Learning the Semantics of Words and Pictures

Kobus Barnard and David Forsyth
Computer Division, University of California, Berkeley
{kobus, daf}@cs.berkeley.edu

Abstract

We present a statistical model for organizing image collections which integrates semantic information provided by associated text and visual information provided by image features. The model is very promising for information retrieval tasks such as database browsing and searching for images based on text and/or image features. Furthermore, since the model learns relationships between text and image features, it can be used for novel applications such as associating words with pictures, and unsupervised learning for object recognition.

1. Introduction

We present a method which organizes image databases using both image features and associated text. By integrating the two kinds of information during model construction, the system learns links between the image features and semantics which can be exploited for better browsing (§3.1), better search (§3.2), and novel applications such as associating words with pictures, and unsupervised learning for object recognition (§4). The system works by modeling the statistics of word and feature occurrence and co-occurrence. We use a hierarchical structure which further encourages semantics through levels of generalization, as well as being a natural choice for browsing applications. An additional advantage of our approach is that since it is a generative model, it implicitly contains processes for predicting image components—words and features—from observed image components. Since we can ask if some observed components are predicted by others, we can measure the performance of the model in ways not typically available for image retrieval systems (§4). This is exciting because an effective performance measure is an important tool for further improving the model (§5).

A number of other researchers have introduced systems for searching image databases. This work includes search by text [1, 2], search by image feature similarity [3-6], search by segment features [7], search for specific types of images using more compressive methods [8, 9], and search by image sketch [1]. A few systems combine text and image data. Search using a simple conjunction of keywords and image features is provided in Blobworld [7]. Here the image segment color is translated in a pre-processing step into one of a handful of color categories. Thus, image feature search becomes a text search, and standard database systems can be used for the query. This is efficient, but potential for more sophisticated use of image features is limited. Webseer [10] uses similar ideas for query of images on the web, but also indexes the results of a few automatically estimated image features. These include whether the image is a photograph or a sketch and notably the output of a face finder. Going further, Cascia et al integrate some text and histogram data in the indexing [11]. Others have also experimented with using image features as part of a query refinement process [12]. Enser and others have studied the nature of the image database query task [13-15]. The model we build on is developed in [16]. Others also argue for statistical models of data for image retrieval [17]. Finally, in the area of using associated text for image understanding the work of Srihari and others [18-22] bears mentioning.

For the image retrieval component of this work we insisted that browsing was well supported. This is in contrast with many existing systems where the main access to the images is through query. This puts the burden on the user to pose the correct question, and the system to provide the correct prompts. It is easier for the user to find an image of interest if some structure can be imposed on the collection and exposed to the user. Other work emphasizing this philosophy include the application of multidimensional scaling using the Earth Mover's distance to image displays [23]. Our interest in browsing leads us to consider a hierarchical model which imposes a coarse to fine, or general to specific, structure on the image collection. Such structure is part of semantics, and therefore we propose that a hierarchical system is better poised to capture semantics than a flat one.

2. Modeling Image Dataset Statistics

Our model is a generative hierarchical model, inspired by one proposed for text by Hofmann [16, 24]. This model is a hierarchical combination of the assymetric clustering model which maps documents into clusters, and the symmetric clustering model which models the joint distribution of documents and features (the "aspect" model). The data is modeled as being generated by a fixed hierarchy of nodes, with the leaves of the hierarchy corresponding to clusters. Each node in the tree has some probability of generating each word, and similarly, each node has some probability of generating an image segment with given features. The documents belonging to a given cluster are modeled as being generated by the nodes along the path from the leaf corresponding to the cluster, up to the root node, with each node being weighted on a document and cluster basis. Conceptually a document belongs to a specific cluster, but given finite data we can only model the probability that a document belongs to a cluster, which essentially makes the clusters soft. We note also that clusters which have insufficient membership are extinguished, and therefore, some of the branches down from the root may end prematurely.

The model is illustrated further in Figure 1. For this work we model documents as a sequence of words and a sequence of segments, with the segments being taken from the Blobworld representation [7]. To the extent that the sunset image illustrated is in the third cluster, as indicated in the figure, its words and segments are modeled by the nodes along the path shown. Taking all clusters into consideration, the document is modeled by a sum over the clusters, weighted by the probability that the document is in the cluster. Mathematically, the process for generating the set of observations D associated with a document d can be described by

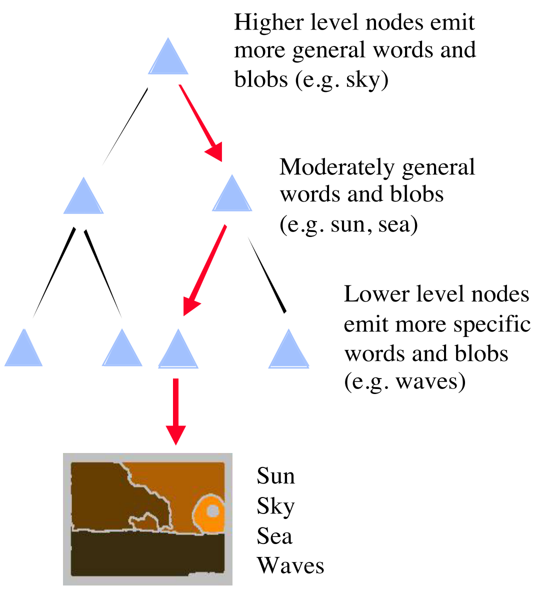

Higher level nodes emit more general words and blobs (e.g. sky)
Moderately general words and blobs (e.g. sun, sea)
Lower level nodes emit more specific words and blobs (e.g. waves)
Sun
Sky
Sea
Waves

Figure 1. Illustration of t he generative process implicit in the statistical model. Each document has some probability of being in each cluster. To the extent that it is in a given cluster, it is modeled by being generated by sampling from the nodes on the path to the root.

$$P(D \mid d) = \sum_{c} P(c) \prod_{i \in D} \left(\sum_{l} P(i \mid l, c) P(l \mid c, d) \right) \tag{1}$$

where c indexes clusters, i indexes items (words or image segments), and l indexes levels. Note that given the hard hierarchy, a cluster and a level together specify a node. In words, equation (1) describes a weighted sum over aspects which have tied parameters to force the hierarchical structure. Since the aspects model the joint distribution of documents and items, the weighting $P(l \mid c, d)$ is a function of the document. When we encounter a new document, we can either re-fit the model to estimate $P(l \mid c, d)$, or use a cluster specific approximation, which generally works well, and is much cheaper to compute.

The probabilities for an item, $P(i \mid l, c)$, is conditionally independent, given a node in the tree. A node is uniquely specified by cluster and level. In the case of a word, $P(i \mid l, c)$ is simply tabulated, being determined by the appropriate word counts during training. For image segments, we use Gaussian distributions over a number of features capturing some aspects of size, position, colour, texture, and shape. These features taken together form a feature vector X. Each node, subscripted by cluster c, and level l, specifies a probability distribution over image segments by the usual formula:

$$P(\mathbf{x} \mid c, l) = \frac{1}{(2\pi)^{d/2} \det\left(\Sigma_{c,l}\right)^{1/2}} \exp\left(-\frac{1}{2} \left(\mathbf{x} - \mathbf{u}_{c,l}\right) \Sigma^{-1} \left(\mathbf{x} - \mathbf{u}_{c,l}\right)^{T} \right)$$

In this work we assume independence of the features, as learning the full covariance matrix leads to precision problems. A reasonable compromise would be to enforce a block diagonal structure for the covariance matrix to capture the most important dependencies.

To train the model we use the Expectation-Maximization algorithm [25]. This involves introducing hidden variables $H_{d,c}$ indicating that training document d is in cluster c, and $V_{d,i,l}$ indicating that item i of document d was generated at level l. Additional details on the EM equations can be found in [16].

We chose a hierarchical model over several non-hierarchal possibilities because it best supports browsing of large collections of images. Furthermore, because some of the information for each document is shared among the higher level nodes, the representation is also more compact than a similar non-hierarchical one. This economy is exactly why the model can be trained appropriately. Specifically, more general terms and more generic image segment descriptions will occur in the higher level nodes because they occur more often. Of course, this assumes a loose definition of general. For example, if the word "the" is not removed from free text descriptions of images, then it will be treated as a general term.

We mention a few implementation details relevant for scalability and avoiding over-training. The training procedure, as described, can cluster a few thousand images in a few hours on a state of the art PC. Since the resource requirements, most notably memory, increase rapidly with the number of images, going significantly beyond this requires extra care. We have experimented with several approaches. In the first approach we train on a randomly selected subset of the images until the log likelihood for held out data, randomly selected from the remaining data, begins to drop. The model so found is then used as a starting point for the next training round using a second randomly selected set of images. This method helps avoid over-training as well as reducing resource usage.

A second method for reducing resource usage is to limit cluster membership. We first compute an approximate clustering by training on a subset. We then cluster the entire dataset but only maintain the probability that a point is in a cluster for the top 20 clusters. The rest of the membership probabilities are assumed to be zero for the next few iterations, at which point the top 20 clusters are re-estimated. Finally, we have experimented with first clustering with reduced tree depth to get approximate clusterings.

3. Testing and Using the Basic Model

We tested our method for stability by running the fitting process a number of times on the same data with different initial conditions, as the EM process is known to be sensitive to the starting point. The resulting clusterings had similar character, although numerical tests showed that the exact structure was in fact somewhat sensitive to the starting point. In a more important test we reran the experiment, but now hid a percentage of the data (5, 10, 15, and 20 percent) from the training process. Interestingly, this had little effect on the variance of the cluster process. Thus we conclude that the clustering process depends significantly more on the starting point than on the exact images chosen for training.

The next general test was to verify that clustering on both image segment features and text has an advantage over either alone. Figure 2 shows 16 images from a cluster found using text only, Figure 3 shows 16 images found using only image features, and Figure 4 shows 16 images from each of two adjacent clusters, found using both text and features. The cluster found using only text contains images related through the word "ocean", but within this category contains a fair variety of images. Figure 3 consists of visually similar images, but has combined red coral and red flowers. Figure 4, on the other hand, has broken the ocean images along visual lines, separating images with significant amounts of red (mostly coral), and ones which are predominantly blue. Figure 2 shows that the breakdown is lost when only words are used, and Figure 3 shows that some of the semantics are lost when only image features are used.

One reason why clustering on both the text and image segment features is generally more appropriate is simply that people relate to images using both semantic and visual content. For example, we may be interested in pictures with flowers, but we may also be interested in predominantly red flowers. The attribute red is not likely to be added as a keyword, because its meaning is inherent in the image. Using the combined text/feature strategy in a hierarchical system, images with the associated word "flowers" may be broken into associated clusters which have predominantly red flowers, predominantly yellow flowers, and predominantly green garden scenes.

The combined clustering approach is also best for an important pragmatic reason. Browsing the results of clustering with image segment features alone verifies that they can provide some semantic coherence. Therefore it is reasonable to expect that image features can sometimes provide semantic coherence even if it is not available in the words. Similarly, clustering on text alone yields some visual coherence, and it is reasonable to expect that they sometimes capture visual similarity which goes beyond the capacity of our feature set, especially in the face of segmentation errors.

3.1. Browsing

Most image retrieval systems do not support browsing, likely because it is difficult to define and implement. Rather these systems force the user to specify what they are looking for with a query. This does not help the user learn what kind of images can be found. Setting up image databases so that their content is easy to internalize and thus navigate is difficult, and normally involves much human input. One of our goals in this work is to automate this task.

A key issue to browsing is whether the clusters found make sense to the user. If the user finds the clusters coherent, then they can begin to internalize the kind of structure they represent. Furthermore, a small portion of the cluster can be used to represent the whole, and will accurately suggest the kinds of pictures that will be found by exploring that cluster further. Thus we posit that an important first test of browsing suitability is whether the clusters make sense to humans.

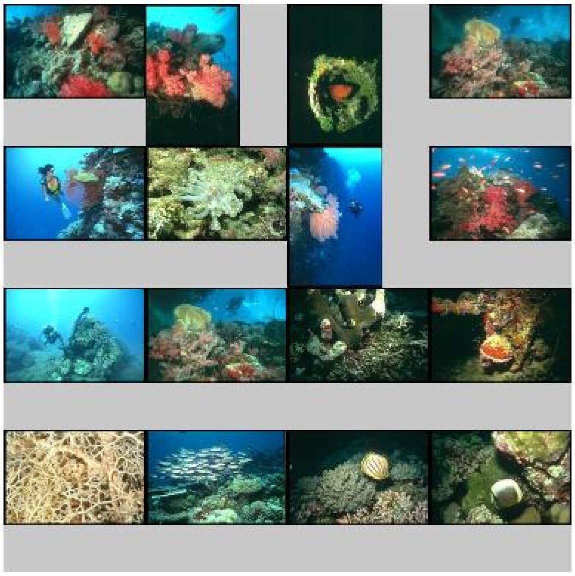


Figure 2. Some of the images from an ocean theme cluster found by clustering on text only. This cluster contains most the images in the two clusters in Figure 4, but the red corals are mixed in with the other more general ocean pictures.

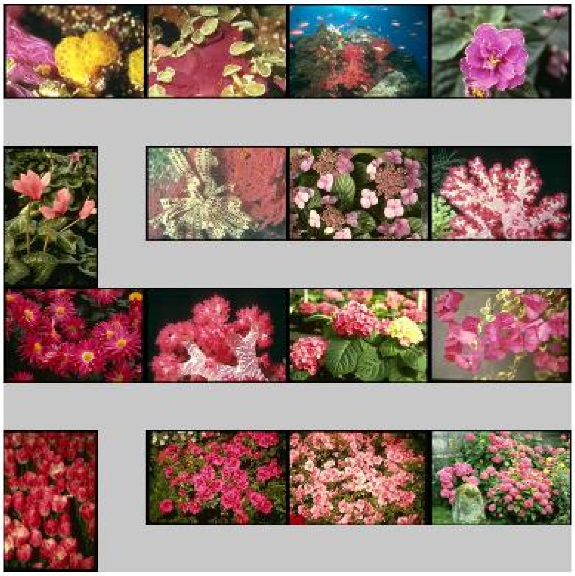


Figure 3 An example of a cluster found using image features alone. Here the coral images are found among visually similar flower images. Clearly some semantics are lost, although the grouping across semantic categories is interesting.

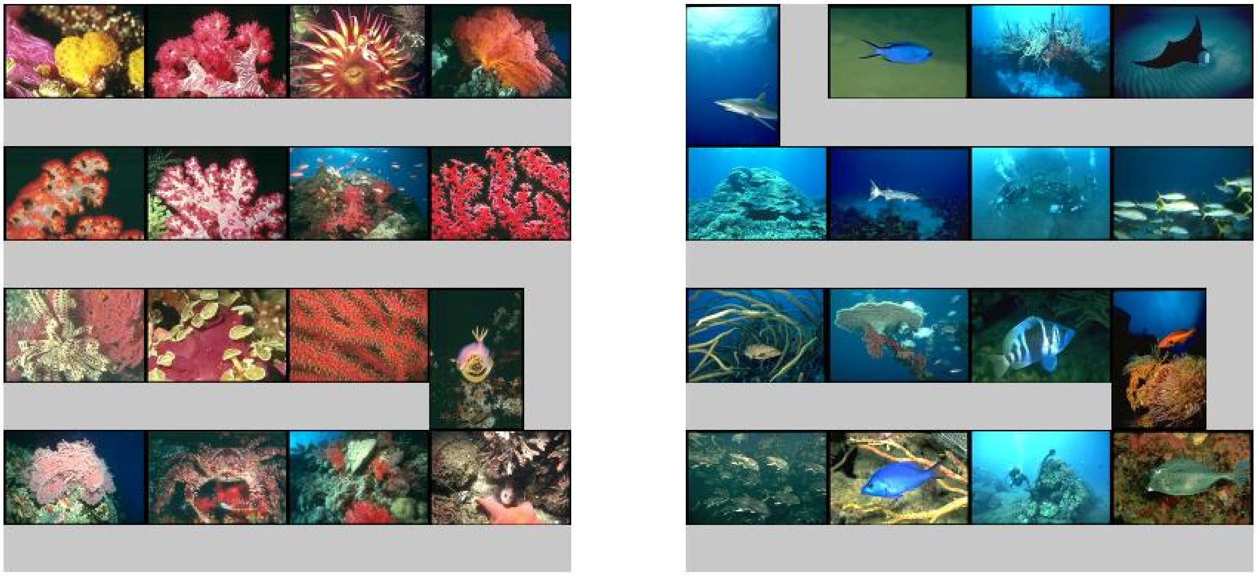


Figure 4. Two adjacent clusters computed using both text and image segments. The words clearly help ensure consistency in overall ocean/underwater themes, as well as making it so that the clusters are in neighboring leaves of the tree. Using image segment features in the clustering promotes visual similarly among the images in the clusters, and here ensures that the red coral is separated from the other ocean scenes which are generally quite blue.

To investigate this question we set up the following experiment. We first clustered roughly 3000 Corel images into 64 clusters. An additional 64 random clusters were generated from the same images. The random clusters where matched in size to randomly selected true clusters. The two sets were combined so that the clusters and non-clusters were randomly interspersed. A subject was then asked to view the clusters and say whether a candidate cluster had sufficient coherence to be considered a coherent unit. The definition of coherence was left up to the subject, but the subject was aware that both semantics and visual content could play a role, and that the occasional anomaly did not necessarily mean that a cluster should be rated as a non-cluster, if the overall coherence was adequate. We found that our subject was able to separate the two classes with an accuracy of 94%. Thus we conclude that our clusters likely have sufficient coherence to be useful for browsing. Specifically, the presentation of a small subset of the cluster as thumbnails should indicate to the user whether that region of the data is worth exploring.

3.2. Search

A second important facility for image databases is retrieval based on user queries. We wish to support queries based on text, image features, or both. We also would like the queries to be soft in the sense that the combinations of items is taken into consideration, but documents which do not have a given item should still be considered. Finally, we would like the queries to be easily specified without reference to images already found.

Our approach to searching is to compute the probability of each candidate image of emitting the query items. Defining search by the computation of probabilities very naturally yields an appealing soft query system. For example, if we query the Corel images for an image with “tiger” and “river”, we get a reasonable result despite the fact that both words do not appear together with any single image. However, “river” clusters with “water”, possibly helped by image segment features, and therefore a number of images of interest have high probability with the query. Figure 5 shows the top results of the “river” and “tiger” query.

Given a set of query items, Q, and a candidate document d, we can express the probability that a document produces the query by:

$$P(Q|d) = \sum_{c} P(Q|c,d)P(c|d)$$

$$= \sum_{c} \left\{ \prod_{q \in Q} \left(\sum_{l} P(q|l,c)P(l|c,d) \right) P(c|d) \right\} \qquad (2)$$

Documents with a high score are then returned to the user. A second approach is to find the probabilities that the query is generated by each cluster, and then sample from the clusters, weighted by the probability that the cluster emits the query. This often works reasonably well because cluster membership plays the dominant role (at least for broad trees) in generating the documents, which simply reflects the fact that the clusters are coherent. Nonetheless, we have found that the results using (2) are sometimes significantly better, and rarely worse.

We have implemented search for arbitrary combinations of words and image features. For the purposes of experimentation we specify features simply by selecting segments from our available data, and using the features of the selected segments. Clearly providing a more flexible method of specifying image features is an important next step. Part of such a system could employ a user’s selection from features suggested by a returned result set. This is as explored in the many “query by example” image retrieval systems. However, this is only one strategy, and we emphasize that our system can retrieve images without starting from an example image. This captures the users needs in some ways, but not in others. Put differently, we can query for a dog by the word “dog”, and if we want blue sky above the dog, then we can add the appropriate segment feature to the query. Working with pieces of other images is not required.

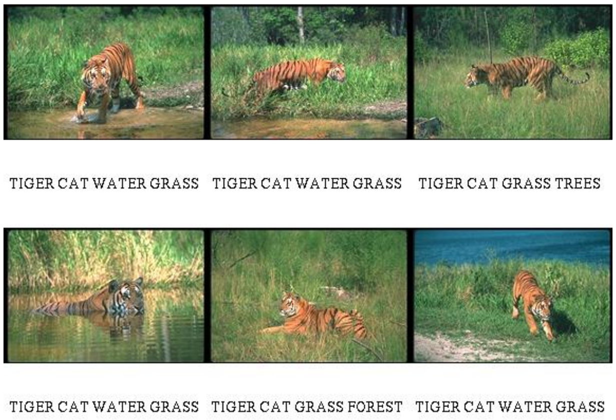


Figure 5. The results of the “river” and “tiger” query. The words “tiger” and “river” never appear together. Among tiger pictures, ones with the word “water” tend to get selected because “river” and “water” appear together in the training set. The same applies to segments are similar to ones which co-occur with the word “river” in training.

4. Pictures from Words and Words From Pictures

Given the above search strategy, one can build an application which takes text selected from a document, and suggests images to go with the text. This “auto-illustrate” application is essentially a process of linking pictures to words. However, it should be clear by symmetry that we can just as easily go the other way, and link words to pictures. This “auto-annotate” process is very interesting for a number of reasons. First, given an image, if we can produce reasonable words for it, then we can use existing text search infrastructure to broaden searches beyond the confines of our system. For example, consider image search by user sketch. If the sketch contains an orange ball in the upper right corner, and if that segment is associated with the words “sun” and "sunset", then text based query engines can be automatically consulted for interesting images based on these words, without having to be indexed by feature. The images found could then be analyzed in more detail.

The association of text with images is even more interesting from a computer vision perspective because it is a form of minimally supervised learning of semantic labels for image features. Although extreme accuracy in this task is clearly wishful thinking, we argue that doing significantly better than chance is useful, and with care could be used to further boot-strap machine recognition. We describe one step in this direction in the following section.

To use the model to attach words to pictures, we compute the probability that an image emits a proposed word, given the observed segments, B:

$$\begin{aligned} P(w \mid B) &= \sum_c P(w \mid c,B)P(c \mid B) \\ &\propto \sum_c P(w \mid c,B)P(B \mid c)P(c) \\ &= \sum_c P(w \mid c,B)\prod_{b\in B} P(b \mid c)P(c) \\ &= \sum_c \left(\sum_l P(w \mid c,l)P(l \mid c,B) \right) \prod_{b\in B} \left(\sum_l P(b \mid l,c)P(l \mid c) \right) P(c) \end{aligned} \tag{3}$$

At a glance the term $P(l \mid c,B)$ is difficult to compute. Interestingly, substituting the average value, $P(l \mid c)$, only slightly degrades the answer, compared with using a more accurate estimate of $P(l \mid c,B)$. The more accurate estimate is obtained by essentially refitting the model for a few iterations using the observations B, but without updating node specific parameters.

Figure 6 shows some sample results. To further test how well the annotation procedure works, we use the model to predict image words based only on the segments, and then compare the predicted words to the keywords. We perform this test on the training data, and two different test sets. The first set is randomly selected from the held out set from the proposed training data. The proposed training data comes from a number of Corel CD's. For the second set we use images from other CD's.

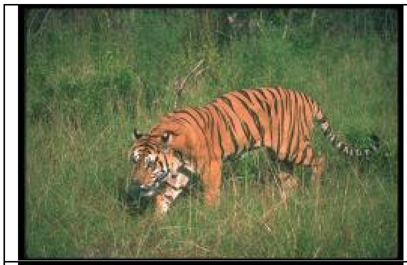
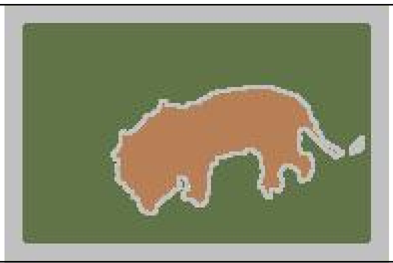
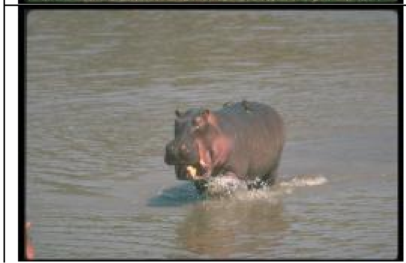
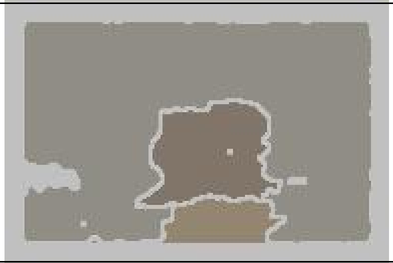
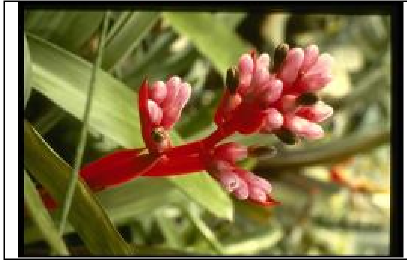
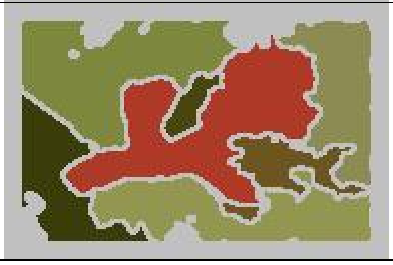
		Keywords GRASS TIGER CAT FOREST Predicted Words (rank order) tiger cat grass people water bengal buildings ocean forest reef
		Keywords HIPPO BULL mouth walk Predicted Words (rank order) water hippos rhino river grass reflection one-horned head plain sand
		Keywords FLOWER coralberry LEAVES PLANT Predicted Words (rank order) fish reef church wall people water landscape coral sand trees

Figure 6. Some annotation results showing the original image, the Blobworld segmentation, the Corel keywords, and the predicted words in rank order. The test images were not in the training set, but did come from the same set of CD's used for training. Keywords in upper-case are in the vocabulary. The examples chosen descend in accuracy from excellent (top) to poor (bottom). Flower examples similar to this one often do well, as the large, colorful segments provide a significant cue. However, in this case, the blobs show more affinity to the ones in non-flower clusters. The prediction of the word "reflection", which is correct by the image, but not by the available keywords, illustrates how this method can be used to finesse some very difficult problems. Reflection is proposed by the system because it occurs with words like "water" and segments like the ones in the image.

It is important to consider such a hold out set because the each Corel CD contains images with a single theme.

Given a test document and the associated words, we average the predicted probability of each observed keyword, which is the quantity the calculation above seeks to maximize. We scale these probabilities by the probability of occurrence, assuming uniform statistics, which specifies the extent to which the model predicts the word relative to complete ignorance. It is possible to do better than complete ignorance without considering image content by guessing words in proportion to their frequency of occurrence. Thus a more rigid test is how we do against this process, and thus we report a similar score, where each prediction was normalized by the overall probability of the word given the model. For a third measure we look at the 15 top ranked words, scoring each inversely to its rank. Thus, if a document word is predicted at rank 5, then it gives a score of 1/3. This measure is meant to capture the process of eyeballing the results. If a word does not occur in the top 15 or so, we loose interest in looking for it. Specifying a fixed cutoff also makes such a measure less sensitive to vocabulary size.

Table 1 summarizes the annotation result using the three scoring methods and the three held out sets. We average the results of 5 separate runs with different held out sets. Using the comparison of sampling from the word prior, we score 3.14 on the training data, 2.70 on non-training data from the same CD set as the training data, and 1.65 for test data taken from a completely different set of CD's.

4.1. Links to recognition

Given the task, we are not surprised that the results using images from different CD's are somewhat lower than the results with images from the same CD as the training set. The nature of the task being attempted is exemplified by predicting words like SKY in images of airplanes, after being trained on images of rhinos and tigers. Doing significantly better than chance on this general task indicates that the system has learnt some correspondences between image components and words. This means that the system has learnt, with minimal supervision, something about recognition. This in turn is interesting in the face of a key vision problem, namely how to approach general recognition. Systems have been built which are relatively effective at recognizing specific things, usually under specific circumstances. Doing well at these tasks has generally required a lot of high quality training data. We also use a lot of data, but it is of a much more available nature. There is no shortage of image data with text, especially if one includes video. It seems that the information required is contained in these data sets, and therefore looking at the recognition problem in this way should bear fruit.

	Rank Average	Relative to uniform prior	Relative to actual prior
Training data	0.64 (0.01)	4.92 (0.24)	3.14 (0.18)
Test data, from same CD set as training	0.60 (0.02)	4.53 (0.28)	2.70 (0.20)
Test on data from different CD set from training.	0.50 (0.02)	3.82 (0.13)	1.65 (0.12)

Table 1. Annotation results on the three kinds of test data, with three different scoring methods. Error estimates are given in parenthesis. The maximum possible score is 1.0. A score of 0.5 indicates that an average, predicted words are at rank 7—in other words, quite likely to be in the top 10.

5. Discussion

An important characteristic of the system is that we can measure its performance in a principled way by taking advantage of its predictive capabilities. In the previous section we used this to measure the annotation performance. The next step is to exploit this capability to improve the model. We can use our performance measure to study things like tree topology, data pre-processing, compromises made for scalability, training parameters, and word and segment vocabulary selection, as discussed further below.

One possible problem facing the application of this work to more general datasets, is that there may be words in the vocabulary which have no relevance to visual content. At best, such words cause random noise, but in doing so they take away probability from words which are more relevant. Some of these words could be culled if they are standard stop words, but this will leave dataset-dependent stop words. An example of such a word might be the name of the person who catalogued the image. Such words could be removed by observing that their emission probabilities are spread out over the nodes. Whether this automatic vocabulary reduction method makes sense depends on the nature of the data set. In the case of the Corel data set, the key word sets are both small and pertinent, and a quick investigation indicates that for this kind of data, global vocabulary selection is not appropriate.

A more serious difficulty with attaching words to pictures is that it relies on semantically meaningful segmentation. Such segmentations for general images will not become available in the near future. Our method works because given a healthy amount of data, some segmentations work well enough, and some words are sufficiently correlated with the segment features. Many words and segments, however, just add to the noise. For example, we expect that with good data, we should be able to learn a relationship between an orange ball in the upper half of the image and the word "sun". On the other hand, it would not surprise us if words like "omelet"—an actual Corel keyword— are never reliably predicted. The omelet is simply not obvious, even to a human, in any of the pictures, and is rarely, if ever, segmented as such. In general, some segments are more useful than others.

Given this analysis, it is compelling to use the model to identify which words and image segments can be related given the data and which cannot. It should then be possible to use this information to improve the training process. For example, if the word omelet cannot be predicted by image segments, then it may be best to remove it from further training. The same should apply to image segments. If a given segment cannot be predicted very well by the associated words, then perhaps it should be culled from the data set.

We stress that these strategies are possible because we have a way to measure performance. This is but one of the advantages of using a generative statistical model. By using such a model to integrate the semantic information available in text and image segments, we can support powerful browsing, creative searching, novel applications such as attaching words to images, and may provide some insight to the object recognition process.

6. References

[1] M. S. Lew, "Next-generation web searches for visual content," *IEEE Computer*, vol. 33, pp. 46-53, 2000.

[2] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments," *IEEE Transactions on Image Processing*, in press.

[3] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, pp. 22-32, 1995.

[4] J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei, "Content-based image indexing and searching using Daubechies' wavelets," *International Journal on Digital Libraries*, vol. 1, pp. 311-328, 1997.

[5] J. Z. Wang, "SIMPLIcity: A region-based image retrieval system for picture libraries and biomedical image databases," *Proc. ACM Multimedia Conference*, Los Angeles, 2000.

[6] J. Z. Wang, Semantics-sensitive integrated matching for picture libraries and biomedical image databases, Stanford University, Ph.D, 2000, available from http://www-db.stanford.edu/~wangz/project/thesis.

[7] C. Carson, S. Belonge, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using Expectation-Maximization and its application to image querying," in review.

[8] M. M. Fleck, D. A. Forsyth, and C. Bregler, "Finding Naked People," *Proc. 4th European Conference on Computer Vision*, 1996.

[9] M. Das, R. Manmatha, and E. M. Riseman, "Indexing Flower Patent Images using Domain Knowledge," *IEEE Intelligent Systems*, vol. 4, pp. 24-33, 1999.

[10] C. Frankel, M. J. Swain, and V. Athitsos, "Webseer: An Image Search Engine for the World Wide Web," U. Chicago TR-96-14,, 1996.

[11] M. L. Cascia, S. Sethi, and S. Sclaroff, "Combining Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1998.

[12] F. Chen, U. Gargi, L. Niles, and H. Schütze, "Multi-modal browsing of images in web documents," *Proc. SPIE Document Recognition and Retrieval*, 1999.

[13] P. G. B. Enser, "Query analysis in a visual information retrieval context," *Journal of Document and Text Management*, vol. 1, pp. 25-39, 1993.

[14] P. G. B. Enser, "Progress in documentation pictorial information retrieval," *Journal of Documentation*, vol. 51, pp. 126-170, 1995.

[15] L. H. Armitage and P. G. B. Enser, "Analysis of user need in image archives," *Journal of Information Science*, vol. 23, pp. 287-299, 1997.

[16] T. Hofmann, "Learning and representing topic. A hierarchical mixture model for word occurrence in document databases," *Proc. Workshop on learning from text and the web*, CMU, 1998.

[17] A. Berger and J. Lafferty, "Information retrieval as statistical translation," *Proc. ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[18] R. Srihari, Extracting Visual Information from Text: Using Captions to Label Human Faces in Newspaper Photographs, SUNY at Buffalo, Ph.D., 1991.

[19] V. Govindaraju, A Computational Theory for Locating Human Faces in Photographs, SUNY at Buffalo, Ph.D, 1992.

[20] R. K. Srihari, R. Chopra, D. Burhans, M. Venkataraman, and V. Govindaraju, "Use of Collateral Text in Image Interpretation," *Proc. ARPA Image Understanding Workshop*, Monterey, CA, 1994.

[21] R. K. Srihari and D. T. Burhans, "Visual Semantics: Extracting Visual Information from Text Accompanying Pictures," *Proc. AAAI '94*, Seattle, WA, 1994.

[22] R. Chopra and R. K. Srihari, "Control Structures for Incorporating Picture-Specific Context in Image Interpretation," *Proc. IJCAI '95*, Montreal, Canada, 1995.

[23] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," *Proc. Sixth International Conference on Computer Vision*, 1998.

[24] T. Hofmann and J. Puzicha, "Statistical models fro co-occurrence data," Massachusetts institute of technology, A.I. Memo 1635,, 1998.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1-38, 1977.