1-2007

Integrating semantic templates with decision tree for image semantic learning

Ying LIU

Dengsheng ZHANG

Guojun LU

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Integrating Semantic Templates with Decision Tree for Image Semantic Learning

Ying Liu[1], Dengsheng Zhang[2], Guojun Lu[2], and Ah-Hwee Tan[1]

[1] Emerging Research Lab, School of Computer Engineering,
Nanyang Technological University, Singapore, 639798
{liuying, asahtan}@ntu.edu.sg
[2] Gippsland School of Computing and Information Technology,
Monash University, Vic, 3842, Australia
{dengsheng.zhang, guojun.lu}@infotech.monash.edu.au

Abstract. Decision tree (DT) has great potential in image semantic learning due to its simplicity in implementation and its robustness to incomplete and noisy data. Decision tree learning naturally requires the input attributes to be nominal (discrete). However, proper discretization of continuous-valued image features is a difficult task. In this paper, we present a decision tree based image semantic learning method, which avoids the difficult image feature discretization problem by making use of semantic template (ST) defined for each concept in our database. A ST is the representative feature of a concept, generated from the low-level features of a collection of sample regions. Experimental results on real-world images confirm the promising performance of the proposed method in image semantic learning.

Keywords: Decision tree, Image semantic learning, Semantic template, Image feature discretization.

1 Introduction

In order to reduce the 'semantic gap', many algorithms have been proposed to associate low-level image features with high-level semantics. Machine learning tools such as SVM, Neural Networks and Bayesian classification are often used for image semantic learning [1,2,3]. On the other hand, some other researchers have found in their experiments that decision tree learning such as ID3, C4.5 and CART are mathematically much simpler and perform well in concept learning for image retrieval purposes. Decision tree learning is an extensively researched solution to classification tasks and has great potential in image semantic learning. Compared with other learning methods, decision tree learning is not only simpler but also robust to incomplete and noisy input features [4,5].

The difficulty in applying decision tree induction to image semantic learning lies in the difficulty in image feature discretization which is a challenging task [6]. To benefit image learning process, discrete image feature values should correspond to meaningful conceptual names. Algorithms like ID3 [4] require the value of the input attributes to be discrete. Some algorithms have been designed to handle continuous-valued attributes [7,8]. For example, C4.5 [7] uses a minimal entropy heuristic to find

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4352, Part II, pp. 185 – 195, 2007.

binary-cuts for each attribute in order to discretize continuous attributes. Based on the understanding that multi-interval discretization could be better than only binary discretization and that it can lead to more compact decision trees, entropy-based multi-interval discretization method is introduced to find multi-level cuts for each attribute [8]. However, these generally designed algorithms usually do not provide meaningful quantization of image feature space. It has been reported that although C4.5 can handle continuous attributes, it does not work as well in domains with continuous attribute values as in domains with discrete attribute values [9].

To avoid the difficult image feature discretization problem in decision tree learning, we propose to convert low-level color (texture) features into color (texture) labels by making use of the semantic templates defined for each concept in the database. These semantic templates (STs) are integrated with decision tree (DT) learning to form the proposed algorithm, which is applied to an RBIR system to implement image retrieval with high-level semantics.

The remaining of the paper is organized as follows. In Section 2, we explain semantic template generation and low-level image feature discretization. Section 3 provides the details of the proposed algorithm. The experimental results are given in section 4. Finally, Section 5 concludes this paper.

2 Semantic Templates and Image Feature Discretization

Our purpose in building a decision tree is to associate the low-level features of image regions with high-level concepts. For our natural scenery image database, the following 19 concepts (classes) are selected: grass, forest, blue sky, sea, flower, sunset, beach, firework, tiger, ape fur, eagle, building, snow, rock, bear, night sky, crowd, butterfly and mountain. Each of these 19 concepts is given a concept label from 0,1,…, to 18 in sequence. The input attributes of the decision tree are the low-level region features and the output is one of the 19 concepts.

This section first describes the input attributes used in our algorithm, which are the low-level region features including color and texture. Then, we explain how to construct semantic template (ST) for each concept and how to make use of ST for image feature discretization.

2.1 Low-Level Image Features

In our system, each database image is segmented into different regions using JSEG [10]. For each region in the database, color feature and texture feature are extracted as its low-level features. The color feature we use is the HSV space dominant color as described in [11]. Gabor texture feature of each region is obtained using the POCS-ER algorithm proposed in [12]. Each dimension of the color and texture features is normalized to the range [0,1].

2.2 Semantic Templates Construction and Image Feature Discretization

The low-level color and texture features have to be discretized to be useful in decision tree learning, as they have continuous values. As explained before, proper discretization of continuous image features is still an open challenge. To avoid this problem, we propose to convert continuous color (texture) features into color (texture) labels by introducing semantic templates (STs),.

In order to construct STs, firstly, we collect a set of 40 sample regions from the database for each of the 19 concepts defined and form a training data set of 760 regions in total. For every concept, a semantic template (ST) is defined as the centroid of the low-level features of all the 40 sample regions. For the j_{th} sample region in class *i*, where $(j=0,...,39)$ and $(i=0,...,18)$, its color and texture features are given by: $\{h_j^i, s_j^i, v_j^i\}$ (dominant color in HSV space) and $\{\mu_{00_j}^i, \sigma_{00_j}^i, ..., \mu_{35_j}^i, \sigma_{35_j}^i\}$ (Gabor feature with 4 scales and 6 orientations) respectively.

Taking the first dimension of color feature and the first dimension of texture feature as examples, the centroid of the color and texture features can be calculated as:

$$\begin{aligned} \overline{h}^i &= \frac{1}{40}\sum_{j=0}^{39} h_j^i \\ \overline{\mu}_{00}^i &= \frac{1}{40}\sum_{j=0}^{39} \mu_{00_j}^i \end{aligned} \tag{1}$$

Thus, we obtain a set of 19 STs, denoted as $ST_i = \{C_i, T_i\}$ with *i*=0,…,18. where $C_i = \{\overline{h}^i, \overline{s}^i, \overline{v}^i\}$ and $T_i = \{\overline{\mu}_{00}^i, \overline{\sigma}_{00}^i, ..., \overline{\mu}_{35}^i, \overline{\sigma}_{35}^i\}$ are the 'representative' color and texture features of concept *i*. We refer to these representative color and texture features as *color-template* and *texture-template* respectively.

By calculating the Euclidean distance between the color (texture) feature of a region and the color (texture) template of each concept, the color (texture) label of the region can be obtained, which is the label of the concept corresponding to the minimum distance. Taking color feature as example, the following algorithm explains how to obtain the color label for the j_{th} sample region in class *i*.

1) Calculate the distance between the color feature of this region and the color-template of each concept *m* (*m*=0,…,18) as

$$d_{c(j,i)}^m = \sqrt{(h_j^i - \overline{h}^m)^2 + (s_j^i - \overline{s}^m)^2 + (v_j^i - \overline{v}^m)^2} \tag{2}$$

2) Find the minimum distance, $d_{c(j,i)}^{m_{\min}} = \min d_{c(j,i)}^m$;

3) The color label of this region is: $m_{\min}$

As a result, each region is represented by its color label, texture label. Both color and texture labels have 19 possible values each (0,…,18). In this way, by making use of semantic templates, the continuous low-level image features are converted to discrete values. This discretization process is different from conventional methods which try to quantize each dimension of the image feature into different intervals. Unlike other methods which are designed for general purpose applications, our method is custom built for discretizing image features. In addition, the proposed method is computationally simple and easy to implement. To discretize an attribute, the proposed method needs only to compute the Euclidean distance between a region feature and each of the 19 semantic templates, in order to get the concept label that corresponds to the minimum of all the distances. The method used in C4.5 requires the computation of the entropies introduced by all possible partition boundaries to find the binary discretization boundary which corresponds to the minimum entropy.

Then, for each of the two partitions, the above process is repeated until the stop condition is achieved. Moreover, the above process has to be applied to each dimension of the image feature! Thus, in order to be used in our database, this algorithm has to be performed on each dimension of the 3D color feature and 48D texture feature. The multi-interval discretization method is even more computationally expensive [8].

3 The Proposed Algorithm

A decision tree can be obtained by splitting the training data into different subsets based on all the possible values of an attribute. The root node contains all the instances in the training set. Then, an input attribute is selected to split the training set into different subsets corresponding to different possible values of the attribute. This process is recursively applied to each derived subset up to the leaves of the tree. There are different ways to split the data resulting in different trees. To obtain small tree, a key issue is to select the most significant attribute at each level of the tree from those attributes which are not yet used for decision making [13]. Information gain is the most commonly used measure to decide which attribute to test at each non-leaf node of the tree [13].

In this section, we first explain how to select the most significant attribute for decision making, and then provide the details of the learning method.

3.1 Most Significant Attribute

Decision tree learning algorithms such as ID3 and C4.5 select the attribute with greatest information gain, based on the concept of entropy. Entropy, in information theory, characterizes the (un)certainty of an arbitrary collection of examples. The higher the entropy, the more information is needed to describe the data. Given a set *S*, containing *m* possible outcomes, the entropy of set *S* is defined as:

$$H(S) = \sum_{i=1}^{m} -P_i * \log_2 P_i \tag{3}$$

where P_i is the proportion of instances in *S* that takes the i_{th} value of outcome. Information gain measures the reduction in entropy (gain in information) induced by splitting the dataset on a certain attribute *E*.

$$Gain(S,E) = H(S) - \sum_{v \in \mathrm{E}} \frac{|S_v|}{|S|} H(S_v) \tag{4}$$

where *v* is a value of *E*, $|S_v|$ is the subset of instances of *S* with *E* having value *v*, $|S|$ is the number of instances in *S*. The second term in equation (5) is the average entropy of attribute *E* over all possible values of *E*. We refer to this simply as 'entropy' of *E*. From equation (5), the attribute with greatest information gain is equivalent to the attribute with least entropy [13].

In our case, among the three input attributes: dominant color, Gabor texture feature, and spatial location; spatial location is obviously less significant for natural scenery images than color and texture features. We need to decide between color and texture which is more significant and should be used at the first level of the decision tree. In our experiments, the average entropy of color feature and texture feature are 2.06 and 1.69 respectively. It means that color feature provides more information in describing the images and should be used at the first level of the decision tree for decision making.

Now that we know color feature should be used first for decision making, the question is how to make decision? Given a region with color label *i*, should it be classified as class *i*? Or we need texture feature to make the decision which class it belongs to? To answer these questions, we propose to calculate the classification accuracy for each class using color feature and texture feature. The algorithm below explains the process of calculating the classification accuracy of color feature for each class.

1) For each class (concept), calculate the centroid of the color features of all the 40 regions as its *color-template*, $C_i = \{\bar{h}^i, \bar{s}^i, \bar{v}^i\}$ for $(i = 0,...,18)$.
2) Initialize an array as Correct_Num[*i*] = 0, to count the number of correct classifications.
3) For the j_{th} sample region in class *i*, calculate its Euclidean distance $d^{m}_{c(j,i)}$ to the *color-template* of each class, where $(m = 0,...,18)$.
4) Find the minimum distance $d^{m_{\min}}_{c(j,i)}$. If $m_{\min} = i$, then we consider this region correctly classified based on color feature, and

   Correct_Num[*i*] = Correct_Num[*i*] + 1.
5) Repeat steps 2-4 for all the sample data.
6) Obtain the probability of correct classification for each class *i*, as:

$$P_c[i] = \text{Correct\_Num}\ [i] / 40, \quad \text{for } i=0,\dots,18. \tag{5}$$

This algorithm provides the classification accuracy for each class as the probability of correct classification $P_c[i]$ using color feature. The classification accuracy $P_t[i]$ for texture feature can be calculated in the similar way. The results are given in Table 1. This table also provides the classification accuracy $P_{ct}[i]$ using both color and texture features. The results in Table 1 are obtained for the training data of 760 regions and are used to identify which feature (color, texture or the combination of both) gives the best classification accuracy for each concept and hence which template (color template, texture template, or both) should be used as the ST of a concept.

It is found that some concepts can be well represented by their color features and the use of texture feature does not increase the classification performance. Similarly, for some concepts, texture feature alone is sufficient to represent the concept with significant accuracy, while for others, the classification accuracy is higher when both color and texture features are combined. For example, for sunset, the classification accuracy is 0.925, 0.65, 0.90 using color feature, texture feature and the combination of both, respectively. In this case, color feature alone represents the sunset region with highest accuracy. On the other hand, some concepts such as *firework* are well characterized by their texture features and including color feature in this case degrades the classification performance. This is because of the fact that different firework regions have different colors but their texture patterns are similar. For some other concepts like *tiger*, both color and texture features are needed for their representation.

Table 1. Classification Accuracy Using Color, Texture and Color combined with Texture

Concept Label	Concept	Color Feature	Texture Feature	Color & Texture
0	Grass	0.65	0.7	**0.875**
1	Forest	0.475	0.775	**0.925**
2	Blue sky	0.375	0.25	**0.8**
3	Sea	0.525	0.35	**0.725**
4	Flower	0.475	**0.7**	0.55
5	Sunset	**0.925**	0.65	0.9
6	Beach	0.6	0.35	**0.85**
7	Firework	0.0	**0.65**	0.45
8	Tiger	0.4	0.55	**0.675**
9	Ape fur	**0.775**	0.125	0.775
10	Eagle	**0.875**	0.325	0.675
11	Building	0.45	0.55	**0.675**
12	Snow	**0.9**	0.45	0.9
13	Rock	**0.725**	0.325	0.575
14	Bear	0.4	0.475	**0.85**
15	Night sky	0.875	0.675	**0.95**
16	Crowd	0.45	**0.75**	0.6
17	Butterfly	**0.45**	0.075	0.4
18	Mountain	0.275	0.35	**0.45**
Average		0.558	0.478	0.716

In conclusion, which template to use to represent a concept, we should choose the one that provides the highest classification accuracy. This selection has been shown in Table 5.1 by making bold-face the highest classification accuracy for every concept.

3.2 Deriving the Decision Rules

Once the semantic template for each concept has been constructed, we generate a set of decision rules to map the low-level features of a region to a high-level concept. The input attributes to the decision tree include: color feature, texture feature. The following algorithm details the process of decision tree induction.

1) At the first level of the tree, the color feature of a region is compared with each of the 19 *color-templates*, to obtain its color label.
2) For concepts such as *sunset, ape fur, eagle, snow, rock* and *butterfly*, decisions can be made at the first level of the tree, as indicated by results in Table 1. That means, if the color label of a region is 5 (or 9, 10, 12, 13, 17), it is classified as *sunset* (or *ape fur, eagle, snow, rock, butterfly*).
3) Otherwise, texture feature is required for further decision making. There are two cases at the second level of the decision tree: if the texture label of a region is 4(7,16), then it is classified as *flower* (*firework, crowd*); for rest of the regions, both color feature and texture features have to be used in decision making. For instance, if both the color label and texture label of a region is 6, then it is classified as *beach*.

Hence, we derive a set of decision rules described in terms of color and texture labels as follows:

*__If__ the **Color label** of a region is '5(or 9,10,12,13,17)', **then** it is classified as *'sunset'*(or ape fur, eagle, snow, rock, butterfly).
*__If__ the **Color label** of a region is **NOT** {5,9,10,12,13,17} **AND** its **Texture label** is 4 (or 7,16), **then** it is classified as *'flower'*(or firework, crowd).
*__If__ both the **Color label and Texture label** of a region are 0 (or 1,2,3,6,8,11,14,15,18)', **then** it is classified as *'grass'*(or forest, blue sky, sea, beach, tiger, building, bear, dark sky, mountain).

For regions which do not belong to any of the 19 classes, we classify them as 'unknown'. To test the decision rules, a number of regions for each of the 19 concepts are collected from the database as test data. The test data consists of three testing sets of sizes 19*20, 19*25 and 19*30 that contain 20, 25 and 30 regions for each of the 19 concepts respectively. Given a test region with its color feature and texture feature, we first obtain its color label and texture label. Then, the above decision rules are applied to find out the concept of this region. Tested on the three test datasets, the average classification accuracy for the 19 classes are 0.784, 0.758 and 0.748 respectively. The overall average for the three test tests is 0.763.

In the proposed algorithm, the order in which features are selected to make decision is important to the classification outcome. The impact on classification accuracy can be seen in case of a conflict. For example, a region has color label as 9(ape fur) and texture label as 4 (flower). In this conflict, we are unable to classify this region without an ordering mechanism to specify the feature that should be given priority for decision making. Due to the limitation of low-level image feature in describing images, such conflict is unavoidable. Entropy (information gain) measures the capability of an attribute in describing the database images. In the case of such conflict, by selecting the attribute with smaller entropy (greater information gain), we intend to choose the feature which is more reliable for decision making.

3.3 Analysis

The proposed algorithm employs semantic templates to convert continuous valued image feature into discrete values. In this section, we compare the performance of the proposed method with ID3 and C4.5. The testing data are as described in Section 3.2, that is, three datasets with 19*20, 19*25 and 19*30 regions respectively.

ID3 and C4.5 are implemented using WEKA machine learning package [14]. Since ID3 does not accept continuous input values, we use color label, texture label as its input attributes. For C4.5, we use low-level color and texture features directly as input attributes, and let C4.5 discretize image features by itself. Fig. 1 provides the average of the classification accuracies for all the 19 concepts using different learning methods. The results show that the proposed method outperforms the other two. This proves that the image feature discretization method we proposed is more effective than the binary discretization method used in C4.5 for natural scenery image semantic learning.

These results provide evidence from our database to confirm the promising performance of the proposed method. However, extensive comparison of different learning methods is beyond the scope of this paper.

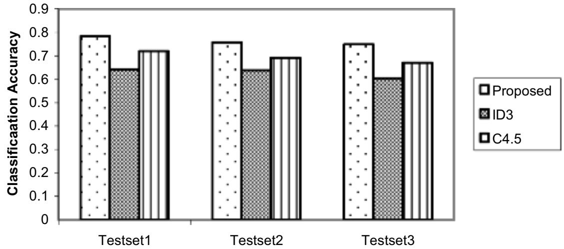

Fig. 1. Average classification accuracies for 19 concepts using different learning methods

4 Results and Analysis

4.1 Retrieval with Concepts

With the decision rules generated as in Section 3, we implement an RBIR system with high-level semantics. It is assumed that each image has at least one dominant region expressing its semantics. For most categories in our database, there is one dominant region in the relevant images. The images under a few categories contain two dominant images. For instance, all the 'firework' images in the database include regions of concepts 'firework' and 'night sky', 'bear' region always comes with 'snow' in category 'North pole bears'.

Our system supports users with query by keyword and query by specified region. That is, users can either submit a keyword as query, or specify the dominant region in the query image. In our experiments, we specify the dominant region of a query image. Using the proposed method, the concept of the specified region can be obtained. The system first finds a subset of images from the database containing region(s) of same concept as that of the query. Then, based on their low-level color and texture features, these images are ranked according to their EMD [15] distances to the query image. We refer to this process as '*retrieval by concepts*'.

4.2 Database and Performance Evaluation

We use 5,000 Corel images as our test set. These images are of 50 categories which are semantically different. 'JSEG' segmentation produces 29187 regions, an average of 5.84 regions per image. To measure the retrieval performance, we calculate the average Precision (Pr) and Recall (Re) of 40 queries with different total number of images retrieved (K=10,20,…100) and obtain the Pr~Re curve. The query images are selected from most of the 50 categories except those with very abstract labels. Example queries and the corresponding dominant regions are given in Fig. 2.

4.3 Experimental Results

The performance of our RBIR system using the proposed method for concept learning (*retrieval with concepts*) is compared with that of the RBIR system with only low-level image features (*retrieval without concepts)*. The results in Fig.3 show clearly that using high-level concepts, the retrieval performance measured by Pr~Re is improved. As examples, the retrieval results for a few queries are given in

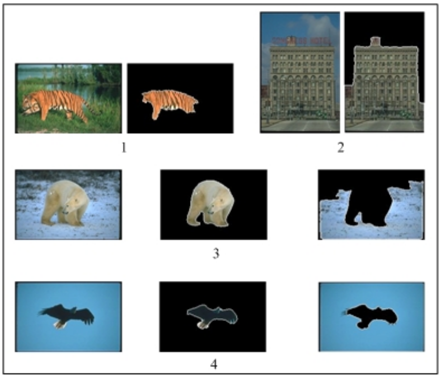

Fig. 2. Examples of query image and dominant regions

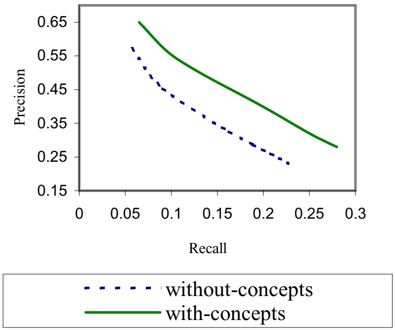

Fig. 3. Retrieval with/without concepts

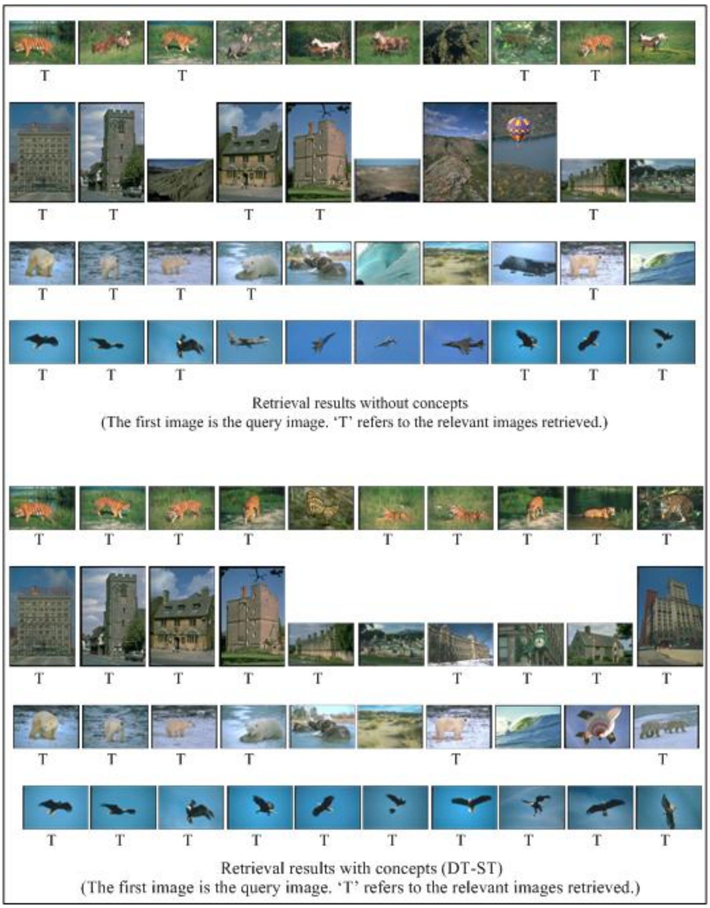

Fig. 4. Retrieval with/without high-level concepts

Fig. 4. Among the top 10 images retrieved for query 1, 2, 3 and 4, the number of relevant images found by performing '*retrieval with concepts*' is 9,9, 6, 10, respectively, whereas there are 4, 5, 5 and 6 images correctly found using '*retrieval without concepts*'.

Our experimental results prove that using the proposed learning method, the 'gap' between the user semantics and low-level image features is reduced and the retrieval accuracy is improved

5 Conclusions

In this paper, we proposed a simple and effective learning method to relate low-level image features with high-level semantic concepts, in an attempt to reduce the 'semantic gap' in content-based image retrieval. By integrating semantic templates with decision tree learning, this method avoids the difficult image feature discretization problem in normal decision tree learning. Experimental results prove that the proposed method is effective for image semantic learning.

While it is yet an open research question that which learning tool is best for image semantic learning, we proved in this paper that decision tree learning can be a strong tool to learn image semantics, if properly employed. In our future work, we will extend the proposed method to learn image concepts in other domains by using different types of image features and defining new semantic templates.

References

1. C. P. Town and D. Sinclair, "Content-based Image Retrieval using Semantic Visual Categories," Society for Manufacturing Engineers, Technical Report MV01-211, 2001.
2. E. Chang and S. Tong. "SVM Active - Support Vector Machine Active Learning for Image Retrieval," in Proc. of ACM Inter. Multimedia Conf. (2001)107-118.
3. L. Fei-Fei, R. Fergus and P. Perona. "Learning Generative Visual Models from few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," in Proc. of Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision. (2004) 178-185.
4. J. R. Quinlan, "Induction of Decision Trees," Springer Machine Leaning, (1986) 81-106.
5. M. Pal and P. M. Mather. "Decision Tree Based Classification of Remotely Sensed Data," in Proc. of 22nd Asian Conference on Remote Sensing (ACRS). (2001)245-248.
6. I. K. Sethi and I. L. Coman, "Mining Association Rules Between Low-Level Image Features and High-Level Concepts," SPIE Data Mining and Knowledge Discovery, (2001)279-290.
7. J. R. Quinlan, "C4.5, Program for Machine Learning." Morgan Kaufmann, Los Altos, Califonia, 1993.
8. P. Perner and S. Trautzsch, "Multi-Interval Discretization Methods for Decision Tree Learning," in Advances in Pattern Recognition, A. Amin, Dori D., Pudil P. and Freeman H., Editors. (1998) 475-482.
9. J. Dougherty, R. Kohavi and M. Sahami. "Supervised and Unsupervised Discretization of Continous Features," in Proc. of 12th Inter. Conf. on Machine Learning. (1995)194-202.

10. Y.Deng, B.S.Manjunath and H.Shin: Color Image Segmentation, in Proc. IEEE Computer Society Conf. on Comp. Vision and Pattern Recognition, CVPR '99, vol.2, (1999)446-451.
11. Y.Liu, D.S.Zhang, G.Lu, W.Y. Ma: Region-Based Image Retrieval with Perceptual Colors, Pacific-Rim Multimedia Conference (PCM2004), Tokyo, (2004)931-938.
12. Y.Liu, D.S.Zhang, G.Lu, W.Y. Ma: Study on Texture Feature Extraction from Arbitrary-Shaped Regions for Image Retrieval, Inter. Multimedia Modeling Conf. (MMM2006), Beijing, (2006)264-271.
13. T. Mitchell: Decision Tree Learning, in *Machine Learning*, T. Mitchell, Editor. The McGraw-Hill Companies, Inc. (1997)52-78.
14. http://www.cs.waikato.ac.nz/ml/weka/.
15. Rubner, Y., Tomasi, C., and Guibas, L.: A Metric for Distributions with Applications to Image Databases, in Proc. of IEEE Inter. Conf. on Computer Vision(ICCV), (1998)59-67.