Generic Image Classification Using Visual Knowledge on the Web

Keiji Yanai

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

yanai@cs.uec.ac.jp

ABSTRACT

In this paper, we describe a generic image classification system with an automatic knowledge acquisition mechanism from the World Wide Web. Due to the recent spread of digital imaging devices, the demand for image recognition of various kinds of real world scenes becomes greater. For realizing it, visual knowledge on various kinds of scenes is required. Then, we propose gathering visual knowledge on real world scenes for generic image classification from the World Wide Web. Our system gathers a large number of images from the Web automatically and makes use of them as training images for generic image classification. It consists of three modules, which are an image-gathering module, an image-learning module and an image classification module. The image-gathering module gathers images related to given class keywords from the Web automatically. The learning module extracts image features from gathered images and associates them with each class. The image classification module classifies an unknown image into one of the classes corresponding to the class keywords by using the association between image features and classes. In the experiments, the system demonstrated potential for generic image classification/recognition using images gathered from the World Wide Web automatically as training images.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Experimentation

Keywords

Web image mining, image gathering, image classification

1. INTRODUCTION

Due to the recent spread of digital imaging devices such as a digital camera, a digital video recorder and an image scanner, we can easily obtain digital images of various kinds of real world scenes, so that the demand for generic image recognition of various kinds of real world images becomes greater. It is, however, difficult to apply conventional image recognition methods to such generic recognition, because most of their applicable targets are restricted. Therefore, at present, it is difficult to deal with semantics of images of real world scene automatically. Henceforth, semantic processing for images such as automatic attaching keywords to images, classification and search in terms of semantic content of images is desired.

So far, automatic attaching keywords [1, 5, 8, 17] and semantic search [2] for an image database have been proposed. In these works, since training images with correct keywords were required, commercial image collections were used as training images, for example, Corel Image Library. However, most of images in commercial image collections are well-arranged images taken by professional photographers, and many similar images are included in them. They are different from images of real world scenes taken by the people with commodity digital cameras.

In this paper, we propose gathering visual knowledge on real world scenes for generic image classification from the World Wide Web. In other words, this research is "*Web Image Mining*" for generic image classification. To say it concretely, our system utilizes images gathered automatically from the World Wide Web as training images for generic image classification instead of commercial image collections. We can easily extract keywords related to an image on the Web (Web image) from the HTML file linking to it, so that we can regard an Web image as an image with related keywords. 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 is expected that diverse training images enable us to classify diverse real world images.

The main targets of the conventional works on Web mining are numeric data and text data. However, there are a large number of multimedia data such as images, movies and sounds on the Web. We think that use of multimedia data on the Web, namely visual knowledge on the Web, is promising and important for resolving real world image recognition/classification.

The processing in our system consists of three steps. In the gathering stage, the system gathers images related to given class keywords from the Web automatically. In the

MM'03, November 2–8, 2003, Berkeley, California, USA.

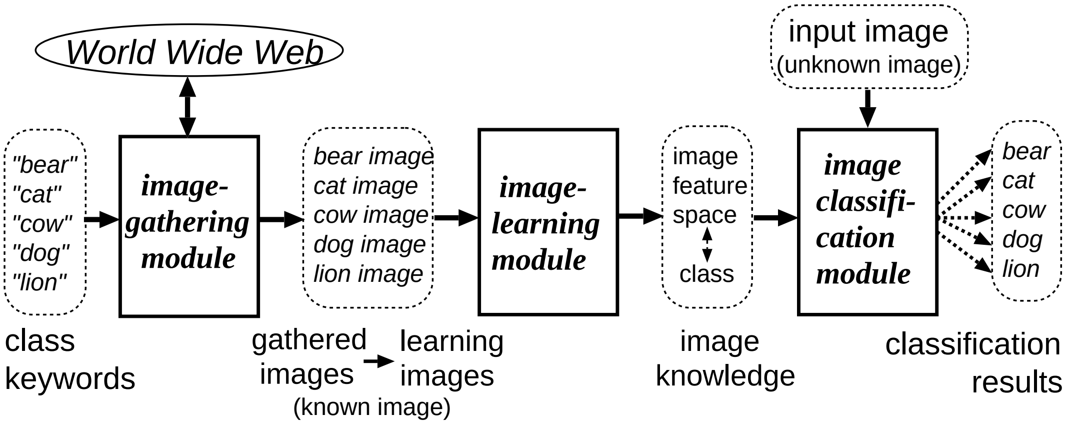

Figure 1: Proposed system, which is constructed as an integrated system of an image-gathering module, an image-learning module and an image classification module.

learning stage, it extracts image features from gathered images and associates them with each class. In the classification stage, the system classifies a unknown image into one of the classes corresponding to the class keywords by using the association between image features and classes. The system is constructed by integrating three modules, which are an image-gathering module, an image-learning module, and an image classification module (Figure 1).

In this paper, we describe methods of image-gathering from the World Wide Web, learning from gathered images and classification of an unknown input image. Next, we describe experimental results and conclusions.

2. IMAGE GATHERING

First of all, we have to prepare several kinds of class keywords, which represent the classes into which unknown images are classified. For example, cow, dog and cat. For each class keyword, we gather related images from the Web as training images. For gathering images from the Web, we use the Image Collector [19, 20] as an image gathering module.

At present, some commercial image search engines on the Web such as Google Image Search, Ditto and AltaVista Image Search are available. Their preciseness of search results is, however, not good since they employs only keyword-based search. Then, some integrated search engines employing both keyword-based search and content-based image retrieval have been proposed. WebSeer [6], WebSEEk [16] and Image Rover [15] have been reported so far. These systems search for images based on the query keywords, and then a user selects query images from search results. After this selection by the user, the systems search for images that are similar to the query images based on image features. These three systems carry out their search in an interactive manner.

The objective of our image-gathering module is absolutely different from ones of these conventional Web image search engines including commercial Web image search engines. Their objective is searching for highly relevant images, the number of which is relatively small. So they have adopted interactive search. Unlike these system, our image gathering module requires gathering a large number of relevant images for the image learning module automatically, so that we adopt non-interactive search without user's intervention during the gathering process. For this reason, it is no problem that the processing time of the gathering modules gets very long.

Furthermore, the three research systems quoted above require crawling over the Web in advance for gathering Web images and making large indices of images on the Web. Hence, they require a large-scale web-crawling mechanism for the whole Web and continuous web-crawling to keep their indices up-to-date for practical use. However, they limited crawled Web sites in their experiments, and did not make large indices covering the whole Web. This shows difficulty to make these system more practical like Google image search. In contrast to those systems, due to exploiting of existing keyword-based search engines and on-demand image-gathering, our system does not require a large-scale web-crawling mechanism and making a large index in advance. Therefore, our system can be used for practical use unlike other works on Web image search.

2.1 Overview of Image Gathering Module

The image-gathering module gathers images from the Web related to class keywords. Note that we do not call this module the image "search" module but the image "gathering" module, since its objective is not to search for a few highly relevant images but to gather a large number of relevant images.

Since an image on the Web is usually embedded in an HTML document that explains it, the module exploits some existing commercial text-based Web search engines and gathers URLs (Universal Resource Locator) of HTML documents related to the class keywords. In the next step, using those gathered URLs, the module fetches HTML documents from the Web, analyzes them and evaluates the intensity of relation between the keywords and images embedded in HTML documents. If it is judged that images are related to keywords, the image files are fetched from the Web. According to the intensity of relation to the keywords, we divide fetched images into two groups: images in group A having stronger relation to the keywords, which can be decided as being relevant ones only by HTML document analysis, and others in group B. For all gathered images, image features are computed.

In content-based image retrieval (CBIR), a user provides

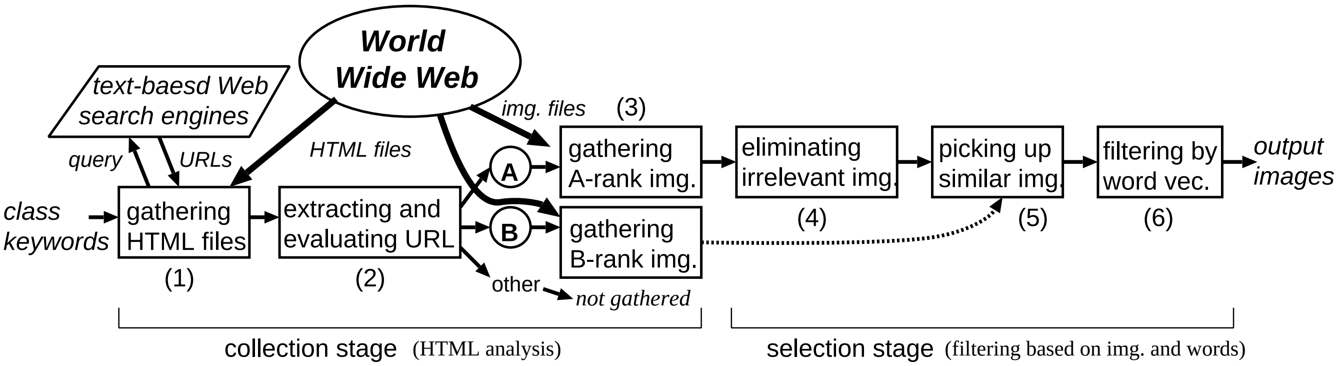

Figure 2: Processing flow of image-gathering from the Web, which consists of the collection stage and the selection stage.

query images or sketches to the system, because it searches for images based on the similarity of image features between query images and images in an image database. In our image-gathering module, instead of providing query images or sketches, a user only needs to provide query (class) keywords to the module. Then, we select images strongly related to the keywords as group A images, remove noise images from them, and regard them as query images. Removing noise images is carried out by eliminating images which belong to relatively small clusters in the result of image-feature-based clustering for group A images. Images which are not eliminated are regarded as appropriate images to the class keywords, and we store them as output images. Our preference of larger clusters to smaller ones is based on the following heuristic observation: an image that has many similar images is usually more suitable to an image represented by keywords than one that has only a few similar images. Next, we select images that are similar to the query images from group B in the same way as CBIR, and add them to output images. Figure 2 describes this processing flow. The detail of the image gathering module and the experimental results of the gathering module solely are described in [19] and [20].

The processing of the image-gathering module consists of collection and selection stages.

2.2 Collection Stage

In the collection stage, the system obtains URLs using some commercial web search engines, and by using those URLs, it gathers images from the web. The detail of the algorithm is as follows.

1. A user provides the system with two kinds of query keywords. One is a main keyword that best represents an image, and the other is an optional subsidiary keyword. For example, when we gather “lion” images, we use “lion” as a main keyword and “animal” as a subsidiary keyword. The reason why we divide keywords into two kinds is that subsidiary keywords are replaced with newly generated keywords in the query expansion processing described later.
2. The system sends the main and subsidiary keywords as queries to the commercial search engines and obtains the URLs of the HTML documents related to the keywords (Figure 2 (1)).
3. It fetches the HTML documents indicated by the URLs.
4. It analyzes the HTML documents, and extracts the URLs of images embedded in the HTML documents with image-embedding tags (“IMG SRC” and “A HREF”) (Figure 2 (2)). For each of those images, the system calculates a score that represents the intensity of the relation between the image and the query keywords. Note that an evaluation method used here is based on simple HTML tag analysis, which is similar to a method commonly used in Web image search engines [10, 15, 16]. The score is calculated by checking the following conditions.

   Condition 1: Every time one of the following conditions is satisfied, 3 points are added to the score.
   - If the image is embedded by the “SRC IMG” tag, the “ALT” field of the “SRC IMG” includes the keywords.
   - If the image is linked by the “A HREF” tag directly, the words between the “A HREF” and the “/A” include the keywords.
   - The name of the image file includes the keywords.

   Condition 2: Every time one of the following conditions is satisfied, 1 point is added.
   - The “TITLE” tag includes the keywords.
   - The “H1, ..,H6” tags include the keywords, assuming these tags are located just before the image-embedding one.
   - The “TD” tag including the image-embedding tag includes the keywords.
   - Ten words just before the image-embedding tag or ten words after it include the keywords.

   If the final score of an image is higher than 3, the image is classified into group A. If it is higher than 1, the image is classified into group B. URLs of the images with no score are ignored. The system only fetches files whose images belong to either group A or B (Figure 2 (3)). The reason why we made Condition 1 highly-evaluated is that our preliminary experiments turned out that an ALT field, link words and a file name had high tendency to include keywords related to the image.

   If the size of a fetched image-file is larger than a certain predetermined size, the image is handed to the selection stage.
5. In case the HTML document does not include image-embedding-tags at all, the system fetches and analyzes other HTML documents linked from it in the same manner described above, if it includes a link tag (“A HREF”) which indicates URL of HTML documents on the same web site. In the current implementation, we limit links

followed to depths of only one step, since in general the content of an Web page gets farther from the given keywords as following links deeply.

2.3 Selection Stage

In the selection stage, the system selects more appropriate images for the query keywords out of the ones gathered in the collection stage. The selection is based on the image features described below.

1. The system first makes image feature vectors for all the collected images. In the current implementation, our system uses a 6 × 6 × 6 color histogram in the Lu^*v^* color space.
2. For images in group A, the distance (dissimilarity) between two images is calculated based on the quadratic form distance [7], which takes account of the proximity between bins of the histogram in the color space.
3. Based on the distance between images, images in group A are grouped by the hierarchical cluster analysis method. Our system uses the farthest neighbor method (FN). In the beginning, each cluster has only one image, and the system repeats merging clusters until all distances between them are more than a certain threshold. We adopt a hierarchical clustering method out of the existing many clustering methods, because a hierarchical clustering method requires the minimum distance as the predetermined constant and does not require the number of clusters in advance, which is different from the k-means clustering method.
4. It throws away small clusters that have fewer images than a certain threshold value, regarding them as being irrelevant. It stores all images in the remaining clusters as output images (Figure 2 (4)).
5. It selects images from group B whose distances to the images in the remaining clusters of group A are small and adds them to the output images(Figure 2 (5)).

After this image-feature-based selection, for raising the precision, we introduced word vectors of HTML files with embedded images into the image selection stage in addition to image feature vectors. The image-gathering module carries out the second selection for group B images by using word vectors extracted from the HTML documents with embedded images for improving results. Introducing the word vectors enables it to eliminate images embedded in the HTML documents whose topics are irrelevant and to ignore them, since a word vector can be regard as a feature vector representing the content of an HTML document.

6. The system eliminates HTML tags and extracts words (only nouns, adjectives, and verbs) from HTML documents with embedded images selected by the aforementioned image feature selection. It counts the frequency of appearance of each word in the HTML documents, selects the top 500 words in terms of the frequency, and makes a 500-dimensional word vector whose elements are word frequencies weighted by Term Frequency and Inverse Document Frequency (TFIDF) [13] for each of the images. All word vectors W_i are normalized so that $|W_i| = 1$.

   In addition, we used the Latent Semantic Indexing (LSI) methods [3], which can abstract semantic content of HTML documents from word vectors. These methods compress word vectors with singular value decomposition in the same way as Principal Component Analysis (PCA). We compressed a 500-dimensional word vector into one of 100 dimensions.
7. For selected images in group A, it clusters their word vectors based on the k-means clustering method. In the experiments, we set the number of remaining clusters in the image-feature-based selection for group A to k. We used the inner product as the dissimilarity (distance) between word vectors.
8. From selected images in group B, it picks up images whose distance to the masses of clusters in group A is less than a certain threshold in terms of word vectors. They are output images of group B found by the word-feature-based selection, while unselected images in group B are thrown away. This processing means selecting images embedded in HTML documents with relevant topics from group B images on the supposition that all images in group A are relevant (Figure 2 (6)).

At present, these two kinds of image-filtering described here are processed independently. Although we tried to combine these processes into one in our preliminary experiments, it was difficult to decide the weight for combining two kinds of features and we did not obtain good results.

We utilize four kinds of thresholds at the image-feature based filtering and the word-feature based filtering. At the first selection, we use one for FN clustering, one for removing small clusters, and one for selecting some images from collected images as group B. At the second selection, we use one for selecting images from group B. Based on the results of the preliminary experiments, in which the average precision of collected raw images in group A and B were about 80% and 40% respectively, we have adjusted them so that 90% of images in group A remains and 30% of images are selected from collected images in group B on the average.

2.4 Query Expansion and Re-gathering

In our image-gathering module, the more URLs of HTML documents we obtained, the more images we could gather. However, for one set of query keywords, the number of URLs obtained from Web search engines was limited because commercial search engines restrict the maximum number of URLs returned for one query. Thus, we introduce the query expansion method [14] for generating automatically new sets of query keywords for search engines.

The system extracts the top ten words (only nouns, adjectives, and verbs) with high frequency except for initial query keywords from all HTML files with embedded output images of the initial image gathering, and regards them as subsidiary query keywords. It generates ten sets of query keywords by adding each of ten subsidiary words to a main keyword, and then obtains a large number of URLs for the ten sets of query keywords. Then, for carrying out the second image gathering, using obtained URLs, the system goes through the collection and selection stages again. This technique enables the number of images gathered from the Web to increase greatly.

3. IMAGE LEARNING AND CLASSIFICATION

We make image classification by image-feature-based search, which is a k-nearest neighbor variant. First, in the learning stage, the image-learning module extracts image features from images gathered by the gathering module and associates image features with the classes represented by the class keywords. Next, in the classification stage, we classify an unknown image into one of the classes corresponding to the class keywords by comparing image features.

In our method of image classification, image features of not only a target object but also non-target objects such as background included in the image are used together as a clue of classification, since non-target objects usually have strong relation to a target object. For example, a cow usually exists with grass field and/or fence in farm, and a lion usually exists in Savannah or zoo. Although the number of combination of a target object and non-target objects is large, we think that we can deal with this largeness by gathering a large amount of image from the Web and using them as training images. Here, we do not set up "reject", and then all test images are classified into any class.

3.1 Signatures and Earth Mover's Distance

We exploit two kinds of image features for learning and classification: *color signature for block segments*, and *region signature for region segments*. A *signature* describes multi-dimensional discrete distribution, which is represented by a set of vectors and weights. In case of *color signatures*, a vector and a weight correspond to a mean color vector of each cluster and its ratio of pixels belonging to that cluster, respectively, where some color clusters are made in advance by clustering color distribution of an image. In case of *region signatures*, a set of feature vectors of regions and their ratio of pixels represents a region signature.

To compute dissimilarity between two signatures, Earth Mover's Distance (EMD) has been proposed [12]. Intuitively, given two signatures, one can be seen as a mass of earth properly spread in the feature space, the other as a collection of holes in the same space. Then, the EMD measures the least amount of work needed to fill the holes with earth. Here, a unit of work corresponds to transporting a unit of earth by a unit of ground distance which is a distance in the feature space. The EMD is based on the transportation problem and can be solved efficiently by linear optimization algorithms.

Formally, let $P = \{(\mathbf{p}_1, w_{p_1}), ..., (\mathbf{p}_m, w_{p_m})\}$ be the first set with m elements, where $\mathbf{p}_i$ is the feature vector and w_{p_i} is its weight; $Q = \{(\mathbf{q}_1, w_{q_1}), ..., (\mathbf{q}_n, w_{q_n})\}$ the second set with n elements; and $d_{ij} = d(\mathbf{p}_i, \mathbf{q}_j)$ the ground distance matrix where d_{ij} is the distance between $\mathbf{p}_i$ and $\mathbf{q}_j$. The EMD between sets P and Q is then

$$\mathrm{EMD}(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \qquad (1)$$

where $\mathbf{F} = [f_{ij}]$, with $f_{ij} \geq 0$ the flow between $\mathbf{p}_i$ and $\mathbf{q}_j$, is the optimal admissible flow from P to Q that minimizes the numerator of (1) subject to the following constraints:

$$\sum_{j=1}^{n} f_{ij} \leq w_{p_i} \ (1 \leq i \leq m) \quad , \quad \sum_{i=1}^{m} f_{ij} \leq w_{q_j} \ (1 \leq j \leq n)$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \quad = \quad \min(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j})$$

In addition, an easy-to-compute lower bound for the EMD between signatures with equal total weights is the distance between their centers of mass in case that the ground distance is defined by the Euclidean distance.

The distance between two discrete distributions represented by signatures can be computed by using the EMD, while conventional distance such as the Euclidean distance expresses one between two points represented by vectors. Since the number of elements of a signature is variable, the signature representation is superior to conventional fixed-size color histograms in terms of expressiveness and efficiency. The EMD are found to be the most excellent distance on the average among distances commonly used in CBIR, as indicated by the prior work of Y.Rubner et al. [11].

We describe two kinds of feature-extracting and classification methods using the EMD in the following sections.

3.2 Color Signatures

To obtain *color signatures*, first, we normalize the size of training images into 240×180, and divide them into 16 and 9 block regions as shown in Figure 3. We make a color signature for each of these 25 block regions. The number of blocks is decided as 25 by the preliminary experiments. Next, we select some dominant colors by clustering color vectors of each pixel into color clusters by the k-means method. In the experiments, the number of color clusters is 15 or less, and it is decided in order not to make a cluster whose weight is less than 0.005. We make a color signature for each block with elements consisting of a mean color vector of each cluster and its ratio of pixels belonging to that cluster. A mean color vector is represented by the Lu^*v^* color space which is designed in order that the Euclidean distance between two points in this space matches the human color sense, so that we use the Euclidean distance as ground distance.

In the classification stage, first, we extract color signatures from each block in an image to be classified (a test image) in the same way as the learning stage after normalizing its size. We obtain 25 sets of signatures for one test image. Next, we search all blocks of training images of each class for the block with the minimum distance (dissimilarity) to each block of the test image. Here, the distance is computed by the EMD. In the next step, we sum up the minimum distances between the test image and training images of each class for 25 all blocks. This search and computation is carried out for all the classes. We compare the total distances among all the classes, and we classify the test image into the class whose total distance is the smallest. In the actual implementation, we used lower bound of the EMD to reduce frequency of computation of the EMD.

3.3 Region Signatures

To obtain *region signatures*, we carry out region segmentation for images instead of dividing images into block segments after normalizing their size as shown in Figure 4. Many methods of region segmentation have been proposed so far. Here, we employ a simple segmentation method based on k-means clustering used in [18] and a sophisticated color segmentation method, JSEG [4].

In case of using k-means, first, we divide a learning image into 4×4 small blocks, and for each block we compute a mean color vector in the Lu^*v^* color space and a texture feature vector, which consists of square means of HL elements, LH elements and HH elements obtained by Daubechies-4 wavelet transform to each 4×4 block. Both vectors are

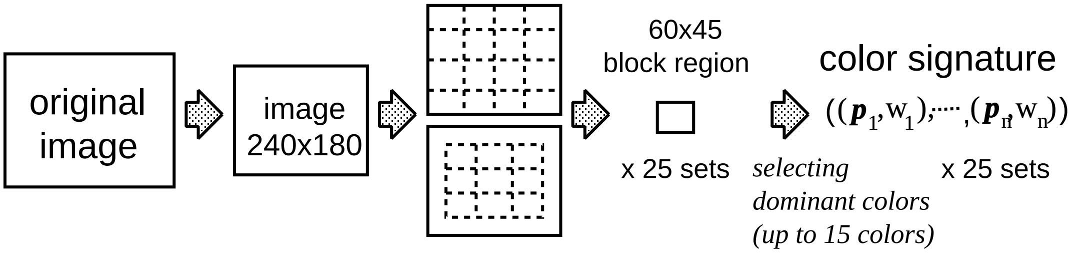

Figure 3: Color signatures for block segments.

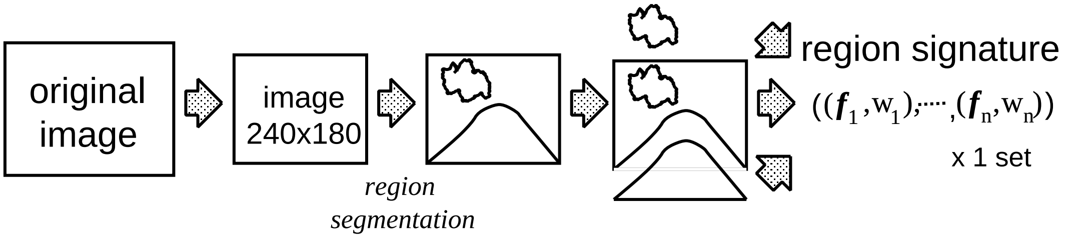

Figure 4: Region signatures for region segments.

three-dimension, so that a six-dimension feature vector is obtained for each block. Next, we cluster all blocks in a learning image into some regions by the k-means method in the similar way as computing color signatures. In the experiments, the number of color clusters is 15 or less, and it is decided in order not to make a cluster whose weight is less than 0.005. Then, we compute a mean 6-dimension feature vector for each region. In addition, for making a region signatures we extract three more features about the shape of a region. We use normalized inertia of order 1 to 3 as three features to describe the shape of a region. Finally, we make a region signature with elements consisting of a nine-dimensional feature vector for each region and its ratio of pixels belonging to that region.

In case of using JSEG as a segmentation method, first, we divide an image into some region segments by JSEG, and then extract a nine-dimensional feature vector from each segment in the same way as the case of k-means. Finally, we obtain one region signature for one image.

In the classification stage, we employ the k-nearest neighbor (k-ANN) method to classify an unknown input image into a certain class. The value of k is decided as 5 by the preliminary experiments. We used the Euclidean distance as ground distance to compute the EMD.

3.4 Conventional Color Histogram

For comparing conventional methods with EMD-based methods, we also make classification experiments using a color histogram in Lu^*v^* color space, higher-order local autocorrelation features [9] and DCT (Discrete Cosine Transform) coefficients as image features. Each dimensions are 64, 25 and 5, respectively. Total dimension is 94.

In the learning stage, in the same way as the color signature, we divide an image into 25 blocks, and compute a feature vector for each block. Next, we compress each feature vector into a 20-dimension vector by the principal component analysis (PCA). The cumulative contribution rate of the eignspace spanned by the 20 principal vectors exceeds 95 percent.

In the classification stage, first, we extract image features from each block in each test image in the same way as the learning stage, and compress them into 20-dimension feature vectors. We obtain 25 sets of image feature vectors for each test image. Next, we search all blocks of training images of each class for the block with the minimum distance to each block of the test image. Here, the distance is computed as the Euclidean distance. In the next step, we sum up the minimum distances between the test image and training images of each class for 25 blocks. This computation is carried out for all the classes. Finally, we compare the total distances among all the classes, and classify the test image into the class whose total distance is the smallest.

4. EXPERIMENTAL RESULTS

We made ten kinds of experiments from no.1 to no.10 shown in Table 1. For no.8, we made experiments three times using three different training image sets gathered from the Web independently. Note that we used the query expansion technique for only experiments no.8' and no.8".

In the experiment no.1, we gathered images from the Web for 10 kinds of class keywords related to animals shown in Table 2. By the image-gathering module about ten thousands URLs were fetched from six major text search engines, Google, InfoSeek, Excite, Lycos, InfoNavi and Goo Japan. The total number of gathered image was 4582, and the precision by subjective evaluation was 68.2%, which is defined to be $N_{OK}/(N_{OK} + N_{NG})$, where N_{OK}, N_{NG} are the number of relevant images and the number of irrelevant images to their keywords. In the left side of Table 2, we show the number of URLs of HTML files fetched from Web search engines and image URLs extracted from HTML files, the number of images collected in the collection stage and images selected in the selection stage, the number of clusters made in the selection stage and, the precision of gathered images.

In the right side of Table 2, we show the classification

Table 2: Results of image-gathering (left) and classification (right) in the experiment no.1

	#URLs	#URLs	#collected images			# of	#selected images				exp. no.1		
class	HTML	images	A	B	A+B	clusters	A	B	A+B	pre.	rec.	pre.	F
bear	4451	3764	279	515	794	24	269	150	419	56.4	21.0	31.1	25.1
cat	4390	3460	291	868	1159	15	240	114	354	62.0	28.0	60.9	38.4
dog	4482	3133	280	741	1021	24	267	303	570	75.7	40.0	23.3	29.4
elephant	4305	3094	325	628	953	27	295	211	506	65.5	25.0	23.1	24.0
tropical fish	4113	3225	173	635	808	15	165	110	275	89.9	22.0	74.6	34.0
lion	4342	5497	307	669	976	26	283	221	504	77.0	45.0	25.2	32.3
penguin	4351	3588	310	818	1128	24	272	304	576	57.0	33.5	29.0	31.1
sheep	4383	3350	209	523	722	18	200	147	347	64.0	13.0	34.2	18.8
tiger	4492	3673	277	442	719	24	253	152	405	68.7	24.0	32.2	27.5
whale	4384	3769	385	809	1194	31	354	238	592	72.4	66.5	39.0	49.2
total/avg.	43693	36553	2836	6648	9484	22.8	2598	1950	**4582**	**68.2**	31.8	37.3	**34.3**
avg. by region (1)											29.4	30.3	**29.8**
avg. by region (2)											26.5	27.3	**26.9**
avg. by color hist.											25.9	30.2	**27.8**

Table 1: Nine experiments.

	# of	# of	precision	test images	
no.	classes	images	(%)	#	source
1	10	4548	68.2	CV	Web
2	10	3102	100†	CV	Web
3	10	500	100‡	CV	Corel
4	10	4548	68.2	50	Corel
5	10	3102	100	50	Corel
6	20	5694	61.2	CV	Web
7	20	3485	100†	CV	Web
8	20	5694	61.2	50	W+C††
9	20	3485	100†	50	W+C††
8'	20	16498	???	50	W+C††
8"	20	32321	???	50	W+C††
10	50	22725	???	LO	Web

†selection of correct images by hand
‡Corel Image as a learning set
††Web images + Corel images
CV: cross-validation, LO: leave-one-out

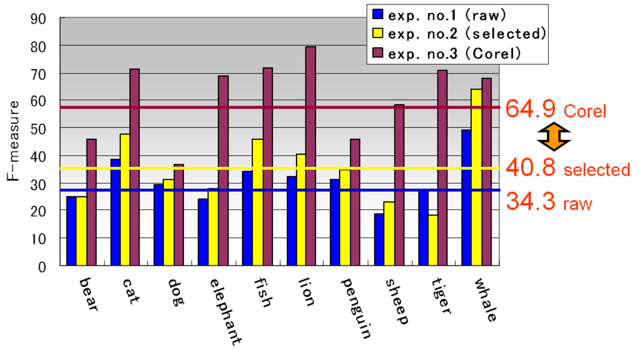

Figure 5: Results of experiments no.1, 2 and 3 by the color signature.

result evaluated by the 10-fold cross-validation. In this section, the tables describe only results by color signatures in each class, since most of results by color signatures are superior to results by region signatures using k-means, results using JSEG and the results by the conventional color histogram method. In the tables, "region (1)" and "region (2)" mean region signature using the k-means clustering and region signature using the JSEG region segmentation method and "color hist." means the color histogram method described in Section 3.4. In the tables, the recall is defined to be M_{OK}/M_{test}, the precision is defined to be $M_{OK}/(M_{OK}+M_{NG})$ and F-measure is the harmonic mean of the recall and the precision, where M_{OK}, M_{NG}, and M_{test} are the number of correctly classified images, the number of incorrectly classified images, and the number of test images for each class, respectively. All values are represented in percentage terms. In the experiment no.1, we obtained 34.3 as the F-measure value by color signatures (Figure 5).

In the experiment no.2, we selected only correct images for each class from gathered images by hand, and the classification experiment was carried out using them. The result is shown in Table 3. Compared to no.1, the F-measure increased. Especially, the result of "whale" was good, since most of "whale" images on the Web were images of "whale watching" scene and they are similar to each other.

In the experiment no.3, we made a classification experiment not for Web images but for the 500 images of 10 classes picked up from Corel Image Gallery. The classification result evaluated by the 10-fold cross-validation is shown in Table 3. Since Corel Image Gallery includes many similar images to each other unlike Web images, a high F-measure value, 68.1, was obtained by region signatures (1). This suggests that Corel images are not as diverse as Web images and easier to classify.

In the experiment no.4 and no.5, we used the gathered images in the experiment no.1 and no.2 as training images and the Corel images as test images. The results are shown in Table 3. In no.4 and no.5, we obtained 25.4 and 31.5 as F-measure, respectively. Since "dog", "tropical fish", "lion", "penguin" and "whale" have some typical patterns and both of the gathered images and the Corel images include the images with the similar typical patterns, their F-measure achieved relatively high values. On the other hand, since "bear", "cat", "elephant", "sheep" and "tiger" had no typical patterns, their F-measures were relatively low.

In the experiment no.6 and no.7, we made an experiment for 20 class keywords which include many different kinds of words in the same way as the experiment no.1 and no.2. Figure 6 shows part of the images gathered from the Web in the experiment no.6 and no.7. Compared to the expected F-measure, 5.0, in case of the random classification, we ob-

Table 3: Results of image-gathering and classification in the experiment no. 2, 3, 4 and 5

	exp. no.2			exp. no.3			exp. no.4			exp. no.5		
class	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F
bear	17.1	46.2	25.0	36.0	62.1	45.6	8.0	15.4	10.5	4.0	40.0	7.3
cat	34.3	78.7	47.8	61.2	85.7	71.4	4.1	33.3	7.3	6.1	42.9	10.7
dog	58.6	21.5	31.4	24.0	75.0	36.4	24.0	14.8	18.3	58.0	21.3	31.2
elephant	25.0	32.1	28.1	68.0	69.4	68.7	34.0	34.7	34.3	16.0	25.8	19.8
tropical fish	35.7	62.5	45.5	58.0	93.5	71.6	22.0	61.1	32.4	30.0	46.9	36.6
lion	47.9	35.1	40.5	82.0	77.4	79.6	30.0	19.5	23.6	36.0	27.3	31.0
penguin	47.9	27.3	34.8	50.0	42.4	45.9	26.0	19.7	22.4	48.0	25.5	33.3
sheep	17.1	36.4	23.3	80.0	46.0	58.4	8.0	23.5	11.9	4.0	18.2	6.6
tiger	10.7	60.0	18.2	72.0	69.2	70.6	4.0	7.4	5.2	10.0	45.5	16.4
whale	75.0	55.6	63.8	94.0	53.4	68.1	86.0	32.6	47.3	86.0	40.6	55.1
avg. by color	36.9	45.5	**40.8**	62.5	67.4	**64.9**	24.6	26.2	**25.4**	29.8	33.4	**31.5**
avg. by region (1)	35.4	37.2	**36.2**	67.1	69.2	**68.1**	23.2	20.7	**21.9**	26.0	22.8	**24.3**
avg. by region (2)	30.0	30.6	**30.3**	65.0	67.1	**66.0**	20.4	16.7	**18.4**	24.0	21.5	**22.7**
avg. by color hist.	29.3	39.5	**33.6**	45.9	48.6	**47.2**	16.2	13.8	**14.9**	23.8	20.0	**21.7**

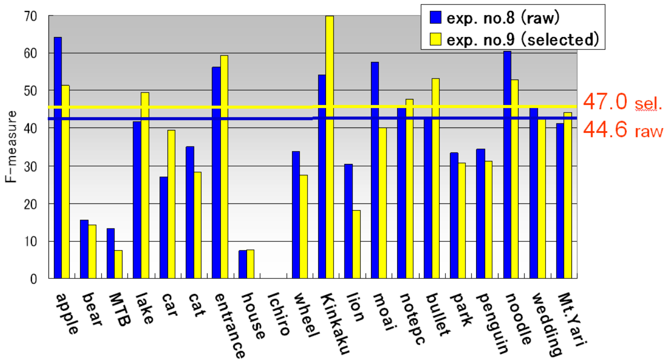

Figure 7: Results of experiments no.8 and 9 by the color signature.

tained much better F-measure, 42.3 and 46.7 shown in Table 4. These results are superior to the result of the experiment no.1 and no.2 for only 10 classes, because all classes used in no.1 and no.2 are related to animals and their training images include many similar images even between different classes. In case of "apple", "Kinkaku Temple" and "noodle", their result were about 60.0, since their scene have some typical patterns and many of their images were applicable to them. On the other hand, for "house", "MTB" and "Ichiro" we obtained only a very low F-measure value, since "house" images had much variation, most part of the body of a mountain bike was only a frame and its size in the images was smaller compared to the background, and "Ichiro", who is a famous baseball player, images had no typical pattern. These results indicate that the difficulty to classify images depends on the nature of the class greatly.

In the experiment no.8 and no.9, we used the gathered images in the experiment no.6 and no.7 as training images and a special test image set as test images. We make a special test image set by selecting various kinds of 50 typical images for each class from Corel Image Gallery and Web images by hand. The classification results are shown in Table 4. In no.8 and no.9, we obtained 44.6 and 47.0 as F-measure by color signatures, respectively (Figure 7). These results are comparable to conventional works of generic image recognition. However, unlike them, we provide training images not by hand, but by gathering images related to class keywords from the World Wide Web automatically.

In addition, for the experiment no.8, we made experiments three times using three different training image sets gathered from the Web independently. We used the query expansion technique for only experiments no.8' and no.8" to gather more images. Table 5 shows the results of three kinds of the experiment no.8, in which we used 50 typical images for each class as test image sets. Three results were nearly equivalent in case of the color signatures and the color histogram. Part of this reason is considered to be because training image sets made in no.8' and no.8" include most of the images gathered in no.8.

In the experiment no.10, we made a classification experiment for 50 class keywords shown in Table 6, which were selected from words related to nature, artifacts and scene as shown in Table 6. We obtained 34.2, 49.0 and 40.3 as the recall, the precision and the F-measure, respectively, by color signatures (Table 5). This results are comparable to the results of the experiment of 20 classes. This indicates that the difficulty of classification depends on the dispersion of image features of each class in the image feature space, not simply on the number of classes. It is hard to collect such various kinds of images as images used in the experiment no.10 by means of commercial image databases, and it has come to be possible by image-gathering from the World Wide Web.

In all the experiments except the experiment no.3, the results by the color signature were superior to the ones by two kinds of the region signature and the conventional color histogram. There are supposed to be two reasons for this. One is that region segmentation for complex real world images are sometimes failed semantically. For example, two distinct objects are segmented as one region. In that case, there is possibility that the color signature using the block segmentation gives better results than the region signatures. Another reason is that one image generates only one region signature, while one image generates 25 color signatures. Since one image is divided into 25 blocks, 25 color signatures are generated for one image. We think the second reason why the color signature brought better results is that we used 25 times image features per one image in the experiments by the color signature compared to the ones by the region

Table 4: Results of the experiment no. 6, 7, 8 and 9

	exp. no.6			exp. no.7			exp. no.8			exp. no.9		
method	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F
apple	30.0	56.8	39.3	30.0	96.0	45.7	52.0	83.9	64.2	36.0	90.0	51.4
bear	25.7	30.5	27.9	18.8	93.8	31.2	14.0	17.9	15.7	8.0	66.7	14.3
mountain bike	10.0	60.9	17.2	6.2	100.0	11.8	8.0	40.0	13.3	4.0	66.7	7.5
Lake Biwa [a)]	42.9	20.8	28.0	43.8	44.9	44.3	64.0	31.1	41.8	52.0	47.3	49.5
car	16.4	79.3	27.2	27.5	78.6	40.7	16.0	88.9	27.1	26.0	81.2	39.4
cat	52.1	21.7	30.6	68.8	19.4	30.2	54.0	26.0	35.1	62.0	18.3	28.3
entrance [b)]	53.6	34.6	42.0	61.3	39.2	47.8	68.0	47.9	56.2	76.0	48.7	59.4
house	15.7	88.0	26.7	1.3	100.0	2.6	4.0	50.0	7.4	4.0	100.0	7.7
Ichiro [c)]	2.1	100.0	4.2	6.4	100.0	12.0	0.0	0.0	0.0	0.0	0.0	0.0
Ferris wheel	15.7	62.9	25.1	16.2	100.0	28.0	22.0	73.3	33.8	16.0	100.0	27.6
Kinkaku Temple [d)]	59.3	41.7	49.0	48.8	86.7	62.4	78.0	41.5	54.2	60.0	83.3	69.8
lion	13.6	86.4	23.5	23.0	100.0	37.3	18.0	100.0	30.5	10.0	100.0	18.2
Moai	19.3	96.4	32.1	11.2	100.0	20.2	42.0	91.3	57.5	26.0	86.7	40.0
note-size PC	35.7	86.2	50.5	26.2	87.5	40.4	30.0	93.8	45.5	32.0	94.1	47.8
Shinkansen train [e)]	47.9	23.8	31.8	42.5	34.3	38.0	66.0	31.7	42.9	58.0	49.2	53.2
park	51.4	23.0	31.8	75.0	17.0	27.8	52.0	24.8	33.5	82.0	19.0	30.8
penguin	36.4	38.1	37.2	30.0	52.2	38.1	32.0	37.2	34.4	24.0	44.4	31.2
noodle [f)]	68.6	53.6	60.2	70.0	40.6	51.4	60.0	61.2	60.6	64.0	45.1	52.9
wedding	42.1	35.8	38.7	47.5	37.3	41.8	46.0	45.1	45.5	52.0	36.6	43.0
Mt.Yari [g)]	60.0	31.0	40.9	58.8	28.3	38.2	70.0	29.2	41.2	78.0	31.0	44.3
avg. by color	34.9	53.6	**42.3**	35.7	67.8	**46.7**	39.8	50.7	**44.6**	38.5	60.4	**47.0**
avg. by region (1)	34.3	37.7	**35.9**	37.0	45.5	**40.8**	40.1	43.1	**41.5**	42.1	47.9	**44.8**
avg. by region (2)	28.3	31.2	**29.7**	33.5	37.6	**35.4**	35.9	37.9	**36.9**	39.3	44.1	**41.6**
avg. by color hist.	26.8	31.4	**28.9**	29.0	39.4	**33.4**	30.4	39.5	**34.4**	28.1	49.6	**35.9**

a) the biggest lake in Japan b) a school entrance ceremony c) the name of a famous baseball player d) a famous temple in Japan e) Japanese bullet train f) Chinese noodle g) a famous mountain in Japan

Table 5: Results of the experiment no. 8, 8', 8" and 10

	exp. no.8			exp. no.8'			exp. no.8"			exp. no.10		
method	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F	rec.	pre.	F
avg. by color	39.8	50.7	**44.6**	41.7	46.9	**44.2**	40.1	50.4	**44.7**	34.2	49.0	**40.3**
avg. by region (1)	40.1	43.1	**41.5**	36.6	40.1	**38.3**	34.7	35.1	**34.9**	27.6	28.6	**28.1**
avg. by region (2)	35.9	37.9	**36.9**	31.2	31.5	**31.4**	28.6	27.1	**27.9**	22.2	22.7	**22.4**
avg. by color hist.	30.4	39.5	**34.4**	32.7	38.2	**35.2**	34.9	37.0	**35.9**	23.9	39.8	**29.9**

signature. Therefore, in case of color signatures, the classification took about 25 times as long processing time as in case of region signatures. In terms of processing time, the region signature is superior to the color signature.

5. CONCLUSIONS

In this paper, we described a generic image classification system equipped with an image-gathering modules from the World Wide Web, and proposed "*Web Image Mining*", which is making use of images gathered automatically from the World Wide Web as training images for generic image classification instead of image collections made by hand.

Although classification rate obtained in the experiments for generic real world images is not high and not sufficient for practical use, the experimental results suggest that generic image classification using visual knowledge on the World Wide Web is one of the promising ways for resolving real world image recognition/classification.

The current system is made by connecting three kinds of modules sequentially and independently. For future works, we will integrate these modules, especially, the image-gathering module and the image-learning module.

At present, we used k-nearest neighbor-like classifier as a classification method, since the signature representation for the EMD is different from the vector representation and widely-used machine learning techniques such as SVM, neural networks and probabilistic inference cannot be applied to it. So that the processing time for classification increases in proportion to the number of learning images, which is crucial problem for scaling up the system. We have to make much improvement in learning and classification methods and extraction of image features for reducing the processing time and obtaining more improved classification rate.

There are many issues to be solved except ones mentioned above. How many classes does a generic classification system have to treat for practical use? How many training images is required for each class? What should we define as a "class"? "House"? Or "Roof", "Wall", "Door" and "Window"? Then, what kind of "house" should the system know? "Western-style House"? "African House"? "Nomadic House"? Or all kinds of "Houses" on the earth? Because of such issues, evaluation is the biggest problem for

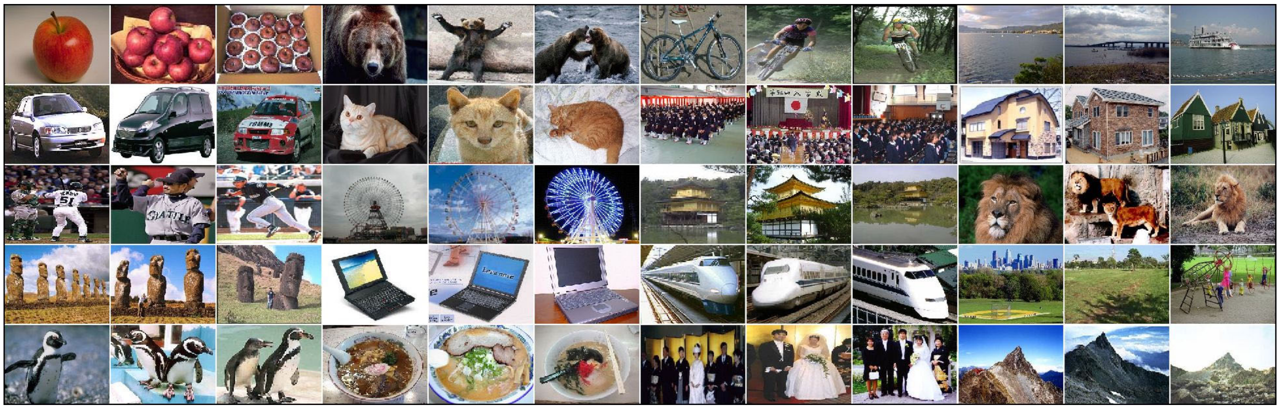

Figure 6: Part of the images gathered from the Web in the experiment no.6 and no.7.

Table 6: 50 kinds of class keywords

type of words	class keywords
nature: animal	elephant, cow, dog, lion, koala
nature: aquatic animal	penguin, tropic fish, whale, seal, swan
nature: plant	Christmas tree, rose, tulip, cactus, bonsai
nature: people	Ichiro, Tiger Woods, Morning-Musume (Japanese idol group), baby, Shishimai (classic dancing)
artifact: vehicle	Shinkan-sen train, police car, racing car, steam locomotive, airplane
artifact: building	Ferris wheel, bay bridge, Tokyo tower, Kinkaku temple, huge statue of Buddha
artifact: tools	notebook PC, chair, digital still camera, mountain bike, helmet
artifact: foods	Chinese noodle, apple, orange, sushi, French food
scene: special event	school entrance ceremony, wedding, seeing cherry blossoms, school sports festival, rice planting
scene: sight	Mt.Fuji, Ayers Rock, sunset, waterfall, Japanese-style open-air bath

generic image classification/recognition. Experimental results sometimes depend on learning sets and training sets more greatly than classification algorithms.

6. REFERENCES

[1] K. Barnard and D. A. Forsyth. Learning the semantics of words and pictures. In *Proc. of IEEE International Conference on Computer Vision*, volume II, pages 408–415, 2001.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Recognition of images in large databases using a learning framework. Technical Report 07-939, UC Berkeley CS Tech Report, 1997.

[3] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[4] 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.

[5] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicons for a fixed image vocabulary. In *Proc. of European Conference on Computer Vision*, 2002.

[6] C. Framkel, M. J. Swain, and V. Athitsos. WebSeer: An image search engine for the World Wide Web. Technical Report TR-96-14, University of Chicago, 1996.

[7] J. Hafner, H. S. Sawhney, et al. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. on PAMI*, 17(7):729–736, 1995.

[8] 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.

[9] N. Otsu and T. Kurita. A new scheme for practical flexible and intelligent vision systems. In *Proc. of IAPR Workshop on Computer Vision*, pages 431–435, 1988.

[10] N. Rowe and B. Frew. Automatic caption localization for photographs on World-Wide Web pages. *Information Processing and Management*, 34(1):95–107, 1998.

[11] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84(1):25–43, 2001.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[14] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[15] S. Sclaroff, M. LaCascia, S. Sethi, and L. Taycher. Unifying textual and visual cues for content-based image retrieval on the World Wide Web. *Computer Vision and Image Understanding*, 75(1/2):86–98, 1999.

[16] J. R. Smith and S. F. Chang. Visually searching the Web for content. *IEEE Multimedia*, 4(3):12–20, 1997.

[17] J. Z. Wang and J. Li. Learning-based linguistic indexing of pictures with 2-D MHMMs. In *Proc. of ACM International Conference Multimedia*, pages 436–445, 2002.

[18] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[19] 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.

[20] K. Yanai. Image collector II: A system for gathering more than one thousand images from the web for one keyword. In *Proc. of IEEE International Conference on Multimedia and Expo*, volume I, pages 785–788, 2003.