Image Retrieval System Based on Machine Learning and Using Color Features

Janez Demšar, Dragan Radolovič, and Franc Solina
janez.demsar@fri.uni-lj.si, franc.solina@fri.uni-lj.si

Faculty of Computer and Information Science, University of Ljubljana
Tržaška cesta 25, SI-1000 Ljubljana, Slovenia

Abstract. We describe an interactive system for content based image retrieval. The system presents the user with 15 randomly selected images from the database. The user grades the images with one of five possible grades (YES, yes, neutral, no, NO) according to what he is looking for. The system returns the first 15 images with the highest probability of YES grade. The attributes used are a combination of color features. Three different machine learning techniques are compared.

Keywords: content based image retrieval, machine learning, color features

1 Introduction

The progress of computer industry has turned a cheap personal computer from an enhanced typewriter and number crunching machine into a powerful multimedia device. With availability of cheap, yet fast mass storage media, this led to foundations of large collections of images, audio and video files. Contrary to the textual databases, multimedia collections are much harder to organize into searchable libraries, which decreases their usefulness [1].

The simplest way to organize a multimedia collection is to manually define a hierarchy of 'themes' and divide the objects onto the defined groups. To find a particular object, the user steps down the hierarchy to the desired subgroup and then browses the acquired objects. Although the idea is straightforward, it has certain drawbacks. It requires a well defined and extendible structure and a great amount of work for manual classification of each object, including the objects that arrive later. In addition, such a structure is normally not disjunctive, i.e. one and the same object (or a subtheme) falls into different groups (or themes), which can be confusing. Our personal experience using such commercial collections is that searching through the structure can be slow and inefficient.

The alternative is to do the things automatically. In systems of this kind, objects are normally not organized in any kind of hierarchy and, in principle, no keywords or similar descriptions are given manually. Instead, certain simple attributes are extracted automatically and the user queries by specifying the values of attributes. Attributes can be given directly, as the user input, or indirectly, by showing the system one or more images with the contents we are looking for.

F. Solina and A. Leonardis (Eds.): CAIP'99, LNCS 1689, pp. 480–488, 1999.

Searching algorithm which compares the values of attributes of the images in the collection with the required values can range from a simple distance measuring to sophisticated artificial intelligence methods.

The second approach is seldom used. The 'manual' approach is reliable in the sense that we always know what to expect, while the alternative might perform better but it might also not work at all. Our opinion is, however, that the amount of multimedia data is becoming infeasible for the manual handling and that the automatic methods must be explored and improved.

In this work we limited ourselves to collections of images. Also, our intention is not to develop a super-fast and super-accurate image retrieving system but merely to explore the usability of machine learning methods in this area and find the solutions for overcoming the encountered drawbacks. We will first describe the related work of other researchers and briefly introduce some previous method that incorporates a machine learning algorithm. After discussing the major difficulties, we shall present a new method for querying image databases.

2 Related Work

Existing systems for content based image retrieving (CBIR) generally use attributes that are manually or automatically extracted from images and then stored and managed in conventional database systems [2]. Besides that, precalculated attributes are often too domain specific or too general. Chabot System [3] for example, integrates a relational database containing keyword and other conventional data with color analysis technique to allow searching by keywords and dominant colors. It allows queries as "*mostlyOrange* and *someBlue*" which should, presumably, describe images of sunset over seas and lakes. The problem with this approach is in finding the right combination of attributes; query must be often refined. Also, those queries do not seem to describe the content of the image accurately enough.

Searching of an image database requires from the user to select or grade some initial images according to their likeliness to the searched images or to set some boundaries for the values of attributes. Attributes used in this context as well as the distances between the attributes must be fairly simple and fast for computation. Different types of attributes can be used. The most popular are *Color attributes* as they can be computed fast and in a straightforward manner. Several types of color attributes and their combinations can be used (histograms, moments, primary colors, averages etc.) which can also be computed separately in predefined parts of the image. Color attributes are not sensitive to location, rotation, scale and resolution. On the average they give good results but they miss images which are to a human observer very similar but of different colors. *Texture* [4] is somewhat more difficult to define and compute than color and is more sensitive to resolution. *Shape* (composition, structure) is much more difficult to define and compute than color attributes. Even if one can reliably recover shape attributes the definition of similarity or the distance among different shapes poses a very difficult problem. But since the human perception of similarity of images can not be reduced only to color and texture, this area of research is very important. Due to computational complexity the current shape attributes used in the framework of image retrieval are limited mostly only to edges, corners and interest points [5]. For shape attributes it is particularly more difficult to attain location, rotation and scale invariance.

Over the Web several commercial products and research systems for content based image retrieval can be tested. *QBIC (Query by image content)* is an IBM product [6] which is based mostly on color, color layout and texture attributes. *VIRAGE* [7] uses also composition and structure. *MetaSEEK* [8] combines the previous two search systems with the home grown *Vseek* using color and texture.

3 Machine Learning for Image Retrieving

The idea of using machine learning tools for image retrieval is not new. Our work is based on [9], where the ID3 algorithm is run to learn from example and counter example images, and the resulting classifier acts as an image query. We describe the method as an illustration of a straightforward application of machine learning techniques and of the related problems.

ID3 [10] is a simple learning algorithm from the "top-down induction of decision trees" family, which recursively divides the examples into groups and further into ever smaller subgroups using values of features as criteria, until it creates a 'clean' or almost clean subgroups. For CBIR, images are divided into subgroups until (almost) all images in a subgroup correspond to the same class (*wanted* or *not wanted*). The result of such a learning is a tree structure with each internal node containing a criteria for obtaining the branch that a particular image goes into, and the leaves containing groups of images of the same kind.

The attribute set used in [9] was simple, mostly describing proportions of basic colors in the image or in the central area of the image. The adaptation to the basic algorithm was a search for informative colors. The system used a local optimization to find colors which could be used as attributes in a decision tree. In the case of querying for images of faces it usually found a color which could be recognized as the skin color to be the most informative and used it in decisions of type 'images with less than 10% of this color do not represent faces'.

Experiments were run on a small collection of 167 images, of which 67 were images of a human face and the rest had different contents. Images were classified manually. Randomly chosen 70% of images (119 images) were given to the learning algorithm as a learning data with which it has built a decision tree. The learned ability to distinguish between images of faces and other images was then tested on the remaining examples. Although proportions of correctly classified images were encouraging, the method has never been put into practical use.

1. Machine learning algorithms can be seen as advanced statistical algorithms and, as always, a small sample means unreliable results. ID3 is quite sensitive to this. The described system has 119 examples as learning data. Would the user be prepared to find and manually classify such a great number of examples to perform a query?

2. Another problem with the approach was that ID3 classified each image as having the desired contents or not having it. Estimating the probabilities of having the desired contents would be more appropriate approach since it would enable the image browser to sort the images and present them to the user with the 'best' images first, instead of presenting only the images which the classifier guesses to have the desired contents and hiding the others.
3. The same holds for the user part. User should not be forced to classify each image as being "good" or "bad". Instead, he must be given a chance to grade the images according to how close they are to the desired image.
4. The system seems to rely on its ability to find informative colors. The proportion of, say, red or blue color in the image is probably far less informative than the proportion of the skin color. Thus, using an approximation for the skin color was essential and the search for informative colors was absolutely required. On the other hand, this search is quite slow.

4 Improvements

It is obvious that the described method requires major modifications. We have tested two algorithms besides ID3, and introduced example weights and class probabilities to soften the classification. We have also added some attributes.

4.1 Learning Algorithms

To solve the first problem, we reconsidered the chosen learning algorithm. ID3 is a strong learning algorithm that presents the gathered knowledge in a 'brain-compatible' form. In many cases, we are interested in the obtained decision tree and its explanation, and do not use it to classify unseen cases. The image retrieval problem is, however, of a different kind. Although we can admire the interpretability of the trees derived by the described system, the user does not really care about it. A more primitive learning algorithm that is able to learn with less examples and can estimate probabilities of classes should be used instead of ID3. In practice it does not matter whether the knowledge is 'readable' or not, since it is improbable that an occasional user would like to understand or even modify it. We decided to try out three different algorithms, ID3, k-nearest neighbors and naive Bayesian classifier.

Instead of classifying *ID3* estimated probabilities of classes. This was achieved by stopping the tree induction before the subgroups were clean and, when "classifying", returning the relative frequency of the node examples corresponding to certain class as the probability of that class.

K-nearest neighbors uses a distance measure (Euclidean distance, Manhattan distance, or some other) to find the k nearest neighbors of an example which is being classified. The algorithm usually selects the most frequent class among the neighbors as a prediction for the example's class. In our case, we are interested in probabilities of classes so the system returns relative frequency of the class as an estimate for the probability. Examples are also weighted by their distance from the reference example.

Naive Bayes classifier is based on Bayesian probability formula. Supposing the independence of attributes, the probability that example E is in class r_k is $P(r_k|E) = P(r_k)\prod_{i=1}^{n}\frac{P(r_k|v_i)}{P(r_k)}$, where v_1, v_2, …, v_n are values of attributes for E and the probabilities $P(r_k|v_i)$ and $P(r_k)$ are estimated from the learning set.

Estimating the probabilities enables the system to rank the images. The user is given the same opportunity. Instead of deciding for or against an image, he assigns grades. The grades are converted to weights of examples. The image with greater positive or negative grade is given a greater weight.

The last problem from the previous section was partially eliminated by using more robust learning algorithms and by introducing some new color attributes. The attributes were defined and extracted by Dragan Radolović [11].

4.2 Color Attributes for Image Query

Color histograms are of high dimension and therefore it is difficult to compute the distance between them. The first and second moment of color histograms [12] are much more compact and easier to compare. The **first and second moment of color histogram** is the average RGB color and its dispersion. In our system they are computed on the whole image and in the central part (middle three fifths) of the image. **Compactness of colors** measures the proportions of pixels of "mostly red", "mostly green", "mostly blue" and "gray" colors surrounded by pixels of a similar color. The pixel of color (r, g, b) is "mostly red" if $r - \max(g, b) > \delta$, "mostly green" and "mostly blue" are defined similarly, all remaining pixels are "gray". **Proportions of basic colors** are proportions of pixels of "mostly red", "mostly green", "mostly blue" and "other" colors.

ID3 discretizes the attributes by finding such boundary that informativity of the obtained binary attribute is maximal. For Naive Bayesian classifier, attributes are discretized on five intervals with approximately equal number of examples. K-nearest neighbors normalizes the values to interval $[0, 1]$.

5 Implementation

The color attributes are precalculated for all images in the database. The computation of all color attributes for 1000 images takes less than 10 minutes on a PC. This set of attributes serves as input to the learning part of the system. The learning part of the new system was done by our general machine learning system ML*. ML* is a modular system which incorporates all of the listed learning algorithms, all of them also support example weighting.

First, the system presents the user a certain number (15) of randomly selected images and asks him to grade them (Fig. 1). Each image can be given one of five grades, with the lowest meaning that the image is completely different from what he looks for and the highest meaning that the image is of exactly the right type. The user is not required to classify all the images.

Fig. 1. An example of a query for human faces. The user gave the highest grade to the two faces and the second highest to the image of a group of people. All other images have the lowest grade.

The data posted, grades are converted to classes and weights. The images having the middle grade are skipped. The lower two grades are converted to 'NO' class, with the lowest having weight 1 and the other 0.5. The higher two grades correspond to 'YES' class with the highest having weight 1 and the other 0.5. The precalculated attributes together with the just constructed class and weight values are given to the learning algorithm. The obtained classifier is used to estimate the probabilities of 'YES' class for all other images in the database, which have not been presented to the user yet. The fifteen images with the highest probability of 'YES' class are presented to the user as an answer to his query and examples for its refinement.

If the query was unsuccessful the user can grade the presented set of images to point at the good and the bad examples again. The images are added to the previous examples. The learning algorithm re-learns with the new examples and a new selection of fifteen images is presented to the user again. In the case of a satisfactory answer, the user can refine the query by being more selective when

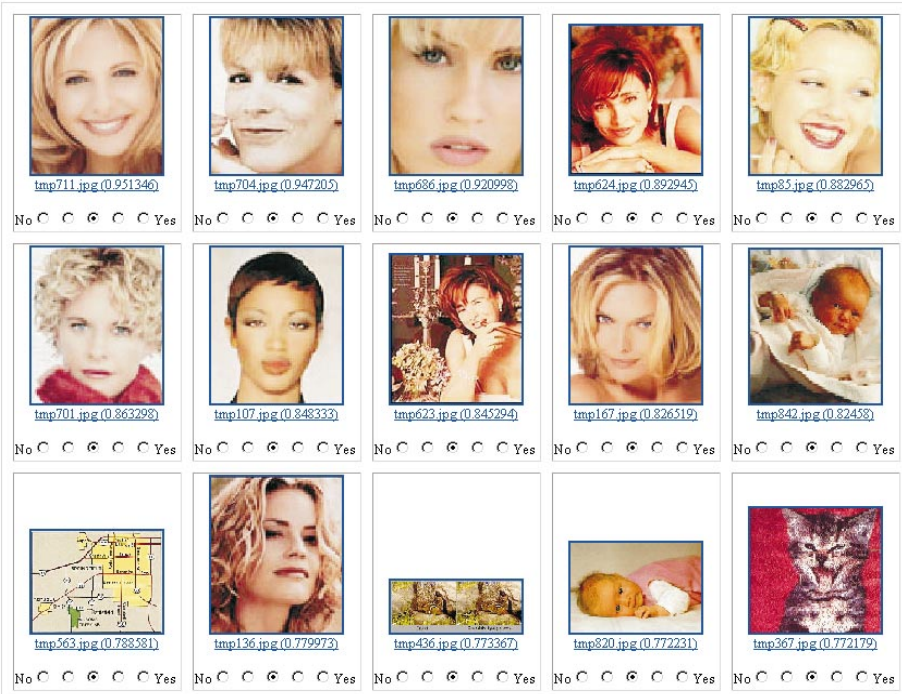

Fig. 2. Answer of the query from Figure 1. Only three of fifteen images are complete misses, all other images represent human faces.

assigning the good grades. The system decreases the weights of images of past queries and thus gives the new examples a bigger importance. The user can count on that and request a larger concept in the beginning and narrow it (become more strict) later, without the good grades from the first rounds of the process interfering in the later rounds.

Finally, if the query was a complete success, the user can request the next fifteen closest images using exactly the same classifier as before.

6 Results

Instead of statistical analyses which can distract the attention from actual usability of the system, we tested the system "visually" by using it to retrieve images from a database containing 1000 images.

Figures 1 and 2 show an example of a query for images of faces and the answer. Note that from the fifteen random images that were initially chosen by the system, only two presented a positive and one half-positive example. The answer is relatively accurate; only three of fifteen images are complete misses.

As expected, the *k-nearest neighbors* method was by far the best of the three learning methods tested. ID3 proved to be unable to handle such a small learning set with a great number of attributes; also, its probability estimation method assigned just a few different probabilities to the images so that many images had the same probability, which is impractical. *Naive Bayes classifier's* poor performance was probably due to the strong correlations between the attributes. K-nearest neighbors method also proved to be efficient in retrieving images of faces, animals, landscapes, cityscapes and similar. It is also fast enough, although the other two methods were a bit faster.

7 Conclusions and Further Work

As mentioned, our goal was to adopt the general machine learning methods for the use in image retrieving systems. Different methods were examined and small adaptations were made to incorporate them in a web based search engine. Experiments show that the most promising method for now is the *k-nearest neighbors*, especially for its ability to work with smaller example sets than the smarter methods. This is not surprising, the fact is that most of working systems for image retrieval already use a simplified version of this method. Its performance could be further improved by refining probability estimation function and how it is influenced by the learning examples of different classes at different distances from the example which is being classified.

Although the system is actually able to retrieve images belonging to simple "concepts", like those mentioned above, the concepts that can be distinguished from each other are much to wide. For example, an image retrieving system is expected to be able to retrieve not just "images of faces" but at least "images of female faces" if not even "images of faces of middle-aged blondes with green eyes, round glasses and not too much make up". It is obvious that the given color attributes do not describe images precisely enough. Therefore, the future work shall focus mostly on searching for new image describing features.

References

1. Ramesh Jain. Visual information management. *Communications of the ACM,* 40(12): 31–32, 1997.
2. V. N. Gudivada, V. V. Raghavan. Content-Based Image Retrieval Systems. *Computer*, 28:18–12, 1995.
3. Ogle V. E. (1995) Chabot: Retrieval from a Relational Database of Images, In *Computer 28*, pp. 40-48.
4. F. Liu, R. Picard. Periodicity, directionality, and randomness: world features for image modeling and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(7):722–733, 1996.
5. C. Schmid, R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 19(5):530–535, 1997.
6. `http://wwwqbic.almaden.ibm.com/`
7. `http://www.virage.com/virdemo.html`

8. http://ctr.columbia.edu/metaseek/
9. J. Demšar, F. Solina. Using machine learning for content-based image retrieving. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume IV, pages 138–142, Vienna, Austria, August 1996.
10. J. T. Quinlan. Induction of decision trees, In *Machine Learning 1*, pp. 81-106. Kluwer Academic Publishers, 1986.
11. Dragan Radolović. *Image database queries based on color information*, B.Sc. Thesis (in Slovene). Faculty of Computer and Information Science, University of Ljubljana, 1998.
12. M. Stricker, A. Dimai. Spectral covariance and fuzzy regions for image indexing. *Machine vision and applications,* 10(2):66–73, 1997.