

# Mining Association Rules between Low-level Image Features and High-level Concepts

Ishwar K. Sethi\*      Ioana L. Coman\*\*  
Daniela Stan\*

\*Intelligent Information Engineering Laboratory,  
Department of Computer Science and Engineering,  
Oakland University, Rochester, MI 48309-4478, USA.

\*\*Department of Electrical Engineering and Computer Science,  
Syracuse University, Syracuse, NY 13244-4100, USA.

## ABSTRACT

In image similarity retrieval systems, color is one of the most widely used features. Users who are not well versed with the image domain characteristics might be more comfortable in working with an Image Retrieval System that allows specification of a query in terms of keywords, thus eliminating the usual intimidation in dealing with very primitive features. In this paper we present two approaches to automatic image annotation, by finding those rules underlying the links between the low-level features and the high-level concepts associated with images. One scheme uses global color image information and classification tree based techniques. Through this supervised learning approach we are able to identify relationships between global color-based image features and some textual descriptors. In the second approach, using low-level image features that capture local color information and through a k-means based clustering mechanism, images are organized in clusters such that images that are “similar” are located in the same cluster. For each cluster, a set of rules is derived to capture the association between the localized color-based image features and the textual descriptors relevant to the cluster.

**Keywords:** Content-based image retrieval, classification trees, k-means clustering, association rules.

## 1. INTRODUCTION

One of the open research areas today is to find methods that will allow efficient searching, browsing and retrieval through image and video collections. There are two frameworks for image retrieval: Text-based and Content-based. The text-based approach can be traced as early as 1970’s. The images are first manually annotated by text descriptors, which are then used by a Database Management System (DBMS) to perform image retrieval. The method was intensively studied, and research groups provided many solutions to the problems of data modeling, multi-dimensional indexing, query evaluation. But there exist two major difficulties with this approach: One is the large amount of human labor required for manual annotation (i.e. weather satellite images collected daily); The other is the annotation impreciseness, caused by the subjectivity of human perception (different people may perceive differently the content of an image).

In the early 1990’s the content-based approach emerged. Instead of manually annotating the images, they are indexed by their own visual content, such as color, shapes, texture, etc. This approach created a new framework for Image Retrieval. The advances in this direction are mainly contributed by the Computer Vision research groups. Since then, both research labs and commercial companies built Image Retrieval Systems which employ different techniques (Chang and Fu [2], Faloutsos et al. [6], Flicker et al. [7], Pentland et al. [13], Sethi et al. [17], Smith and Chang [19]). However, there are still many open issues to be solved.

A fundamental difference between Content-based retrieval systems and Text-based retrieval systems is that the human interaction is an indispensable part of the latter system. Humans tend to use high-level features in everyday life (keywords, text descriptors, etc.). Current Computer Vision techniques can automatically extract from images

---

E-mail correspondence:

Ioana L. Coman: ilcoman@ecs.syr.edu, Ishwar K. Sethi: isethi@oakland.edu, Daniela Stan: dstan@oakland.edu

mostly low level features (color, texture, etc.). And in general, there is no direct link between the high-level concepts and the low-level features.

However, this gap can be reduced by using techniques from Neural Networks, Genetic Algorithms and Clustering research areas that provide powerful learning tools for supervised or unsupervised learning. The nice part about these tools is that they can be used for an off-line processing step, during which a connectivity network (knowledge representation network) is created to point out the links between different low-level features and the high-level concepts. In this paper we consider the use of *decision trees* methodology and a variation of *k-means clustering* to automatically explore the data extracted from images, and find mappings between low level color features and high level textual descriptors.

The investigations we made will show two different approaches in finding these mappings. In the supervised approach, by using decision tree methodology, images are clustered based on their initial keywords assignment in such a way to minimize the classification error. The rules are derived from the resulting induced trees (one tree/keyword) and combined into a comprehensive rule base. The second approach, partially unsupervised, is performed in two stages: First, a modified k-means clustering step is executed for organizing the image collection into a hierarchy of clusters. Through this step, images with “similar” color content are assigned to the same clusters. Then, for each cluster, a mapping function is determined which is based on two things: cluster semantics and statistical properties of the low-level features of the images assigned to the cluster.

The organization of the paper is as follows. Section 2 presents two methods used to encode the color information of images. One captures the global color information while the other captures the localized color distribution in the images. These representations are used to organize the images into clusters by the two clustering techniques. The k-means based technique is used to build a hierarchy of clusters, that will be afterwards mapped into their optimal textual characterization. The decision tree based clustering technique is used to find mappings between textual descriptors (keywords) and global color distribution in a given image. Section 3 provides a brief exposition of decision tree and k-means clustering methodologies and describes the proposed scheme for finding mappings between low-level features and textual features associated with an image. The performance of the two approaches is described in Section 4. Section 5 presents a summary of the work and some concluding remarks.

## 2. IMAGE FEATURE EXTRACTION

At the basis of Image Retrieval systems is feature (content) extraction task. The text-based retrieval approach is based on *text features* (keywords, annotations, etc.) while the content-based retrieval uses *visual features* (color, shape, texture, faces, etc.). Visual features can be further more divided into *general features*, such as color, shape and texture, and *domain specific* features, such as fingerprints, human faces. Domain specific features were extensively studied in Pattern Recognition literature, and they usually involve extra problem domain knowledge. In our experiments we considered “keyword association” between text features and visual features based on color. The following sections will describe the low-level features extracted and the text descriptors association.

### 2.1. Image Color Content Representation

When using color in Image Retrieval systems there are two important considerations: (1) the choice of the color-coordinate system, and (2) the scheme that captures the color distribution in the image. Most of the color systems used today are oriented either towards hardware applications (RGB (Red, Green, Blue), CMY (Cyan, Magenta, Yellow) or YIQ models) or towards software applications where color manipulation is an important task (HSI, HLS, YUV, Munsell system). While all systems were successfully used in different image retrieval systems, the latter ones appear more appropriate because they are based on the intuitive appeal of the artist’s tint, shade, and tone.

All colors are seen as variable combinations of the so-called *primary colors*, red (R), green (G), and blue (B). In order to distinguish one color from another we use the following descriptors: hue, saturation and lightness. *Hue* represents the dominant color as perceived by an observer; when we specify that an object is blue, red or orange we refer to its hue. *Saturation* represents the relative purity or the amount of white light mixed with a hue. *Lightness* is a subjective descriptor that expresses the achromatic notion of intensity of a reflecting object and it is essential in describing the color sensation. In the HSL color model, the lightness component, L, is separated from the color information, and hue and saturation components are related to the way we perceive color. The conversion algorithm from the RGB to HLS color space can be found in Foley *et. al* [8] (pg. 595).

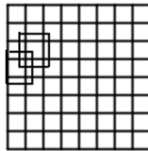
In addition we have to find a representation for the features in an image which will be suitable for classification. We considered two approaches: *global histogram*, to capture the global color information in the image, and a *fixed*

*image-partitioning* scheme, to encode the spatial color distribution in an image.

The *global histogram* of a 24-bit color image is a discrete function  $h(r_k) = n_k/n$ , where  $r_k$  is the  $k$ th color encoding,  $n_k$  is the number of pixels with that color,  $n$  is the total number of pixels in the image, and  $k=0, 1, \dots, 2^{24} - 1$ . A resolution this high is quite unnecessary and would create a feature vector which is unsuitable for the classification task.

In most applications the color space is quantized to a much lower resolution, i.e. 16 bins for hue, 16 bins for saturation and 8 bins for intensity resulting in 2048 combinations. There are two schemes for color quantization: fixed or adaptive. In the fixed quantization approach each color axis is divided uniformly into a specified number of bins. This approach might not yield a statistically optimal representation since some color bins are more likely to be populated than others. In the adaptive quantization the division of the color axis is based on the global color information extracted from a subset of images representative of the actual database. In our experiments, we used the adaptive color quantization technique. We compute the three global histograms over all the images in the database, for hue, saturation and intensity, respectively. For each global histogram, the quantization intervals are determined such that the resulting global histogram for the quantized color space to be as close to a uniform histogram as possible. We use this representation to capture the relationships between some textual descriptors and the overall color distribution in an image.

In the *fixed image-partitioning* representation each image is divided into  $M \times N$  overlapping blocks (Figure 1).



**Figure 1.** The fixed partitioning

Through the overlapping blocks we allow a certain amount of fuzzy-ness to be included in the representation. For each block, we compute three local histograms (hue, saturation, and intensity). From each of these local histograms we extract the location of its *area-peak*. This location is determined as follows: We consider an arbitrary window of fixed size and calculate the area of the portion of the histogram lying within this window. We then determine the location of the window which yields the maximum area. This is the location of the area-peak of the histogram. Using this approach the color distribution of an image is captured in a feature vector with  $3 \times M \times N$  attributes. This representation was successfully used for similarity retrieval (Sethi et al. [17]) and multidimensional indexing (Sethi and Coman [16]) in image databases.

## 2.2. Image Textual Descriptors

Designing an effective image retrieval system is a complex problem. The user judges retrieval systems by the performance of their search tools. But the accuracy of the results is highly dependent on the data representation that supports the database design, and the cataloging and searching methodologies. Database research groups provide many reliable solutions to the previous problems. The difficulty comes from the fact that most of the techniques are developed for text-based documents and do not entirely apply to the new multimedia documents, especially the ones containing visual data (images, videos).

Text-based descriptions are the most common form of data representation used by the database technologies today. They have proven to be very suitable for text-based documents. When used for visual documents several problems arise, the most important being the lack of standard naming conventions and the uneven visual training among the human catalogers. A direct effect is reflected in the confusing results retrieved in some multimedia systems. Finding mappings between low level image features which are automatically extracted and textual descriptors would partially solve some of the naming inconsistencies.

In our experiments we focused our attention to several “keyword” categories in which color information has a high importance: Sunset (254/2100), Marine (619/2100), Arid images (296/2100), Nocturne (92/2100), Landscape (153/2100). In parenthesis is a “close” evaluation of the distribution of those type of keywords on the image database used for testing. In our experiments we looked to find mappings between these semantic concepts and low-level color features which had a reasonable (more than 70%) classification accuracy.

### 3. MINING ASSOCIATION RULES

Clustering is a discovery process in data mining. It groups a set of data in a way that maximizes the similarity within clusters and minimizes the similarity between two different clusters. The discovered clusters can explain the characteristics of the underlying data distribution and serve as foundation for other analysis techniques [12]. In the following sections we present the two techniques we used to induce mappings between high-level semantic concepts and low-level features associated with images.

#### 3.1. Decision trees

Decision trees are hierarchical, sequential classification structures that recursively partition a set of observations (data), thus representing the rules underlying the data. The process of automatic construction of rules in the form of decision trees has been attempted in several disciplines. In general, a decision tree may be used for exploring data in any of the following ways: To uncover mappings from independent to dependent variables that can be used to predict the value of the dependent variable in the future. To reduce the volume of data to a more compact form which provides an accurate summary by preserving the main characteristics. To discover if the data contains well-separated clusters of objects, so they could be interpreted meaningfully in the context of a substantiated theory. A brief exposition of decision tree induction is presented as follows.

Consider a set of *objects*, each of them completely described by a set of *attributes* and a *class label*. A *tree* is a rooted connected acyclic graph, consisting of a set of *internal* nodes (denoted by ovals in a graphic representation) and a set of *leaf* nodes (denoted by rectangles). A *decision tree* is a tree induced on a *training set*, which consists of objects. In a decision tree, a *test (split)* is associated to every internal node. Such tests are logical expressions involving the object's attributes. Each *edge* from an internal node *T* to its children is labelled with a distinct outcome of the test at node *T*. A *class label* is associated to each leaf node. The number of classes is finite. When classifying an example, the role of an internal node is to test the value of the expression based object's attributes, and to send it "down" the corresponding edge. We call a leaf node *pure* if all the training examples at that node belong to the same class. Decision trees are divided into two categories: *univariate decision trees* in which the tests at each internal node involve only one attribute, and *multivariate decision trees* in which the split expressions contain multiple attributes.

The process of building a tree from a set of training samples is called *tree induction*. Most of tree induction systems proposed in the literature follow a greedy top-down approach. They start with an empty tree and the entire training set, and the following algorithm is performed until no splits of the nodes are possible: 1) At the current node *T*, verify if all corresponding training examples belong to the same class *c*. If YES, create a leaf node, label it with the class *c* and Stop. 2) If NO, find the set of all possible splits  $\Sigma$ , and rank each split using a *impurity measure*. 3) Choose the best split  $\sigma$  as the test for the current node, and for each distinct outcome of  $\sigma$  create a child node. 4) Label each edge between the current node and the child nodes with outcomes of  $\sigma$ . Using the outcomes of the test  $\sigma$  partition the training data corresponding to the current node into the child nodes. This approach can be easily implemented through a recursive function. On the other hand, difficulties arise in the way one choses the "best split", determines the "right" size of the tree, and extracts and represents the rule set.

One of the advantages of decision trees methodology is their usefulness in building knowledge based systems. Compared with other intelligent methods, an induced decision tree can be easily translated into a collection of rules (rule base) that can be furthermore integrated into an expert system for automatic decision making. Since every node in the tree can produce a potential rule, we can evaluate the usefulness of the rules before introducing them in the expert system. Moreover, rule bases derived from several induced decision trees can be integrated into the same system.

In general a rule is an expression of the following form:

**IF  $P_1$  AND ... AND  $P_l$  THEN *Conclusion*,**

where

- $P_i$  are of type (*attribute,value*), where in the case of continuous variables *value* is represented by an interval. The conjunction of  $P_1 \dots P_l$  form the *premise* of the rule.
- *Conclusion* represents the class we want to predict.

For future reference we denote the  $i^{th}$  rule as the tuple  $R_{i,k} = (\Pi_i, y_k)$ , where  $\Pi_i$  represents the premise of the  $i^{th}$  rule and  $y_k$  the class label in the conclusion.

When evaluating a rule we have to look for two properties [9]. **Generality:** If the premise is not too restrictive the rule will tend to cover a large number of objects for which we have to associate a conclusion. **Precision:** If the premise is restrictive it will cover just a small fraction of the population, thus assuring a small error rate in classification. There are several measures defined to evaluate the rules [14]:

Measures based on local evaluation functions: They take into consideration only the information brought by the subset described by the premise. In this category we consider the measures based on the theory of statistical estimation (i.e. estimated accuracy) and the ones based on theory of information (i.e. Shannon’s entropy).

Measures based on the complexity of the rule description: These measures try to find a tradeoff between the cost of the rule’s description and the number of rule’s exceptions.

Measures based on the initial class distributions: These measures describe not only the difference in class distributions within the different subsets of objects, but also their relative cardinalities. (i.e. j-measure [9]).

When evaluating rules derived from the decision trees we considered the following measures:

**Estimated Accuracy:** In this approach the assumption is that a rule is considered as being better if the probability of incorrect classification is smaller or if the probability of correct classification is larger. In order to evaluate a rule one has to compute either one of these probabilities. Estimations often used for these probabilities are the relative frequencies. The estimated accuracy is  $E(R_i) = n_{ki}/n_i$ , where  $n_{ki}$  represents the number of objects from the training set covered by the premise  $\Pi_i$  and classified as being in class  $y_k$ , and  $n_i$  is the total number of objects covered by the premise  $\Pi_i$ .

**J-measure:** Proposed by Goodman and Smyth [9], the J-measure captures the average decrease in number of bits necessary to predict a class given the a priori probability distribution  $P(Y = y_k)$ , and the a posteriori probability distribution  $P(y = y_k/\Pi_i)$ . The ability to predict a class  $k$  given the premise  $\Pi_i$  is estimated by

$$j(y_k, \Pi_i) = \frac{n_{ki}}{n_i} \log \frac{\frac{n_{ki}}{n_i}}{\frac{n_k}{n}} - \frac{n_i - n_{ki}}{n_i} \log \frac{\frac{n_i - n_{ki}}{n_i}}{\frac{n - n_k}{n}} \quad (1)$$

where  $n$  is the total number of samples in the training set,  $n_i$  is the total number of objects covered by the premise  $\Pi_i$ ,  $n_k$  is the number of objects with the label  $y_k$ , and  $n_{ki}$  represents the number of objects from the training set covered by the premise  $\Pi_i$  and classified as being in class  $y_k$ . The J-measure is defined as  $J(R_i) = P(\Pi_i) \times j(y_k, \Pi_i)$ , where  $P(\Pi_i)$  is computed as the ratio  $n_i/n$ .

### 3.2. K-means Clustering

We use a variation of k-means clustering to build a hierarchy of clusters. At every level of the hierarchy, the variation of k-means clustering uses a non-Euclidean similarity metric and the cluster prototype is designed to summarize the cluster in a manner that is suited for quick human comprehension of its components. The resultant clusters are further divided into other disjoint sub-clusters performing organization of information at several levels, going for finer and finer distinctions. The results of this hierarchy decomposition are represented by a tree structure in which each node of the tree represents a cluster prototype and at the last level, each leaf represents an image. The hierarchy of the cluster prototypes allows an organized browsing of the entire image collection.

This adaptation of k-means algorithm is required since the color triplets (hue, saturation, and lightness) derived from RGB space by non-linear transformation, are not evenly distributed in the HSL space; the representative of a cluster calculated as a centroid also does not make much sense in such a space. Instead of using the Euclidean distance, we define the measure as in (6) so the distance between two color triplets is a better approximation to the difference perceived by human.

If  $q_i$  and  $t_i$  represent the  $i$ th component in the feature vectors corresponding to two images (Q) and (T), respectively, we denote by  $(h_{q_i}, s_{q_i}, l_{q_i})$  and  $(h_{t_i}, s_{t_i}, l_{t_i})$  the dominant hue-saturation-lightness triplet for the block  $i$  in the image (Q) and (T), respectively. The block similarity is defined by

$$S(q_i, t_i) = [1 + a \times D_h(h_{q_i}, h_{t_i}) + b \times D_s(s_{q_i}, s_{t_i}) + c \times D_l(l_{q_i}, l_{t_i})]^{-1} \quad (2)$$

where  $a$ ,  $b$ , and  $c$  are positive real valued constants that are selected to define the relative importance of hue, saturation and lightness in similarity calculations and  $D_h$ ,  $D_s$ , and  $D_l$  represent the functions that measure similarity in hue, saturation and lightness as defined in (3), (4), and (5).

$$D_h(h_{q_i}, h_{t_i}) = [1 - \cos^k(2\pi\|h_{q_i} - h_{t_i}\|/256)] / 2 \quad (3)$$

$$D_s(s_{q_i}, s_{t_i}) = \|s_{q_i} - s_{t_i}\|/256 \quad (4)$$

$$D_l(l_{q_i}, l_{t_i}) = \|l_{q_i} - l_{t_i}\|/256 \quad (5)$$

The similarity between a query and a target image is computed as

$$S(Q, T) = \frac{\sum_{i=1}^{M*N} b_i S(q_i, t_i)}{\sum_{i=1}^{M*N} b_i} \quad (6)$$

where  $b_i$  stands for the masking bit for block  $i$  (if we intend to perform a partial matching and ignore some of the blocks; default value is 1), and  $M * N$  is the number of blocks.

We define the cluster prototype to be the most similar image to the other images from the corresponding cluster; in another words, the cluster representative is the *clustroid* point in the feature space, i.e., the point in the cluster that maximizes the sum of the squares of the similarity values to the other points of the cluster. If  $C$  is a cluster, its clustroid  $M$  is expressed as:

$$M = \arg(\max_{I \in C} \sum_{J \in C} S^2(I, J)), \quad (7)$$

where  $I$  and  $J$  stand for any two images from the cluster  $C$  and  $S(I, J)$  is their similarity value. We use  $\arg$  to denote that the clustroid is the argument (image) for which the maximum of the sums is obtained.

The partition of the data into a specified number of clusters  $K$ , is built using a splitting criterion that finds the optimal partition as the one that maximizes the criterion of the sum-of-squared-error function:

$$J_e(K) = \sum_{k=1}^K w_k \sum_{I \in C_k} S^2(I, M_k), \quad (8)$$

with  $w_k = 1/n_k$ , where  $M_k$  and  $I$  stand for the clustroid and any image from cluster  $C_k$ , respectively,  $S(I, M_k)$  represents the similarity value between  $I$  and  $M_k$ , and  $n_k$  represents the number of elements of cluster  $C_k$ . The reason of maximizing the criterion function comes from the fact that the proximity index measures the similarity; that is, the larger a similarity index value is, the more two images resemble one another.

Once the partition is obtained, in order to validate the clusters, i.e. whether or not the samples form one more cluster, several steps are involved. First, we define the null hypothesis and the alternative hypothesis as follows:  $H_0$ : there are exactly  $K$  clusters for the  $n$  samples, and  $H_A$ : the samples form one more cluster. According to the Neyman-Pearson paradigm [15], a decision as to whether or not to reject  $H_0$  in favor of  $H_A$  is made based on a statistics

$$T(n) = J_e(K)/J_e(K+1). \quad (9)$$

The statistic is nothing else than the cluster validity index that is sensitive to the structure in the data. To obtain an approximate critical value for the statistic, that is the index is large enough to be ‘unusual’, we use a threshold that takes into account that, for large  $n$ ,  $J_e(K)$  and  $J_e(K+1)$  follow a normal distribution. Following these considerations, we consider the threshold  $\tau$  defined in [5] as:

$$\tau = 1 - 2/(d\pi) - \alpha[2(1 - 8/(d\pi^2))/(nd)]^{1/2}. \quad (10)$$

The rejection region for the null hypothesis at the  $p$ -percent significance level is  $T(n) < \tau$ .

The parameter  $\alpha$  in (10) is determined from the probability  $p$  that the null hypothesis  $H_0$  is rejected when it is true and  $d$  is the sample size. The last inequality provides us with a test for deciding whether the splitting of a cluster is justified.

The mapping between the color features and text features is based on two things: domain semantics and statistical properties of low-level features. For each cluster we define their *optimal textual characterization* as the keyword that maximizes the following function:

$$F(keyword) = \frac{f(keyword)}{\sum_{keyword_i \in T} f(keyword_i)}, \quad (11)$$

where  $f(keyword)$  denotes the number of times a keyword appears in a given cluster  $C$  and  $T$  is the set of keywords that characterizes  $C$ .  $F(keyword)$  is a number between 0 and 1, and the effect of this normalization is to disregard

the sizes of the clusters.  $F(\text{keyword})$  measures the relative importance of a keyword compared to the other keywords occurring in that cluster.

Given a cluster  $C$  containing  $n$  images and its *clustroid* (see (7))  $M$  we compute the *radius* in each dimension analog to the standard deviation in the Euclidean space:

$$\text{radius}_i = (w \sum_{I \in C} (I_i - M_i)^2)^{1/2}, i = 1, \dots, 3 \times M \times N \quad (12)$$

where  $I_i$  and  $M_i$  stand for the  $i$ th component of the feature vector of image  $I$  and clustroid  $M$ , respectively and  $w = 1/n$ . We define the *relevance region*  $R$  of the cluster  $C$  as the region that includes the images whose feature vector components are less than  $\beta \times \text{radius}$  away from the corresponding component of the clustroid. The factor  $\beta$  is determined experimentally.

The *mapping function*  $MAP$  associates to the *relevance region* of a cluster  $C$  its *optimal textual characterization*. In order to provide a computationally feasible method to automatically tag new images added to the image database we perform a dimensionality reduction operation. First, the features are ordered in increasing order of their standard deviation and the ones with the lowest values are selected. The number of selected features is chosen such that the semantic indexing accuracy in the reduced space will be almost as good as the accuracy in the original space. The mapping function for a cluster  $C$  can be expressed as IF-THEN rules in the reduced space as:

**IF**  $\text{feature}_{\sigma_1} \in (M_{\sigma_1} - \beta \times \text{radius}_{\sigma_1}, M_{\sigma_1} + \beta \times \text{radius}_{\sigma_1})$  **AND**  $\text{feature}_{\sigma_2} \in (M_{\sigma_2} - \beta \times \text{radius}_{\sigma_2}, M_{\sigma_2} + \beta \times \text{radius}_{\sigma_2})$  **AND** ... **AND**  $\text{feature}_{\sigma_v} \in (M_{\sigma_v} - \beta \times \text{radius}_{\sigma_v}, M_{\sigma_v} + \beta \times \text{radius}_{\sigma_v})$  **THEN** *keyword*,  
 where  $\sigma$  stands for the permutation that gives the indices of the ordered low-level features,  $v$  stands for the number of the first top selected features, and *keyword* is the most frequent keyword in cluster  $C$ .

## 4. PERFORMANCE EVALUATION

In this section we present some results to show the performance of the two approaches. These rules are based on a database of 2100 images. The features vectors extracted from the images are stored in a Microsoft Access Database. From each image we extracted both the quantized global histogram and the fixed-image partitioning representation.

### 4.1. DecisionTree Rule Induction

Based on the images in the database the following quantization scheme was obtained for the global histogram:

- Hue: (H1, H2, H3, H4, H5, H6, H7, H8) = (0, 25, 41, 61, 138, 200, 213, 241, 359)
- Saturation: (S1, S2, S3, S4, S5, S6, S7, S8) = (0, 9, 16, 24, 33, 42, 56, 80, 100)
- Lightness: (L1, L2, L3, L4, L5, L6, L7, L8) = (0, 13, 27, 38, 47, 56, 65, 78, 100)

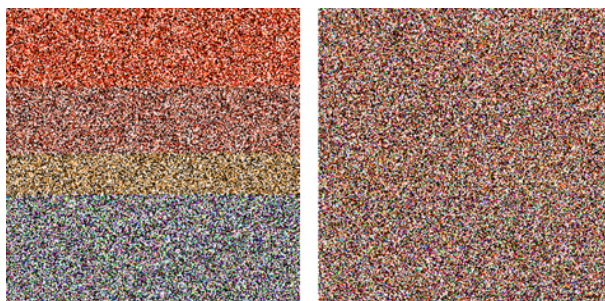
For each image we create a feature vector with 24 components: The first eight represent the relative frequencies (i.e. number of pixels in the specified bin divided by the total number of pixels in the image) for the hue component with respect to the above hue quantization scheme, next eight entries are for the saturation and the last eight for lightness. The feature vector was then augmented with the class association component. As we mentioned before the keywords considered were: Sunset, Marine, Arid images, and Nocturne. The decision tree was induced using the CART[Twoing] option from the SIPINA software package [18]. This software package allows experimentation with different types of decision tree induction methodologies (C4.5, CART, ID3, ChAID, SIPINA, QRMDL, WDTaiqm). All these decision trees induction methodologies were successfully applied to this problem. The details of these methods and a comparison of results are found in [3] and [4]. We designed the following experiment: for each of the semantic concepts we performed 5 trials. In each trial, a subset containing 65%-70% of the feature vectors was randomly sampled for the tree induction step and the remaining data was used for the evaluation step. The stopping rule was based on a combination between a restriction on the minimal size of a vertex and the p-value of  $\chi^2$  test that insures that the distribution of classes in a leaf is different to the distribution in the whole learning sample. At the end a pruning step was performed on the trees induced to reduce their size. Table 1 shows the selected attributes from the feature vector used in the rule bases generated for the different keywords.

For exemplification, we present one rule set derived for the ‘‘Sunset’’ keyword. The hypercube determined by the H1, H2, S6, S8 and L8 attributes is partitioned into six regions (each region has a rule associated). There are three regions which will assign the annotation ‘‘Sunset’’ to an image and they are determined by the following rules:

**Table 1.** CART classification

Keyword	Attributes
Sunset	( H1, H2, S6, S8, L8)
Arid	(H1, H2, H3, S8, L7)
Marine	(H1, H3, H6, L8)
Nocturne	( L1)

**R4:** If an image has more than 49% of pixels with hue values between 0 and 25, has less than 14% of pixels with hue values between 25 and 41, and has more than 26.5% of pixels with saturation values between 80 and 100 (in other words most of the colors are saturated) then the image is considered “Sunset” (with an estimated accuracy of 0.89%). Figure 2 illustrates the color distribution of those images which will be classified as “Sunset” by this rule. A subset of such images is presented in Figure 3.



**Figure 2.** “Sunset” keyword association: Visual representation for Rule 4. The image on the left illustrates an approximation of the relative frequencies imposed by the rule. The image on the right is the randomized representation.

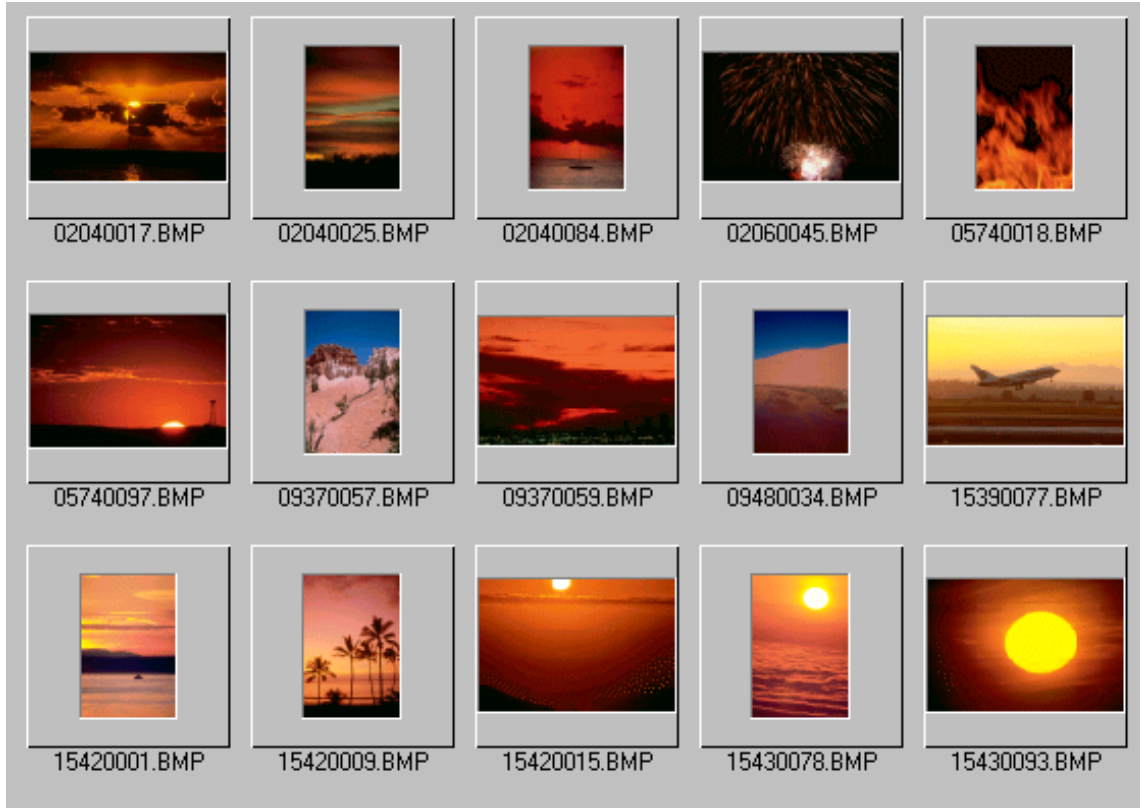
**R5:** If an image has more than 14% of pixels with hue values between 25 and 41, has less than 7.5% of pixels with saturation values between 42 and 56, has more than 26.5% of pixels with saturation values between 80 and 100, and has more than 3.5% of pixels with lightness values between 78 and 100 then the image is considered “Sunset” (with an estimated accuracy of 0.92%).

**R6:** If an image has more than 14% of pixels with hue values between 25 and 41, has more than 26.5% of pixels with saturation values between 80 and 100, and has less than 3.5% of pixels with lightness values between 78 and 100 then the image is considered “Sunset” (with an estimated accuracy of 0.85%).

The remaining regions will categorize the image as not belonging in the “Sunset” class.

## 4.2. K-Means Rule Induction

In the case of the fixed-image partitioning representation, we divide each image from the collection into  $8 \times 8$  overlapping blocks and we compute the area-peak values for the hue, saturation and lightness for each block, each value scaled to the  $[0, 255]$  interval. Therefore, each image has associated a feature vector with 192 components: The first 64 numbers represent the Hue, the next 64 Saturation, and the last 64 the Lightness values, respectively. The corresponding feature vectors from the 2100 images are randomly split in two sets: training set (67% out of 2100 images) and test set (33%). The training set was used to learn the mappings between the low level features and textual descriptors as described in Section 3.2. First, we applied the modified k-means algorithm to derive a two-level hierarchy of clusters (with 30 clusters on the first level and 70 on the second), and the cluster validity was checked for every cluster. The values of the constants  $a$ ,  $b$  and  $c$  in formula (2) are experimentally chosen as being 2.5, 0.5 and 3, respectively. In the rule extraction step, for each cluster from the second level, the components of the feature vectors were ordered in increasing order of standard deviations, and the most frequent keywords were calculated. Keywords whose meanings are related to color information were the most frequent keywords present in



**Figure 3.** A set of images classified as “Sunset” by Rule 4 (CART[Twoing]).

**Table 2.** Rule Sizes and Accuracy for ‘Sunset’ Classification in the K-means approach

Cluster ID	No. of Attributes	Classification Accuracy
28.1	35	77.78%
13.1	20	71.43%
18.2	140	70.59%

the clusters.

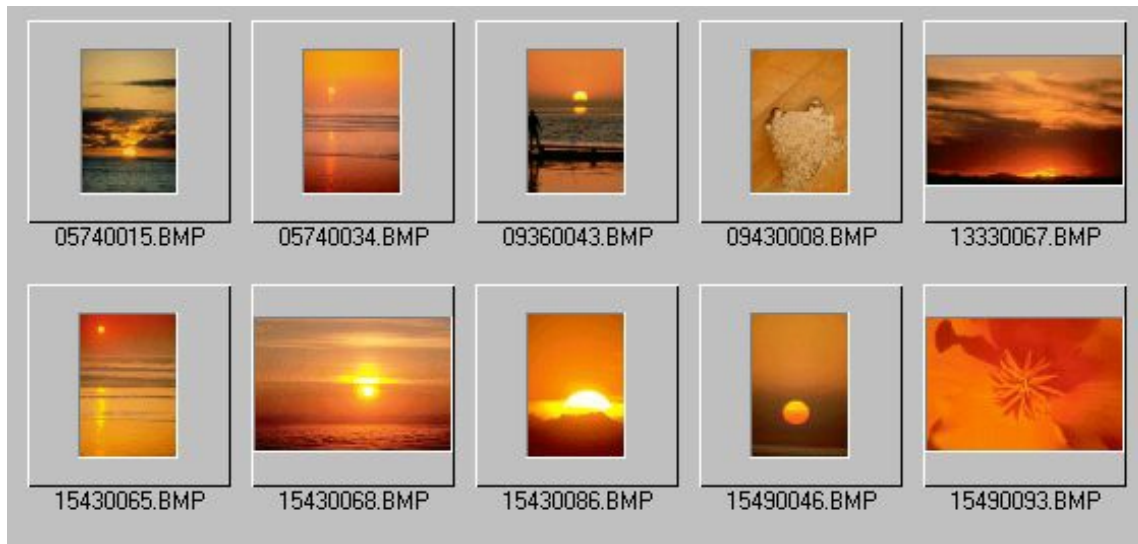
For illustration we present results from the clusters in which the optimal textual characterization was ‘Sunset’ (See Table 2) and ‘Landscape’ (See Table 3). The ‘Sunset’ keyword covers three clusters while ‘Landscape’ is representative in four clusters.

The rule derived for Cluster 28.1 that is semantically characterized by the keyword ‘Sunset’ (with  $v = 35$  and  $\beta = 2$ ) has the precision of 77.78% for the training set and 100% on the testing set. Figure 4 presents some sample images labeled as ‘Sunset’ by this rule.

For Cluster 23.2, whose optimal textual characterization is ‘Landscape’ the rule (with  $v = 80$  and  $\beta = 2$ ) has

**Table 3.** Rule Sizes and Accuracy for ‘Landscape’ Classification in the K-means approach

Cluster ID	No. of Attributes	Classification Accuracy
14.2	80	71.43%
23.2	80	88.24%
21.7	30	62.50%
2.2	30	100.00%



**Figure 4.** A set of images classified as “Sunset” by the rule corresponding to Cluster 28.1 .

the precision of 88.24% for the training set and 100% on the testing set. Some sample images labeled as ‘Landscape’ by this rule can be seen in Figure 5.



**Figure 5.** A set of images classified as “Landscape” by the rule corresponding to Cluster 23.2 .

## 5. SUMMARY AND CONCLUDING REMARKS

In this paper, we presented two approaches to automatic annotation of images. One method is based on extracting association rules from a set of decision trees induced over the set of low-level features extracted from the images in collection. This method yields a set of association rules which have a reasonable classification error. The rule sets are small and the number of features involved is relatively small. This is an advantage especially when we have to

evaluate them in order to integrate them in the Knowledge Base. On the other hand, decision tree induction is a form of supervised learning which require that for the initial training set (the set based on which the decision tree is induced and pruned) to be manually annotated. Because the manual annotation has a subjective side which has to be taken into consideration, the classification error might be higher. The manual annotation error can be reduced by including in the Image Database System an option for the user's input with respect to the accuracy of the keywords assignment for a given image. Using a 'majority-wins' approach one could induce a new tree and derive a new set of rules based on the current classification.

The other method provides a different approach. First, through an unsupervised learning step, the images are arranged in clusters based on their low-level features. The supervised step consists in determining for each cluster which is the optimal textual characterization. The nice part is that one has to look only at a reduced set of images, so this assignment can be relatively accurate. And the results show that this approach has a good classification accuracy for the images assigned to a particular cluster. On the other hand, through the initial k-means clustering step many images might not receive any keyword assignment by being assigned to clusters where their semantic meaning is different than the optimal textual characterization and their features not verifying the rule generated for the cluster. Consequently, out of approximatively 250 'Sunset' images in the collection only 21 are in clusters with an optimal textual characterization 'Sunset' and only 13 are covered by the rules derived from these clusters. However, more images will receive a keyword assignment if the rules will be derived for a lower level of the hierarchy of clusters. By traversing down the hierarchy, finer and finer details are obtained for the image database and so, it is more likely that cluster semantic meanings will be the same with their optimal textual characterizations.

## REFERENCES

1. M. Ben-Bassat, "Use of distance measures, information measures and error bounds on feature evaluation," in *Classification, Pattern Recognition and Reduction of Dimensionality, Volume 2 of Handbook of Statistics* Eds. P. R. Krishnaiah and L. N. Kanal, ed., pp. 773-791, North-Holland Publishing Company, 1987.
2. N. S. Chang and K. S. Fu, "Query-by-Pictorial example," in *IEEE Transactions on Software Engineering*, SE-6(6), pp. 519-524, 1980.
3. I. Coman and I. K. Sethi, "Color Features and High Level Concepts", Preprint 2000.
4. I. Coman, *Algorithms for Efficient Management and Retrieval of Visual Documents*, Ph.D. Thesis, Wayne State University, 2000.
5. R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons, Inc., 1973.
6. C. Faloutsos, M. Flickner, W. Niblack, D. Petkovic, W. Equitz, R. Barber, "Efficient and Effective querying by image content," *Technical report, IBM Research Report*, 1993.
7. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by Image and Video Content: The QBIC system," in *IEEE Computer*, 28(9), pp. 23-32, 1995.
8. J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, 2nd Ed. in C, 1996.
9. R. M. Goodman and P. Smyth. Information-theoretic rule induction. In *Proceedings of European Conference on Artificial Intelligence*, 1988.
10. R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Co., 1992.
11. V. N. Gudivada, V. V. Raghavan and K. Vanapipat, "A Unified Approach to Data Modeling for a Class of Image Database Applications," *IEEE Transactions on Data and Knowledge Discovery*, 1994.
12. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentices Hall Advanced Reference Series, 1998.
13. A. Pentland, R. W. Picard, S. Sclaroff, "Photobook: Content-based manipulation of image databases," in *Multimedia Tools and Applications*, editor Borko Furht, Kluwer Academic Publishers, Boston, pp. 43-80, 1996.
14. R. Rakotomalala, *Graphes d'Induction*, PhD Thesis, University Claude Bernard, Lyon 1, 1997.
15. J. A. Rice, *Mathematical Statistics and Data Analysis*, Duxbury Press, 1995.
16. I. K. Sethi and I. Coman, "Image retrieval using hierarchical self-organizing feature maps," in *Pattern Recognition Letters* 20(1999):1337-1345.
17. I. K. Sethi, I. Coman, B. Day, F. Jiang, D. Li, J. Segovia-Juarez, G. Wei, B. You. COLOR-WISE: A System for Image Similarity Retrieval Using Color. In *Proc. of IS&T/SPIE Symposium on Electronic Imaging: Storage and Retrieval for Image and Video Databases VI*, San Jose, CA, Jan 25-30, 1998.

18. *SIPINA for Windows*. ERIC Laboratory, University of Lyon 2,  
<http://eric.univ-lyon2.fr/ricco/sipina.html> .
19. J. R. Smith and S.-F. Chang, "Single color extraction and image query," in *Proc. IEEE Int. Conf. on Image Proc.*, 1995.
20. D. A. Zighed, J. P. Auray and G. Duru, *SIPINA: Méthode et logiciel*, *Lyon Lacassagne*, 1992.