2006 IEEE International Conference on Systems, Man, and Cybernetics October 8-11, 2006, Taipei, Taiwan

A Content-Based Painting Image Retrieval System Based on AdaBoost Algorithm

Shwu-Huey Yen, Ming-Hsien Hsieh, Chia-Jen Wang, Hwei-Jen Lin

***Abstract*—A content-based painting image retrieval (CBPIR) system based on AdaBoost is proposed. By providing query examples which share the same semantic concepts, e.g., portraits, and incorporating with relevance feedback (RF), the user can acquire the desired painting images. To bridge the gap between low-level features and semantic concepts, a large set of 4,356 features on texture and spatial arrangement of painting images is provided. Utilize the nice characteristic of AdaBoost algorithm that it can combine partial weak classifiers, i.e. features, into a strong one, the system can correctly discover a few most critical features from provided samples and search paintings sharing same features from the database. Our experiment in query of "portrait," based on 3 RFs and an average of 50 repetitions, shows an excellent performance of (approximately) 0.71, 0.84, 0.95 in Precision, Recall, and Top 100 Precision rates. The average execution time, based on 50 repetitions, required in initial query and three RF with training and classifying is approximately 1.2 seconds, thus a complete query takes less than 5 seconds in training and classifying. The system is proved to be accurate in content based image retrieval and also very efficient for on-line users.**

I. INTRODUCTION

The interest in automatic analysis of images based upon their contents has increased significantly with recent developments in digital image collections, World Wide Web, networking and multimedia. Research in content-based image retrieval (CBIR) becomes a major topic since first proposed in the early 1990s. The great deal of research work is devoted into this field: QBIC [1], Virage [2], VisualSEEK [3] etc. Using low-level features, such as color, texture, shape and layout, for CBIR is the most common approach to retrieve images. But images of dissimilar semantic content may share some common low-level features, while images of similar semantic content may be scattered in the feature space, thus CBIR based on low-level features is far from satisfactory.

In this paper, a CBPIR system is proposed for retrieving painting images of the same content, such like "portrait", "still life", "landscape", "vase with flowers", etc. The similarity among images of same content is more implicit and complicated than similarities in low-level features. For example, two portraits may be different in colors, layout, texture, in background, in position (sit, stand, or lying down), etc.; on the other hand, these two paintings all have faces, and the main figures located in the center. Therefore, the proposed system must be able to extract the complex visual similarities as well as the subtle distinctions of painting contents. In the existed CBIR systems, there are not so many systems designed specially for the painting images. For those painting images retrieving systems, most are following the trail of the traditional CBIR, using low-level features to measure the similarity among queries and images from database. Liu *et al.* [4] utilized techniques of data mining with low-level features to analyze the painting style of the artists or paintings. Kushiki *et al.* [5] proposed a content-based retrieval system, based on a combination of low-level features, to classify a collection of the Madonna, Frescoes, and Impressionist Scenes painting images.

De Bonet & Viola [6] presented a concept of "texture-of-texture" such that its measurement is based on the path of a tree of non-linear filtering operations. Each path through the tree creates a particular filter network, which responds to certain structural organization in the image. By down-sample operations and convolutions, a total of 46,875 feature values are obtained for each image. In this way, thousands of very specific features are extracted to approximate a complex visual similarity among images. To solve the problem of complex computations, Tieu *et al.* [7] used the AdaBoost Algorithm to choose twenty most important features out of 46,875 features. Considering these 20 features only, the speed of searching database can be accelerated to one million images per second. Later, Viola *et al.* [8] also applied AdaBoost learning algorithm and yielded an efficient classifier for face detection.

Inspired by these researches, the purpose for extracting complex visual futures of the same content images is realized by defining a large set of features and with the help of the AdaBoost learning algorithm to select most critical features. Thus, an on-line CBPIR system for painting images is so constructed. In Section II, the system framework is introduced, and both original and modified AdaBoost algorithms are discussed. Section III describes the features selection. Section IV provides experimental results of the proposed system. In Section V, some parameters are discussed, and finally, some perspectives are given.

This research has been supported by the National Science Council, Taiwan, R.O.C., under Grant 94-2213-E-032-025.

S. H. Yen is with the PRIA Lab., Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei, Taiwan (Tel: 886-2-26215656, ext. 3305; fax: 886-2-26209749; e-mail: shyen@cs.tku.edu.tw).

M. H. Hsieh was with the PRIA Lab., Department of CS & IE, Tamkang University. He is now in military service (e-mail: the3002@pchome.com.tw).

C. J. Wang, and H. J. Lin are also with the PRIA Lab., Department of CS & IE, Tamkang University (e-mail: {121718, hjlin}@mail.tku.edu.tw).

1-4244-0100-3/06/$20.00 ©2006 IEEE

II. System Framework

Usually, a user may wish to search "portraits" (high-level concept) instead of the paintings with skin color (low-level feature). Our proposed system is designed to bridge the gap such that it classifies painting images according to their "contents" that people are really interested in. In order to extract and combine the features for different queries and be efficient enough for on-line learning, AdaBoost learning algorithm is adopted for training the classifier. In the following, AdaBoost Algorithm [9] and its modified version will be discussed. Since images from the initial query only provide a small amount of positive and negative training data, a device of relevance feedback (RF) is necessary to make more samples available for the system learning and thus improve the retrieval performance. We will also discuss the mechanism of the RF in the proposed system.

A. Original Adaboost Algorithm [9]

The AdaBoost Algorithm is used to boost the performance of a single weak classifier by combining a few weak classifiers to a strong classifier. T weak classifiers with the least error are selected to constitute a final strong classifier. The AdaBoost Algorithm is briefly described as blow.

Given: training images (x_i, y_i), $i=1,\ldots,n$, and $y_i = 0, 1$ for negative/positive training images respectively.

- Initialize the weights $\omega_{1,i} = \frac{1}{2m}$ or $\frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m, l are the number of negative and positive training images.
- For $t = 1, \ldots, T$:
  - For each feature j, a weak classifier h_j is trained. Next, calculate $\varepsilon_j = \sum_i \omega_{t,i} \left| h_j(x_i) - y_i \right|$.
  - Choose k such that $\varepsilon_k < \varepsilon_j, \forall j \neq k$. Let $h_k(\cdot) = h_t(\cdot)$ and $\varepsilon_t = \varepsilon_k$.
  - Update $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$ where $e_i = 0, 1$ for training image x_i being correctly or incorrectly classified by $h_t(\cdot)$, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$.
  - Normalize ω_{t+1} so that it is a distribution.
- The final strong classifier is

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \textit{otherwise} \end{cases} \qquad (1)$$

where $\alpha_t = \log \frac{1}{\beta_t}$.

When the strong classifier, as in (1), has been trained, it evaluates the score $\sum_{t=1}^{T} \alpha_t h_t(x)$ for every image x in the database. If $H(x) = 1$ then x is classified as a positive and a negative otherwise. The score indicates that how similar the image x is matched with the query.

To implement the AdaBoost Algorithm, each weak classifier h_j needs to be trained ahead of time. Because our system is for on-line learning, both accuracy and efficiency should be considered. A simple definition for weak classifier is as followed [10]:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where x is an image from database, $f_j(x)$ is the value of feature j for x, p_j is a parity sign, θ_j is the threshold determined as in (3) with C_n and C_p the negative and positive training set. γ= 0.5 is used for simplicity. Cai [10] suggests that $p_j = 1$ if the average feature values of positive training set is smaller than that of the negative training set and $p_j = -1$ otherwise.

$$\theta_j = \begin{cases} \gamma\left(\frac{1}{|C_n|}\sum_{x\in C_n} f_j(x) + \frac{1}{|C_p|}\sum_{x\in C_p} f_j(x)\right) & \text{if } p_j = 1 \\ (1-\gamma)\left(\frac{1}{|C_n|}\sum_{x\in C_n} f_j(x) + \frac{1}{|C_p|}\sum_{x\in C_p} f_j(x)\right) & \text{otherwise} \end{cases} \qquad (3)$$

B. Modified AdaBoost

To make the AdaBoost Algorithm more prominent for CBPIR, three modifications are made.

(1). Outliers Removing:

For each feature j, as mentioned, the range of $f_j(x_i)$ usually is large for different images x_i. Thus, in determining the threshold θ_j in (3), those extreme values of $f_j(x_i)$ should be removed to reduce the effect of outliers. The task is achieved by following steps.

- For each feature j, sort feature values $f_j(x_i)$, x_i is a training image, $i = 1, 2, \ldots, n$, with increasing order.
- Let $S_{\min} = S_m - 2\cdot(S_M - S_m)$ and $S_{\max} = S_M + 2\cdot(S_M - S_m)$ where S_m and S_M are the values of the top $n/4$ and $3n/4$ feature values respectively.
- The feature value which is not in the range $[S_{\min}, S_{\max}]$ will be removed.

---

(2). Preserving Previous Classifier:

Through the feedback procedure, as discussed in the following, the user would select some false positive images or false negative images from the retrieved images to refine the query. In addition to retraining a new classifier using added samples from RF [7], we preserve the previous training result and multiply it by a weight ρ, $0 \leq \rho \leq 1$. The modified final strong classifier is in (4), h' and α' are the previous weak classifiers and corresponding weights.

$$H(x) = \begin{cases} 1 & \rho\sum\alpha' h'(x) + \sum_{t=1}^{T}\alpha_t h_t(x) \geq \frac{1}{2}\left\{\rho\sum\alpha' + \sum_{t=1}^{T}\alpha_t\right\} \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

In this way, any insignificant features that had been selected previously will become less influential in succeeding feedback rounds due to multiplication by ρ. On the other hand, the truly critical features will be chosen no matter what the training set is, and thus their importance will be reinforced.

(3). Initial Weights of the New Training Images:

After a user selects false positive images and false negative images in the step of RF, together with the training images from previous classifiers, these sample images become the new training images for the next classifier. To improve performance of the next classifier, we will increase the chance that it can correctly classify those images which are falsely classified by the previous classifier. To accomplish this intention, we first evaluate the error ξ of the previous classifier on the new training images, and use it to modify the initial weights of the training samples of the AdaBoost algorithm when retraining the next classifier. The steps are described as below:

- Assign weights $\omega_{1,i} = \frac{1}{2m}$ or $\frac{1}{2l}$ for $y_i = 0, 1$ as before, where *m*, *l* are the number of negative and positive training images.
- Classify the training set using the previous strong classifier $H'(x)$. Compute error $\xi = \sum_i \omega_{1,i} \left| H'(x) - y_i \right|$.
- Update $\omega_{1,i} = \omega_{1,i} \beta^{1-e_i}$ where $e_i = 0,1$ for training image x_i being correctly or incorrectly classified by $H'(x)$, and $\beta = \frac{\xi}{1-\xi}$.
- Normalize: $\omega_{1,i} = \frac{\omega_{1,i}}{\sum_{j=1}^{n} \omega_{1,j}}$. The value of $\omega_{1,i}$ will be used as the initial weight for image x_i when train the next classifier.

C. Relevance Feedback (RF)

As stated in [11] that there is not a direct relation between the low-level feature and high-level concept, so a feedback mechanism is very helpful to improve the performance of the system. With additional information of the user's labeling on the relevance/irrelevance of retrieved images, the system learns the user's query concept, and gradually improves the retrieval result. Since images similar to the training images do not bring much new knowledge to the system, thus, to learn most of the user's target, images selected from a feedback mechanism should be as far as possible from the existed training images. In our system, user will be requested to label "irrelevant" images from the best matching retrieved images, and "relevant" from the worst matching un-retrieved images. Retrieved (or Un-retrieved) images are those images x with $H(x)$ to be 1 (or 0) in Eq. (4), and the best/worst matching is indicated by the magnitude of $\rho \sum \alpha' h'(x) + \sum_{t=1}^{T} \alpha_t h_t(x)$ from (4). The details will be described in Section IV.

III. FEATURE SELECTION

As observing paintings of different contents, they can be roughly divided into categories of portraits, landscape, still life, and others. In portraits, the character usually has a face shown on the center of canvas, and the background of portrait usually is simple. In landscape, there are usually some large objects (like sky, prairie, etc.) with layout to be in horizontal or diagonal and divided into two or three parts. The layout and the background of a still life could be very similar to a portrait, but there might be a desk or something to support the main character. In addition, different from natural images, paintings are not limited in colors. For example, to emphasize desperation of a person, his face may be painted in pale blue rather than skin color. Therefore, we design the features according to the aforementioned characteristics on gray scaled painting images. The features consist of global and local texture, and spatial arrangement on gray scaled painting images. Since the semantic content of a painting is far from characterization by one or two low-level features, and AdaBoost algorithm is very efficient in finding a combination of partial weak classifiers into a strong one. Accordingly AdaBoost with Spatial Angular Distribution (SAD), 644 features, Local Edge Pattern (LEP), 512 features, and Orientational Correlogram (OC), 3,200 features, is a very good match. Our experiments also confirm this point of view.

A. Spatial Angular Distribution (SAD)

Paintings have various layouts based on their contents and a painter's habit. Nevertheless, a portrait usually has a face around the upper center position; a vase with flowers usually has a vase right in the center of the painting, etc. Thus, not only the global layout, in certain location of the painting, e.g. the upper center, its local layouts are also important to understand the content of a painting. 23 blocks [12] are designed to extract global and local layouts as well as textural information. For each block, we will calculate the angle distribution of its edge pixels. For each angle, mean and standard deviation (SD), the average and SD of locations of edge pixels are also calculated. To cope with different painting sizes, position is transformed to relative position with respect to the height and width of the painting. Details are described in the following:

- Use Sobel operator to identify edge pixels. Angles of these edge pixels are calculated. For each block, considering those shaded area only, a histogram of 4 different angles $\theta_{i=1\sim4}$: 0°, 45°, 90°, 135°, is constructed. If an edge pixel with angle A is between two adjacent angles θ_i, θ_{i+1}, then amounts of $1 - \frac{|A-\theta_i|}{|\theta_{i+1}-\theta_i|}$ and $\frac{|A-\theta_i|}{|\theta_{i+1}-\theta_i|}$ will be accumulated to bins θ_i, θ_{i+1} respectively with i+1(mod 4) being considered. Finally, the accumulated value of each bin is divided by the total number of pixels in the block for normalizing.
- The average and SD of the locations, in x- and y-coordinate, of those edge pixels contributed to bin θ_i are evaluated. The average and SD values are calculated

by considering all edge pixels with angles between θ_{i-1} and θ_{i+1} since the amount of bin θ_i is accumulated from those edge pixels. The average and SD of angles to each bin θ_i are calculated in the same way. Together with the value of θ_i in the histogram, there are 7 feature values per angle θ_i. Hence there are 4x7=28 features for each block, and 28x23=644 features for one image.

B. Local Edge Pattern (LEP) [13]:

Extended from LBP [14], LEP represents the local texture of edge image effectively. First, get the texture of each pixel by taking the binary edge image obtained as in SAD to convolute with a mask shown in Fig. 1. Next, construct the LEP histogram h_{lep} for a region R such that the m^{th} bin $h_{lep}(m)$ is evaluated by (5),

$$h_{lep}(m) = \frac{n_m}{N}, \quad m = 0, \ldots, 511 \qquad (5)$$

where n_m is the number of pixels with LEP value m, and N is the number of total pixels in R. In here, the region R will be the whole image.

1	2	4
8	256	16
32	64	128

Figure 1 The mask used for calculating the LEP

C. Orientational Correlogram (*OC*):

One of well known disadvantages of histogram analysis is that it does not contain any spatial information. Qiu [15] proposed an orientational color correlogram (OCC). OCC extends from the traditional gray-level co-occurrence matrix [16] [17] which is widely used for texture description.

In here, only gray-level is considered so called OC (Orientational Correlogram) feature. Gray-levels in image I are quantized into $c_1, \ldots, c_L$ levels. Let I_c denote the set of pixels with its gray-level being c. Define the distance between two pixels p_1 and p_2 in the directional angle θ to be $\delta = |p_2 - p_1|_\theta$. OC is defined as in (6).

$$occ(\delta, \theta, i, j) = \Pr\left(p_1 \in I_{c_i}, p_2 \in I_{c_j}, |p_2 - p_1|_\theta = \delta\right) \quad (6)$$

In our research, gray-levels are quantized into 8 levels, and two orientations, horizontal and vertical, are considered. Distance is treated as the percentage of width (or height) of image when considering horizontal (or vertical) direction. Here we choose 25 distances including 1%, 3%, 5%, ..., 49% of width (or height). Hence, the dimension of OC is $\left(25 \times 2 \times 8^2\right) = 3200$.

IV. EXPERIMENT

In order to implement our system, 1047 paintings are downloaded from various web sites and 192 images are manually labeled "portrait" among them. Since paintings are in different sizes, images are rescaled proportionally so that they all have a height of 128. As the purpose of the system is to classify the semantic contents of paintings, the over all layouts are more important than detailed strokes of paintings. Thus a preprocessing of histogram equalization followed by a 7x7 median filter is applied to images to eliminate the influence of illumination and fine details. The detailed implementations of the system are described as below.

1). The system first randomly provides sample images, 12 images per page, for user to select his target as shown in Fig. 2. User selects about 5 images each that he does or does not want for positive and negative training images.

2). The classified result is presented on the Retrieval Pages. Every image x with $H(x) = 1$ in Eq. (4) will be displayed according to the descending order of the score $\rho \sum \alpha' h'(x) + \sum_{t=1}^{T} \alpha_t h_t(x)$. As in Fig. 3, the first page, i.e. 12 images with the highest scores, of the retrieval result is displayed after the initial query.

Figure 2 The page randomly providing sample images to initiate a query.

Figure 3 The Retrieval Page. Three images are labeled as

3). Relevance Feedback (RF) is next activated. After the Retrieval Pages are displayed, user should make a positive and a negative feedback to retrain the classifier. User is suggested to select about 3 to 5 irrelevant images starting from the first page of the Retrieval Pages, i.e. best matching irrelevant images. As indicated in Fig. 3, three images are selected as a negative RF. For user to make a positive RF, the system exhibits those un-retrieved images, i.e. image x with $H(x) = 0$ in (4). According to [11], the best way to add new positive images as feedback is to pick images farthest from the classify margin, i.e. images with the lowest scores. To avoid that there may be very few images user wants and causes a long searching for picking positive RF, the system shows images starting from the midst page instead of the last page of un-retrieved images. Then user makes his choice about 3 to 5 images as a positive RF. A midst un-retrieval page is shown in Fig. 4, three images are selected as a positive RF. To promote the performance, user is informed that any images in the first page of the Retrieval Page are treated as positive images unless user indicates it is not. So user must check the 1st page of the Result Page at least.

4). The system retrains the classifier according to the new training set obtained from step 3 and gets the refined retrieval result. As seen in Fig. 5, those 12 paintings on the first page of the Retrieval Page are correctly classified as "portrait" after one RF. User can repeat the RF process to get an even better retrieval result.

V. DISCUSSIONS

To evaluate performances of the system, we ask four students to label "portrait" from the collected database. 192 images are so labeled. In the following experiments, we restrict the query on the topic of "portrait" and use these labeled images as the ground truth. In addition, because sample images are randomly generated for user to initiate a query, thus the results will be different every time even if user makes queries for the same topic. Therefore, we repeat the same query 50 times and take the average performance in every experiment discussed below. The Recall Rate, Precision Rate, Top *N* Precision Rate are used to evaluate the performance as in Eq. (7) ~ (9). In experiments, *N* is set to be 100 which takes more than 8 retrieval pages to display and it is more than a user will be interested in general.

$$Recall\ Rate = \frac{\#\ of\ \ portraits\ retrieved}{\#\ of\ \ portraits\ in\ database} \tag{7}$$

$$Precision\ Rate = \frac{\#\ of\ \ portraits\ retrieved}{\#\ of\ \ retrieved\ images} \tag{8}$$

$$Top\ N\ Precision\ Rate = \frac{\#\ of\ \ portraits\ in\ Top\ N\ retrieved\ images}{N} \tag{9}$$

In this study, we did not compare our system with any previous works since there is no similar study on CBPIR. But various experiments are performed to study relations among the number of features selection, and number of RFs in the system.

1). T, the number of weak classifiers chosen in AdaBoost: To balance the performance of retrieval results and the cost of execution time, we perform a set of experiments with respect to different number of features in the system. In Fig. 7, every number is obtained from the average of 50 queries, and 3 RF for each query. In particularly, (d) is the average executing time for 3 feedback iterations, thus a total time of one complete query is multiplied by 4, including the initial query. Note that the Recall rate has not been much influenced by *T*. Consider both Precision and executed time, T = 32 is a reasonable choice.

2). Iteration of RF: In experiments, 3 RFs are used for typical users may have no patience to repeat RFs over 3 iterations. But if we can repeat RFs over and over again, the result is getting perfect as shown in Fig. 6.

Figure 4 An Un-Retrieval Page. Three images are labeled as "relevant".

Figure 5 The first page of Retrieval Pages after the first relevance feedback.

VI. CONCLUSION

In this paper, a CBPIR based on AdaBoost Algorithm is proposed. The AdaBoost learning algorithm is modified to encourage the consequence of the feedback mechanism. How to choose images for positive/negative RF is also discussed to refine succeeding results. Due to the purpose of recognizing the contents of paintings, we adopt features of SAD, LEP and OC, a total of 4,356, and the characteristic of AdaBoost that efficiently combining weak classifiers into a strong one. Our experiment, based on 3 RFs and an average of 50 repetitions, shows an excellent performance of (approximately) 0.71, 0.84, 0.95 in Precision, Recall, and Top 100 Precision rates.

In the subsequent work, we will enlarge the database and collect more ground truth images to further verify the efficiency of the system. Moreover, as indicated from Fig. 7, how to utilize the learning experiences from previous users if the same topic is queried again is also a very interesting question. To combine a keyword query with our system is also under investigated.

Reference

[1] M. Flickner *et al.*, "Query by image and video content: the QBIC system," *IEEE Computers*, Vol. 28 (9), pp. 23-32, 1995.

[2] Gupta, A *et al.*, "The Virage image search engine: an open framework for image management" in *Storage and Retrieval for Image and Video Databases IV,* Proc SPIE 2670, pp. 76-87, 1996.

[3] J. Smith, S. Chang, "VisulSEEK: a fully automated content-based image query system," *Proc. ACM International Conference on Multimedia*, pp. 87-98, 1996.

[4] Jing-Nan Liu and Man-Kwan Shan , "A Personalized Fine Arts Painting Style Query System," 1st Digital Archives Technique Conference, Nangang, Taipei 2002.

[5] A. Kushki, P. Andriutsos, K. N. Plataniotis, A. N. Venetsanopoulos, "Retrieval of Images From Artistic Repositories Using a Decision Fusion Framework," *IEEE Transactions Image Processing*, Vol. 13, No. 3, March 2004.

[6] Jeremy S. De Bonet, Paul Viola, "Structure Driven Image Database Retrieval," Proceedings of the 1997 conference on *Advances in neural information processing systems 10.*

[7] Kinh Tieu and Paul Viola, "Boosting Image Retrieval," Proceedings of the IEEE Conference on *Computer Vision and Pattern Recognition*, Vol. 1, 2000.

[8] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Proceedings of the 2001 IEEE Computer Society Conference on *Computer Vision and Pattern Recognition* (CVPR 2001), Vol. 1, pp. 8-14, 2001.

[9] Y. Freund and R. E. Schapire. "A decision-theoretic generalization of online learning and an application to boosting." *J. Comp. & Sys. Sci.*, 55(1): pp. 119-139, 1997.

[10] Yizheng Cai, "Image Retrieval Using Boosting Algorithm," http://www.cs.ubc.ca/~yizhengc/ .

[11] A. Kushki, P. Androutsos, K. N. Plataniotis, A. N. Venetsanopoulos, "Query Feedback for Interactive Image Retrieval," IEEE Transactions on *Circuits and Systems for Video Technology*, Vol. 14 (5), May 2004.

[12] http://pria.cs.tku.edu.tw/cht/downloads/research/layouts.bmp

[13] Y.-C. Cheng, S.-Y. Chen, "Image classification using color, texture and regions," *Image and Vision Computing*, Vol. 21 (9), pp. 759-776, Sep. 2003.

[14] T. Ojala, M. Pietikainen, "Unsupervised texture segmentation using features distributions," *Pattern Recognition* 32, pp. 477-486, 1999.

[15] Xipeng Qiu, Zhe Feng, Lide Wu, "Boosting Image Classification Scheme," *IEEE ICME*, 2004.

[16] R.M. Haralick, K. Shanmugan, and I. Dinsten, "Texture Feature For Image Classification", IEEE Trans. *Syst. Man. Cybern.*, SMC-3(6): pp. 610-621, 1973.

[17] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms", *International Journal of Computer Vision*, pp. 245-268, 1999.

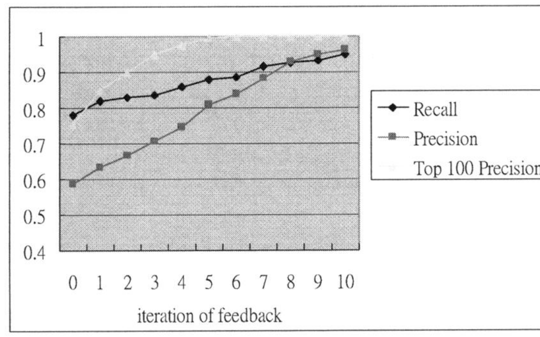

Figure 6 the more feedback processed, the better performance

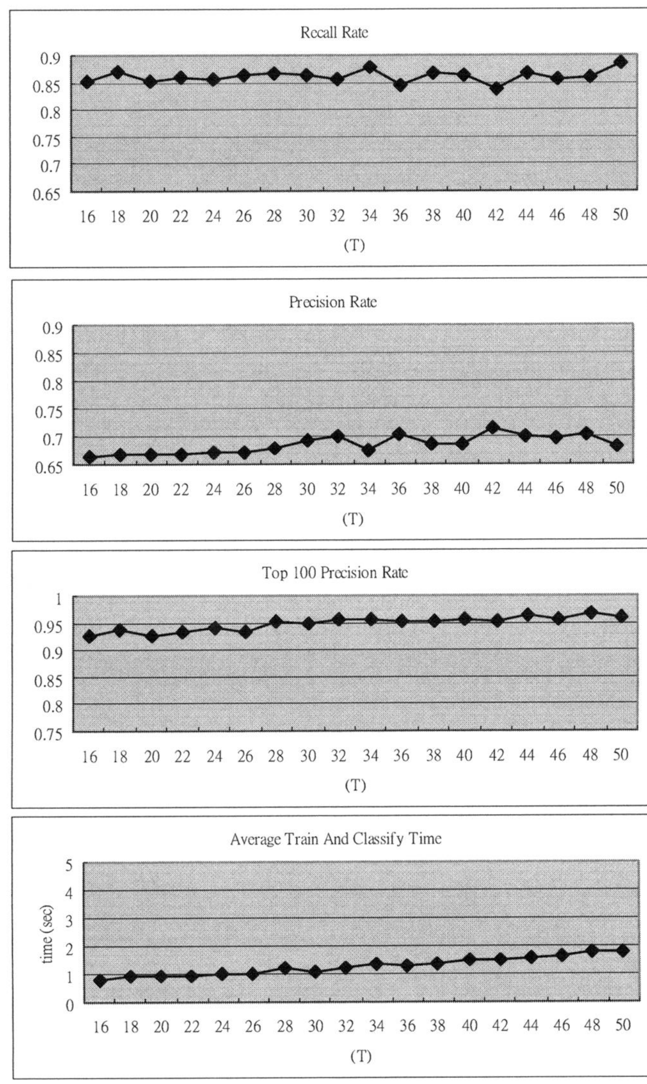

Figure 7 From top to bottom: Recall rate; Precision rate; Top 100 precision; Average Time for training and classifying. Experiments are performed with respect to *T* features chosen.