Image Classification by a Probabilistic Model Learned from Imperfect Training Data on the Web

Keiji Yanai

Department of Computer Science,
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi,
Tokyo, 182-8585 JAPAN

yanai@cs.uec.ac.jp

ABSTRACT

Current approaches to image classification require training images prepared by hand. In this paper, we describe experiments on image classification using images gathered from the Web automatically as training images. To gather images from the Web, we use the probabilistic method we proposed before. In the method, we build a generative model which is based on the Gaussian mixture model (GMM) from imperfect training images gathered from the Web in order to distinguish relevant images from irrelevant images. In this paper, we propose applying the model built during Web image gathering process to generic image classification task. In the experiments, we classified Corel images with the probabilistic model learned from Web images automatically.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Miscellaneous

General Terms

Algorithms, Experimentation, Measurement

Keywords

image annotation, probabilistic image selection, Web image mining

1. INTRODUCTION

Recently there has been much work related to semantic image classification [8, 13, 4] and annotation of words to images [1, 9]. In these studies, image sets gathered by hand or commercial image collections such as Corel image library were used as training images, since training images with keywords were required.

On the other hand, recently we are working on gathering images from the World Wide Web and applying them to generic object recognition tasks as visual knowledge instead of hand-made or commercial image collections [13]. Learning of image concepts from the Web is being paid attention as a new framework to avoid human labor for making training image sets [11, 5, 6]. Web images are as diverse as real world scenes, since Web images are taken by a large number of people for various kinds of purpose. It can be expected that diverse training images enable us to classify/recognize diverse real world images. We believe that use of image data on the Web, namely visual knowledge on the Web, is promising and important for resolving real world image recognition.

In this paper, we describe experiments on image classification using images gathered from the Web automatically as training images. To gather images from the Web, we use the probabilistic method [15] which we proposed recently. In the method, we build a GMM-based generative model from imperfect Web image sets to distinguish relevant images from irrelevant images. Then, in this paper we propose applying the model built during Web image gathering process to generic image classification task.

So far much work regarding Web image search has been proposed as well as commercial services. However, most of them focused on only "search". On the other hand, our purpose of use of images on the Web is "Web image mining"[14, 13], which means searching the Web for images and then using them as visual knowledge for some applications.

Regarding text data, there are many studies about how to gather data from the Web and use it as "knowledge" effectively. While such Web text mining is an active research area, mining image data on the Web poses additional challenges and has seen less research activity. The problem with mining images for knowledge is that it is not known how to reliably automatically determine semantics from image data. This has been refereed to as the semantic gap. To

MDM/KDD'06, August 20, 2006, Philadelphia, Pennsylvania, USA.
Copyright 2006 ACM 1-59593-446-4 ...$5.00.

solve it, it is indispensable to introduce sophisticated image recognition methods into Web mining regarding images.

In [14, 13], we proposed gathering a training data set for generic image recognition from the Web automatically, and have revealed that we could use Web images as "visual knowledge". In this paper, we extend this idea with a sophisticated probabilistic method.

The advantage of our method for Web image gathering [15] is that we can obtain many images relevant to a certain keyword "X" by just providing the keyword "X" without any supervision or any feedback. The combination of HTML analysis and probabilistic image selection enable it. We proposed starting with images evaluated as highly relevant ones by analyzing associated HTML texts as training images. In our previous work [12], we revealed that images whose file name, ALT tag or link word includes a certain keyword "X" are relevant to the keyword "X" with around 75% precision on average. Although the images include 25% irrelevant images, and many of the remaining 75% are not a desired canonical example, they provide an adequate starting point for our approach. We then build a model of a visual concept associated to the keyword "X". We use a generative model based on the Gaussian mixture model to represent "X" model, and estimate the model with the EM algorithm. Next, we "recognize" images evaluated as highly relevant by analyzing associated HTML texts with the model, and select "X" images from them. By repeating this image selection and model estimation for several times, we can refine the "X" model and finally obtain "X" images with the high accuracy.

In our previous work [15], we used a probabilistic model only to select "X" images from Web images. In this paper, we then apply the models built during Web image gathering process to generic image classification task which is not limited to Web images. To investigate these ideas, we built models regarding six keywords through the probabilistic Web image gathering processes, and applied them to image classification of 6 kinds of Corel image sets.

The rest of the paper is as follows. In Section 2, we describe the method to build models and to apply them to image classification task. In Section 3, we explain the experimental results of 6-category image classification. In Section 4, we conclude this paper.

2. METHOD

We apply the probabilistic method employed in our probabilistic Web image gathering system [15] to generic image classification tasks. Therefore, the method we propose in this paper is an extension of the Web image selection method we employed in our previous work.

In our Web image gathering scheme, we gather several hundreds of Web images relevant to a given concept. At first, we provide keywords which represent the visual concept of images we like to obtain. For example, "mountain", "beach" and "sunset". Using Web image/text search engines, we gathered "raw" images related to the given concept from the World Wide Web. The "raw" image always includes many irrelevant images, the ratio of which is 50% or more on average.

Next, we carry out HTML analysis and select "A-ranked" images which are very likely to be relevant. Here, we regard images whose file name, ALT tag or link word includes a certain keyword "X" as "A-ranked" ones [12]. A-ranked images are relevant to the keyword "X" with around 75% precision on average. Using them as training images, we employ a probabilistic method to select only relevant images from all the A-ranked images 25% of which are irrelevant. This is why our method is unsupervised although we use a probabilistic learning method which requires training images.

In our probabilistic learning framework, we allow training data to include some irrelevant data and we can remove them by repeating both estimation of a model and selection of relevant regions of images from all the regions of raw images. We use a generative model based on the Gaussian mixture model to represent models associated to keywords, and estimate models with the EM algorithm. After estimating the model, we "recognize" relevant region out of all regions in all the A-ranked Web images with the model. We repeat this model estimation and region selection. After the second iteration, we use regions selected in the previous iteration as training data for estimating a model.

After obtaining a model, we apply the learned model to image classification. We classify a test image into one of the given class so that the probability of "X" given the image is the largest.

2.1 Segmentation and Image Feature Extraction

To extract image features from each region, we carry out the region segmentation in advance. In the experiments, we used JSEG [2]. After segmentation, we extract image features from each region whose size is larger than a certain threshold. As image features, we prepare three kinds of features: color, texture and shape features, which include the average RGB value and its variance, the average response to the difference of 4 different combination of 2 Gaussian filters, region size, location, the first moment and the area divided by the square of the outer boundary length. An image feature vector we use in this paper is totally 24-dimension. We need to do such pre-processing for all the test images to be classified as well as all of the Web images.

2.2 Training a Model from Imperfect Training Data

As a method to select images, we adopt a probabilistic method with a Gaussian mixture model. This approach is based on the method for learning to label image regions from images with associated text without the correspondence between words and images regions [3, 1]. That method uses a mixture of multi-modal components, each combining a

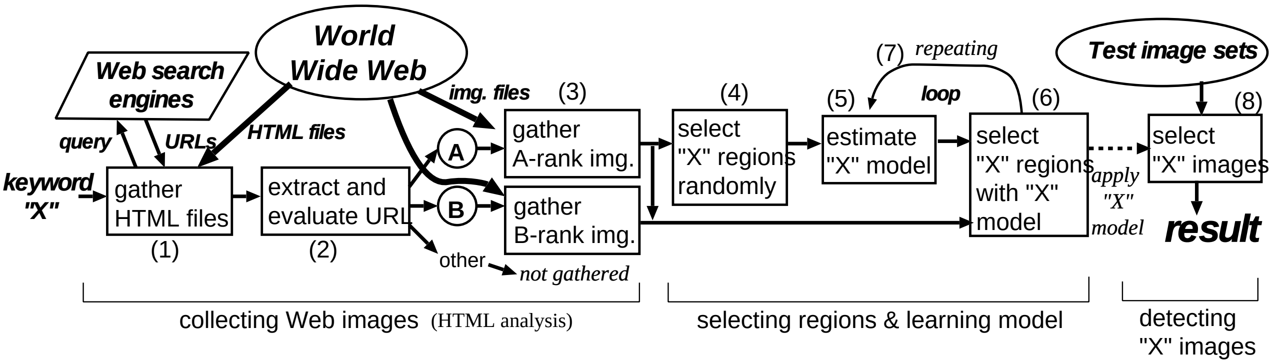

Figure 1: Processing flow of learning from Web and classifying "X" images.

multinomial for words and a Gaussian over image features. Here, we simplify things a bit, and build models of the distribution of image features for a given concept for regions which are obtained by a region segmentation algorithm.

To get a model of regions associated to a certain concept, we need training images. As mentioned before, our basic policy is no human intervention, so that we use images which are highly evaluated by the HTML analysis as training images. Most of such images are relevant ones, but they always include outliers due to no supervision. Moreover, in general, images usually include backgrounds as well as objects associated with the given concept. Therefore, we need to eliminate outlier images and regions unrelated to the concept such as backgrounds, and pick up only the regions strongly associated with the concept in order to make a model correctly. We use only the regions expected to be highly related to the concept to estimate a model. In our method, we need negative training images corresponding to "non-X" images in addition to positive training images. We prepare about one thousand images by fetching them from the Web randomly as negative training images in advance.

Our method to find regions related to a certain concept is an iterative algorithm similar to the expectation maximization (EM) algorithm applied to missing value problems. Initially, we do not know which region is associated with a concept "X", since an image with an "X" label just means the image contain "X" regions. In fact, with the images gathered from the Web, even an image with an "X" label sometimes contains no "X" regions at all. So at first we have to find regions which are likely associated with "X". To find "X" regions, we also need a model for "X" regions. Here we adopt a probabilistic generative model, namely a mixture of Gaussian, fitted using the EM algorithm.

In short, we need to know a model for "X" regions and which regions are associated with "X" simultaneously. However, each one depends on each other, so we proceed iteratively. Once we know which regions corresponds to "X", we can regard images containing "X" regions as "X" images, and therefore we can compute the probability of an "X" image for each image.

2.3 Detail of the Computation

To obtain $P(X|r_i)$, which represents the probability of how much the region is associated with the concept "X", and some parameters of the Gaussian mixture model, which represents a generative model of "X" regions, at the same time, we use an iterative algorithm.

At the initial iteration, we regard A-ranked images which are expected to be highly relevant to the concept "X" by HTML analysis as positive training images, and prepare negative training images by gathering images from the Web in advance. To gather negative training images, we provided Google Image Search with randomly selected 200 adjective keywords which have no relation to noun concepts, and collected 4000 negative training images.

Next, we select n "X" regions randomly from A-ranked images, and select n "non-X" regions randomly from regions which come from negative training images, respectively. In the experiment, we set n as 1000.

Taking positive and negative regions together, we apply the EM algorithm, which is a kind of a probabilistic clustering algorithm, to $2n$ image feature vectors of the regions selected from positive and negative initial training images, and obtain the Gaussian mixture model.

To select positive components and negative components from all components of the mixture model, we compute $P(c_j|X)$ which represents the ratio that the j-th component of the mixture model, c_j, contributes to the concept "X" within the obtained GMM, according to the following formula:

$$\begin{aligned} P(c_j|X) &= \sum_{i=1}^{n} P(c_j|r_i^X)P(r_i^X) \\ &= \alpha \sum_{i=1}^{n} P(r_i^X|c_j)P(c_j)P(r_i^X) \\ &= \alpha \sum_{i=1}^{n} w_j f(r_i^X;\mu_j,\Sigma_j)S_i/\Sigma_{i=1}^{n_X} S_i \end{aligned} \tag{1}$$

where r_i^X is the i-th "X" region, n_X is the number of positive regions, α is a constant for the normalization, and w_j is the weight of j-th Gaussian on the condition of $\sum_{j=0}^{m} w_j = 1$. $f(r_i^X; \mu_j, \Sigma_j)$ is the Gaussian distribution where μ_j and Σ_j are the mean vector and the covariant matrix of the j-th component. This function is represented by the following equation:

$$f(r_i^X; \mu_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_j|}} \exp^{-\frac{1}{2}(r_i^X - \mu_j)^T \Sigma_j^{-1} (r_i^X - \mu_j)} \quad (2)$$

where N is the dimension of the feature vectors. As the same way, we also compute $P(c_j|nonX)$.

Next, we compute p_j^X for all components j as follows:

$$p_j^X = \frac{P(c_j|X)}{P(c_j|X) + P(c_j|nonX)} \quad (3)$$

We select components where $p_j^X > th_1$ as positive components and components where $1 - p_j^X > th_1$ as negative components. Positive components and negative components means Gaussian components associated with the concept "X" and Gaussian components strictly not to associated with "X", respectively. The key point in this component selection process is that mixing positive samples and negative samples together before applying the EM, and throwing away neutral components which belongs to neither positive nor negative components, since neutral components are expected to be associated with image features included in both positive and negative samples and to be useless for discrimination between "X" and "non-X". This is different from other work (e.g.[10]) which estimates two GMMs separately with EM to model positive and negative image concepts.

We regard the mixture of only positive components as an "X" model and the mixture of only negative components as a "non-X" model as. With these models of "X" and "non-X", we can compute $P(X|r_i)$ and $P(nonX|r_i)$ for all the regions extracted from A-ranked images First, we compute $P(r_i|X)$ which is the output of the model of "X' and $P(r_i|nonX)$ which is the output of the model of "non-X" for each region r_i:

$$P(r_i|X) = \sum_{j=1}^{m_X} w_j^X f(r_i^X; \mu_j^X, \Sigma_j^X) \quad (4)$$

$$P(r_i|nonX) = \sum_{j=1}^{m_{nonX}} w_j^{nonX} f(r_i^{nonX}; \mu_j^{nonX}, \Sigma_j^{nonX}) \quad (5)$$

where m_X is the number of positive components, w_j^X is the weight of j-th positive Gaussian on the condition of $\sum_{j=0}^{m_X} w_j^X = 1$, and $f(r_i^X; \mu_j^X, \Sigma_j^X)$ is the Gaussian distribution where μ_j^X and Σ_j^X are the mean vector and the covariant matrix of j-th positive component.

Finally, we obtain $P(X|r_i)$ and $P(nonX|r_i)$ with the Bayesian theorem as follows:

$$P(X|r_i) = \frac{P(r_i|X)P(X)}{P(r_i|X)P(X) + P(r_i|nonX)P(nonX)} \quad (6)$$

For the next iteration, we select the top n regions regarding $P(X|r_i)$ as "X" regions and the top $\frac{2}{3}n$ regions regarding $P(nonX|r_i)$ as "non-X" regions. In addition, we add $\frac{1}{3}n$ regions randomly selected from negative images gathered from the Web in advance to the "non-X" regions. We repeat building models and selecting images for several times. For every iteration, we use newly selected n positive regions and n negative regions as training data.

The detail on the computation of probability described above is basically the same as the method we proposed in [15].

2.4 Applying Trained Models to Classify Test Images

To detect images relevant to a certain concept, we apply the trained models in the same way as region selection during the training stage, and estimate the probability of "X" ($P(X|r_j)$) for all the regions extracted from all the test images. Finally, we regard the mean of the probability of "X" of top T regions within each image as the probability of "X" ($P(X|I_i)$) for each image. This estimation of $P(X|I_i)$ is based on the heuristic that an image having regions whose $P(X|r_j)$ are high can be regarded as an "X" image. Since images usually includes backgrounds as well as target objects, background regions or unrelated regions should be ignored for estimating $P(X|I_i)$. Therefore, we use not all regions but only several important regions to compute the probability of "X" of images. In the experiment, we set T as 2. We compute $P(X|I_i)$ for each test image regarding several kinds of "X" keywords. Finally, we classify a test image into one of the given classes so that $P(X|I_i)$ is the largest.

2.5 Algorithm

To summarize our method we described above, the algorithm is as follows:

(1) Carry out region segmentation for all the images and extract image features from each region of each image.

(2) At the first iteration, regard A-ranked images as positive training images which are associated with the concept "X" and images gathered from the Web with non-noun keywords in advance as negative training images.

(3) Select n "X" regions randomly from positive images, and select n "non-X" regions randomly from negative images, respectively (Figure 1 (4)).

(4) Applying the EM algorithm to the image features of regions which are selected as both positive and negative regions, compute the Gaussian mixture model for the distribution of both "X" and "non-X" (Figure 1 (5)).

(5) Find the components of the Gaussian mixture which contributes "X" regions or "non-X" regions greatly. They are regarded as "X" components or "non-X" components, and the rest are ignored. The mixture of only "X" regions is a model of "X" regions, and the mixture

of only “non-X” is a model of “non-X” regions.

(6) Based on the mixture of “X” components and the mixture of “non-X” components, compute $P(X|r_j)$ and $P(nonX|r_j)$ for all the regions which come from “X” images, where r_j is the j-th region.

(7) Select the top n regions in terms of $P(X|r_j)$ as new positive regions and the top $\frac{2}{3}n$ regions in terms of $P(non\text{-}X|r_j)$ as new negative regions. Add $\frac{1}{3}n$ regions randomly selected from the negative training images to new negative regions.

(8) Repeat from (4) to (7) with newly selected positive and negative regions (Figure 1 (7)).

(9) After repeating several times, apply the trained model to test data sets. Calculate $P(X|r_j)$ for all the regions extracted from all test images, and then obtain $P(X|I_i)$ for all the test images (Figure 1 (8)).

3. EXPERIMENTAL RESULTS

In the experiment, we used six keywords, “apple”, “beach”, “flower”, “lion”, “sunset” and “waterfall”. These keywords was imported from the experiments in [15]. We gathered 1204 A-ranked images from the Web for one keyword on average.

We prepared 50 test images for each keyword by selecting images from the Corel Image Gallery based on their attached keywords, and we also prepared 50 test images consisting of Web images which were not included in training data sets. We carried out four kinds of experiments. First, we used all of the A-ranked images as training images to build models. Next we used only relevant images selected by hand from the A-ranked images as training ones. Note that our final goal is to develop methods which can learn from raw Web images appropriately, although we made the experiment with all relevant training sets. We used Corel images as test data sets in the first two experiments. Table 3 s hows all the results of the six-class image classification.

The second column shows the precision of the raw A-ranked Web images. They were from 67.1% to 87.8% and the average precision was 76.2% This result followed our observation in [12].

The third to fifth columns in Table 1 show the precision, the recall and the F-measure of the result of image classification in case of using the raw A-ranked images as training images, where the F-measure is the harmonic mean of the precision and the recall. The average F-measure was 36.7%. This was not as good as we expected. There are several reasons. The biggest one is that some of Corel images we used as test images were taken in the situation which was not common but too special to be classified. For example, most of the Corel images on “flower” (the right side of Figure 2) are close-up, while most of the Web “flower” images (the left side of Figure 2) include many flowers in fields in one image. In case of “apple” shown in Figure 3, both the Web images and the Corel images are too various to recognize. Including too many types of “apple” images caused the low classification rate. On the other hand, in case of “sunset” and “lion” both Corel and Web images include many similar images (Figure 4 and 5).

Table 1: Results of the six-class image classification. This table includes the precision of the raw A-ranked Web images, the precision, the recall and the F-measure of the two kinds of results of image classification experiments in case of using raw A-ranked images and in case of using only relevant images.

class	training	raw A-ranked			only relevant		
	image	Corel image			Corel image		
	precision	pre.	rec.	F	pre.	rec.	F
apple	66.8	36.4	5.7	**9.9**	27.1	18.6	**22.0**
beach	68.8	29.8	25.8	**27.6**	60.9	72.2	**66.0**
flower	72.2	39.5	18.1	**24.8**	38.3	19.1	**25.5**
lion	87.5	55.1	27.3	**36.5**	100.0	67.7	**80.7**
sunset	67.1	34.1	79.4	**47.7**	42.3	84.5	**56.4**
waterfall	70.9	42.2	49.5	**45.6**	36.2	21.2	**26.7**
AVERAGE	76.2	39.5	34.3	**36.7**	50.8	47.2	**48.9**

Table 2: Results of the six-class image classification in case of using other Web images as test data sets.

class	training	raw A-ranked			only relevant		
	image	Web image			Web image		
	precision	pre.	rec.	F	pre.	rec.	F
apple	66.8	50.0	2.0	**3.8**	31.7	37.3	**34.2**
beach	68.8	33.3	60.8	**43.1**	41.5	52.9	**46.5**
flower	72.2	31.9	29.4	**30.6**	35.0	41.2	**37.8**
lion	87.5	61.8	41.2	**49.4**	100.0	72.5	**84.1**
sunset	67.1	51.3	76.5	**61.4**	54.7	56.9	**55.8**
waterfall	70.9	39.2	39.2	**39.2**	50.0	27.5	**35.4**
AVERAGE	76.2	44.6	41.5	**43.0**	52.2	48.0	**50.0**

The sixth to eighth columns in Table 1 show the precision, the recall and the F-measure of the result of image classification in case of using the only relevant images selected out of the raw A-ranked images by hand. For most of the keywords, the F-measure increased compared to the case of the raw images. This is a reasonable result, since the precision of training images was 100% in this case.

As additional experiments, we made two experiments using hand-selected Web image sets which includes only relevant images as test data sets. The difference between two experiments are whether training sets include irrelevant images. This is the same way as the first two experiments. The results are shown in Table 2. The results are improved slightly, since the nature of Web test images are similar to the nature of training data sets obtained from the Web.

As the third experiment, we added four classes (“baby”, “mountain”, “Chinese noodle” and “laptop PC”) which are not included in the Corel library except “mountain”, and made ten-class classification experiments. These ten classes exactly correspond to the ten keywords we used in the experiments of Web image gathering in [15]. Since the Corel image library has no images corresponding to some of the added classes, we made experiments with only Web images as test data. In addition, we tried RANSAC (RANdom SAmpling Cosensus)[7] during the training process to eliminate the effect of irrelevant training images. According to the RANSAC method, we split the training data into two groups, one of them is used for building the model, and the other is used for evaluation the built model. We repeated this building and evaluating of the models with 30 times. Finally, we selected the best model the evaluation of which was the highest among the 30 models.

The experimental results are shown in Table 3. The results are slightly improved due to RANSAC. The differences are 3.7% in case of raw images and 0.6% in case of all relevant images as training data. In the latter case, irrelevant images are already removed, so the amount of improvement due to RANSAC is very small.

Table 3: Results of the ten-class classification with and without RANSAC.

method	raw A-ranked			only relevant		
	pre.	rec.	F	pre.	rec.	F
normal	25.7	23.4	**24.5**	28.5	37.7	**32.5**
RANSAC	26.9	29.7	**28.2**	29.0	38.5	**33.1**

4. CONCLUSIONS

In this paper, we describe experiments on image classification using images gathered from the Web automatically as training images. We proposed the method to learn model from imperfect training image data by modifying the model employed in the Web image gathering process and to apply it to generic image classification task. Such methods are very important for learning from the Web, since data on the Web always includes noise. The experimental results indicated that the method still needed to be improved, but this framework is one of the promising directions to realize generic image classification/recognition, which is the final goal of our project.

Since this research project is still in the early stage, we have a lot of what to do. For example, we plan to improve the probabilistic method and image features. In this paper, we applied the model of Web image gathering to generic image classification as it was. We are going to modify it for multi-class classification. Recently the use of small parts of images as image features is being paid much attention and enables relatively good performance on image detection and classification [4, 6]. We also plan to import this idea into our method.

5. REFERENCES

[1] K. Barnard, P. Duygulu, N. d. Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[2] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, 2001.

[3] P. Duygulu, K. Barnard, J. d. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *The Seventh European Conference on Computer Vision*, pages IV:97–112, 2002.

[4] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *Proc. of IEEE CVPR Workshop of Generative Model Based Vision*, 2004.

[5] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. of IEEE International Conference on Computer Vision*, pages 1816–1823, 2005.

[6] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. of European Conference on Computer Vision*, pages 242–255, 2004.

[7] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[8] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1–14, 2003.

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

[10] M. R. Naphade, S. Basu, J. R. Smith, C. Y. Lin, and B. Tseng. Modeling semantic concepts to support query by keywords in video. In *Proc. of IEEE Intl. Conference on Image Processing*, pages I:145–148, 2002.

[11] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1483–1490, 2006.

[12] K. Yanai. Image collector: An image-gathering system from the World-Wide Web employing keyword-based search engines. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 704–707, 2001.

[13] K. Yanai. Generic image classification using visual knowledge on the web. In *Proc. of ACM International*

Conference Multimedia, pages 67–76, 2003.

[14] K. Yanai. Web image mining toward generic image recognition. In *Proc. of the Twelfth International World Wide Web Conference*, 2003.

[15] K. Yanai and K. Barnard. Probabilistic web image gathering. In *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 57–64, 2005.

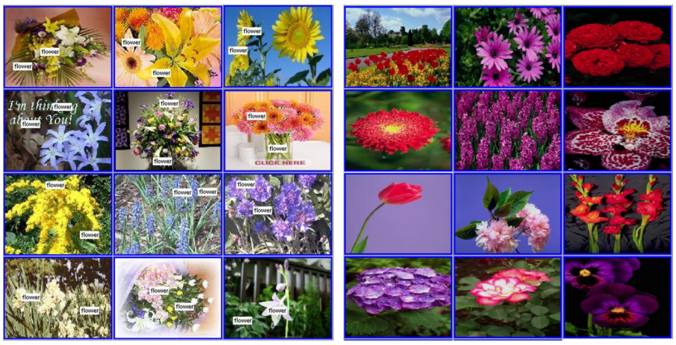

Figure 2: "Flower" Web images (left) and Corel images (right).

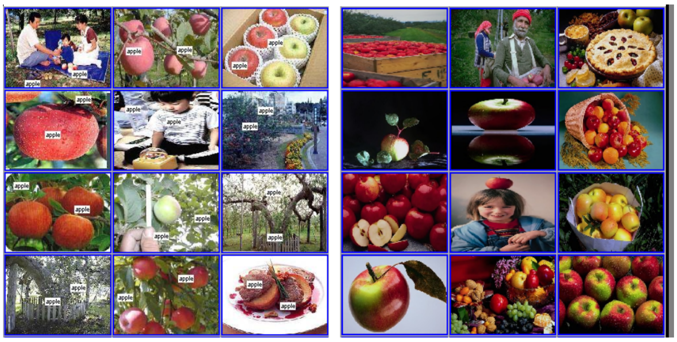

Figure 3: "Apple" Web images (left) and Corel images (right).

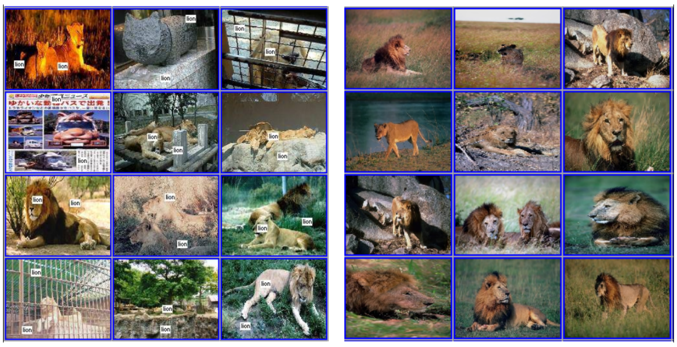

Figure 4: "Lion" images. Web images are on the left, and Corel images are on the right.

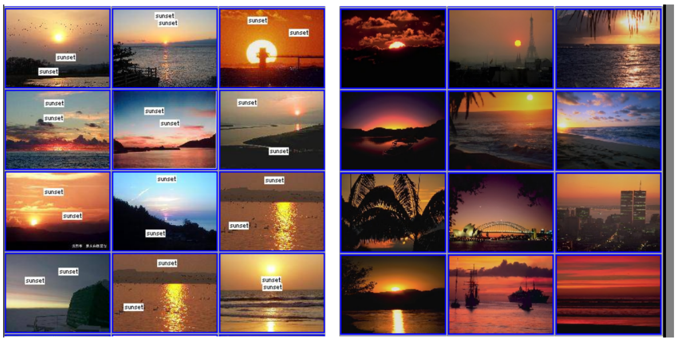

Figure 5: "Sunset" Web images (left) and Corel images (right).