *annotation recall* as

$$\text{Annotation Precision} = \frac{\# \{\text{correct tags predicted}\}}{\# \{\text{tags predicted}\}}$$

$$\text{Annotation Recall} = \frac{\# \{\text{correct tags predicted}\}}{\# \{\text{correct tags}\}}.$$

The average annotation precision over the 10,000 Corel images is 25.38%, while annotation recall is 40.69%. What this translates to is that on an average, 1 in 4 words predicted by our system is a correct tag, and 2 in 5 correct tags are predicted by our system.

The assessment is then extended to an image collection outside of the training database (Corel), namely 1,000 tagged Yahoo! Flickr images. The database has considerable visual coherence with Corel images, hence our learnt models are used on them for annotation. Browsing through the results, we found that although most automatically generated tags are semantically meaningful, and the results are very encouraging in general, a numerical precision and recall score would not reflect this well enough. Many of the original tags are proper nouns (e.g., names of buildings, cities, and people). Instead, we present a

sampling of the annotations generated by our approach, and the corresponding Flickr tags. These results can be seen in Figure 6.

4.3 Annotation-driven Image Search

For retrieval, the assumption is that either the databases is partially tagged, or the search is performed on an image database visually coherent with the ‘knowledge base’ (in our case the learnt models from the Corel dataset). In our experiments, the Corel image dataset is treated as a partially tagged image database, where experiments are performed on other Corel images not used for training the models.

Our retrieval experiments are performed using the 600 trained S-C and C-T models, as described in Section 4.1. In all cases, annotation is first performed using the categorization results, as explained in Section 3. We consider the three image search scenarios described in Section 1.2. For this purpose, we build an image database of 10,000 images chosen from among the 600 tagged Corel categories using a pseudo-random generator. For each scenario, we compare results of our annotation-driven image search strategy with (1) alternative CBIR-based strategies, and (2) random annotation based retrieval (to serve as the lower bound of performance). For the CBIR-driven strategies, we use the IRM distance used in the SIMPLiCity system [26] to get around the missing tag problem in the databases and queries. The choice and parameter selection for the alternative strategies have been chosen empirically over a range of methods and values (details skipped due to lack of space). Performance is assessed using the standard retrieval precision and recall measures used in information retrieval. Precision is the proportion of retrieved images that are relevant, and recall is the proportion of relevant images that are retrieved. An image is considered *relevant* if there is an overlap between the original tags of the query image or query word (as the case may be) and the original tags of the retrieved image.

Scenario 1: Under this scenario, the database does not have any tags. Queries may either be in the form of one or more keywords, or tagged images. Keyword queries on an untagged image database is a key problem in real-world image search. In our experiments, a total of 40 pairs of query words are chosen randomly from among the 417 unique words in our Corel dataset. In our *annotation-driven strategy*, we perform retrieval by first automatically annotating the database, and then retrieving images based on bag of the words distances (Section 3.1) between query tags and our annotation. The alternative *CBIR-based strategy* used for comparison is as follows: Without any image as query, CBIR cannot be performed directly on query keywords. Instead, the system is provided access to a knowledge base of tagged Corel images. A random set of 3 images for each query word is chosen from the knowledge base, and the IRM distances between these images and the database are computed. The average IRM distance over the 6 images is then used for retrieval from the database in each case. These results, along with retrieval after randomly annotating the database, are reported in Figure 7(a). Clearly, our method significantly outperforms the alternative strategy. Moreover, the retrieval performance with our approach is very encouraging.

Scenario 2: The query is an untagged image, and the database is tagged. What is interesting is that the









				
Our Labels	sky, city, modern, building, Boston	door, pattern, Europe, historical building, city	train, car, people, life, city	man, office, indoor, fashion, people
Flickr Labels	Amsterdam, building, Mahler4, Zuidas	Tuschinski, Amsterdam	honeymoon, Amsterdam	hat, Chris, cards, funny
				
Our Labels	lake, Europe, landscape, boat, architecture	lion, animal, wild life, Africa, super-model	speed, race, people, Holland, motorcycle	dog, grass, animal, rural, plant
Flickr Labels	Amsterdam, canal, water	leopard, cat, snagged photo, animal	Preakness, horse, jockey, motion, unfound photo, animal	Nanaimo Torgersons, animal, Quinn, dog, cameraphone

Figure 6: Sample annotation results on Yahoo! Flickr photos.

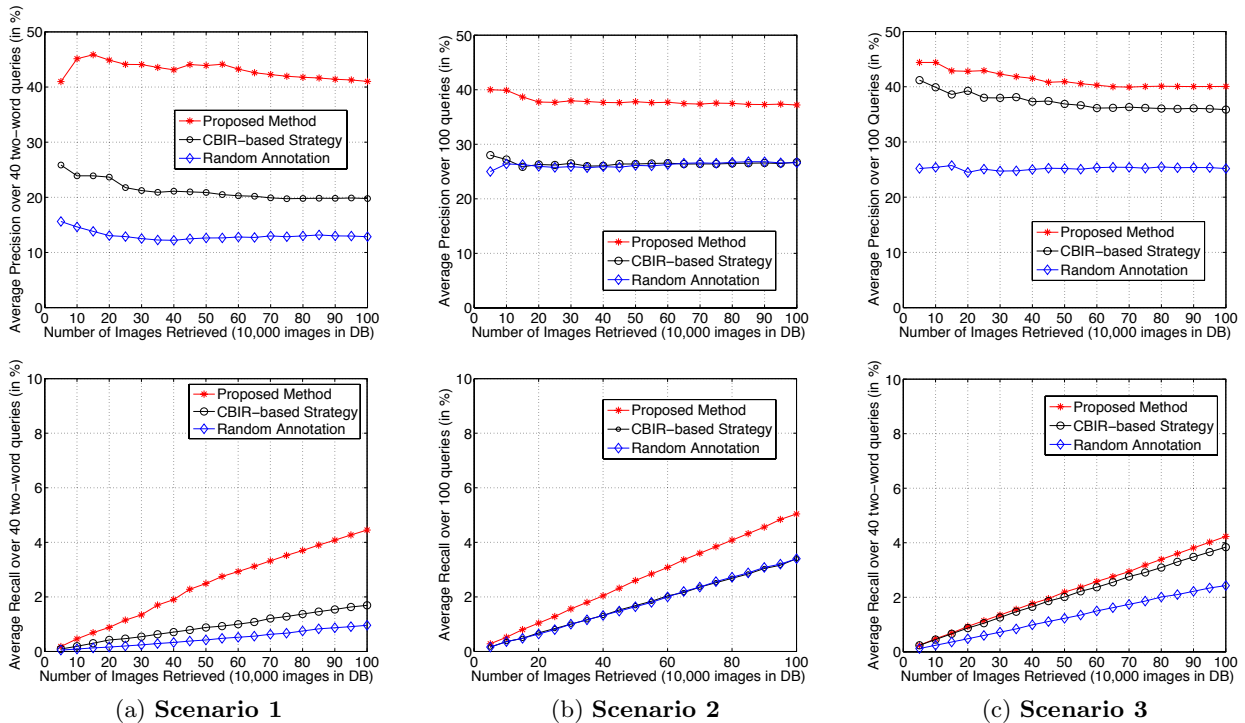


Figure 7: Precision (above) and Recall (below) scores for annotation-driven image search under three different scenarios. (a) Keyword queries on an untagged database. (b) Untagged image queries on a tagged image database. (c) Untagged image queries on an untagged database. Random annotation based results are provided for visualizing the lower bound on performance.

database may be any kind of document database and need not be restricted to words. Our *annotation-driven strategy* case is performed by first annotating the query

automatically, and then ranking the database using bag of words distance. Although our experiments are limited to image databases, this framework provides an opportunity to

search a document database with an untagged image query as well. A set of 100 randomly chosen query images are tested on the 10,000 image database. The alternative *CBIR-based strategy* used for comparison is as follows: The IRM distance is used to retrieve 5 (empirically observed to be the best) most visually similar images, and the union of all their tags are filtered using Eq. 11, where $f(t|I)$ is computed directly from the tags of the 5 images. Retrieval then proceeds with these annotations in a manner identical to our approach. The results, along with the random annotation scheme, are shown in Figure 7(b). As can be observed, our strategy has a significant performance advantage over the alternate strategy. Note that the CBIR-based strategy performs almost as poorly as the random scheme, which can likely be attributed to the instability of directly using CBIR for annotation.

Scenario 3: In this case, neither the query image nor the image database is tagged. A total of 100 random image queries are tested on the 10,000 image database. Our *annotation-driven strategy* is simply to annotate both the query as well as the image database automatically, and then performing bag of words based retrieval. In the absence of any semantic information, the *CBIR-based strategy* used for comparison is essentially a standard use of the IRM distance to rank images based on the query. The results, shown in Figure 7(c), essentially highlight the advantage of performing annotation-based semantic ranking over performing a straightforward visual similarity based retrieval. Clearly, the acquired knowledge captured through the learnt models accounts for this improvement.

5. CONCLUSIONS

We have proposed a novel annotative-driven image retrieval approach with a demonstrated potential for real-world usage. Retrieval is driven by WordNet-based bag of words distances on automatically generated image tags. Annotation is performed using image categorization. A novel generative model based near-realtime categorization method is proposed, which is shown to improve upon best reported image categorization results. Experimental results for annotation and retrieval on Corel images and a Yahoo! Flickr dataset show considerable promise. Annotation-driven retrieval is found to significantly surpass CBIR-based retrieval methods in performance on all scenarios considered, including the case where neither the query nor the database is tagged. Future work includes moving from near-realtime to realtime by filtering techniques. In particular, we can screen the image categories by fast approximation methods, retaining only a small set for accurate computation. We wish to assess the performance of our system on Web images as well. Combining categorization/annotation with traditional CBIR for image search is a potential new direction.

The research is supported in part by the US National Science Foundation. We thank David M. Pennock at Yahoo! for providing test images.

6. REFERENCES

- [1] K. Barnard, P. Duygulu, N. Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching Words and Pictures," *Journal of Machine Learning Research*, 3:1107-1135, 2003.
- [2] C. A. Bouman, "Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures," Software Package. <http://www.ece.purdue.edu/~bouman>.
- [3] G. Carneiro and N. Vasconcelos, "A Database Centric View of Semantic Image Annotation and Retrieval," *SIGIR*, 2005.
- [4] E. Chang, G. Kingshly, G. Sychay, and G. Wu, "CBSA: Content-based Soft Annotation for Multimodal Image Retrieval Using Bayes Point Machines," *IEEE Trans. on Circuits and Systems for Video Tech.*, 13(1):26-38, 2003.
- [5] Y. Chen and J. Z. Wang, "Image Categorization by Learning and Reasoning with Regions," *Journal of Machine Learning Research*, 5(2004):913-939.
- [6] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Identification," *Fellbaum*, 1998.
- [7] F.-F. Li and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *IEEE CVPR*, 2005.
- [8] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli Relevance Models for Image and Video Annotation," *IEEE CVPR*, 2004.
- [9] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang, "Manifold-Ranking Based Image Retrieval," *ACM Multimedia*, 2004.
- [10] X. He, "Incremental Semi-Supervised Subspace Learning for Image Retrieval," *ACM Multimedia*, 2004.
- [11] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans on PAMI*, 16(1):66-75, 1994.
- [12] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic Image Annotation and Retrieval Using Cross-media Relevance Models," *SIGIR*, 2003.
- [13] R. Jin, J. Y. Chai, and L. Si, "Effective Automatic Image Annotation Via A Coherent Language Model and Active Learning," *ACM Multimedia*, 2004.
- [14] Y. Jin, L. Khan, L. Wang, and M. Awad, "Image Annotations By Combining Multiple Evidence & WordNet," *ACM Multimedia*, 2005.
- [15] J. Kittler, M. Hatef, R. P. W. Duin, and J. Mata, "On Combining Classifiers," *IEEE Trans. on PAMI*, 20(3):226-239, 1998.
- [16] J. Li, "A Mutual Semantic Endorsement Approach to Image Retrieval and Context Provision," *MIR Workshop, ACM Multimedia*, 2005.
- [17] J. Li and J. Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Trans. on PAMI*, 25(19):1075-1088, 2003.
- [18] Y. Li, L. G. Shaprio, and J. A. Bilmes, "A Generative/Discriminative Learning Algorithm for Image Classification," *ICCV*, 2005.
- [19] Y.-Y. Lin, T.-L. Liu, and H.-T. Chen, "Semantic Manifold Learning for Image Retrieval," *ACM Multimedia*, 2005.
- [20] W. Liu and X. Tang, "Learning an Image-Word Embedding for Image Auto-Annotation on the Nonlinear Latent Space," *ACM Multimedia*, 2005.
- [21] R. Marée, P. Geurts, J. Piater, and L. Wehenkel, "Random Subwindows for Robust Image Classification," *CVPR*, 2005.
- [22] F. Monay and D. Gatica-Perez, "On Image Auto-Annotation with Latent Space Models," *ACM Multimedia*, 2003.
- [23] G. Miller, "WordNet: A Lexical Database for English," *Comm. of the ACM*, 38(11):39-41, 1995.
- [24] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. on PAMI*, 22(12):1349-1380, 2000.
- [25] T. Volkmer, J. R. Smith, and A. Natsev, "A Web-based System for Collaborative Annotation of Large Image and Video Collections," *ACM Multimedia*, 2005.
- [26] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLicity: Semantics-Sensitive Integrated Matching for Picture Libraries," *IEEE Trans. on PAMI*, 23(9):947-963, 2001.
- [27] X.-J. Wang, W.-Y. Ma, G.-R. Xue, and X. Li, "Multi-Model Similarity Propagation and its Application for Web Image Retrieval," *ACM Multimedia*, 2004.