

A Novel Combinational Relevance Feedback Based Method for Content-based Image Retrieval

Mohammadreza Azodinia and András Hajdu

Department of Computer Graphics and Image Processing, University of Debrecen,
Egyetem tér 1, 4032 Debrecen, Hungary;

Email: m.r.azodinia@inf.unideb.hu, hajdu.andras@inf.unideb.hu

Abstract: Due to the extensive use of images in various fields, using effective approaches to retrieve the most related images given, a query image is of great importance. Content-based image retrieval is the approach commonly used to address this issue. The content-based image retrieval systems use many techniques to provide more accurate and comprehensive answers, among which, is relevance feedback. Relevance feedback is used by the system to help it retrieve more relevant images in response to a query. In this paper, we have proposed a novel relevance feedback method that is able to improve the precision of the content-based retrieval systems. The proposed method is based on multi-query relevance feedback, and similarity function refinement.

Keywords: CBIR image retrieval; CBVIR information retrieval; image database; recommender system

1 Introduction

Nowadays, digital images are commonly utilized in many areas from biology and medicine, to face or finger print recognition. Therefore, effective methods for searching and retrieval of images have received considerable interest. In traditional image retrieval systems keywords and captions representing the images were used to retrieve the images [1, 2]. These captions were generated manually, which was quite inefficient and expensive. Furthermore, the manual generation of the keywords is unable to capture every noticeable keyword that describes an image. In order to overcome the major deficiencies of the conventional approaches, content-based image retrieval (CBIR), also known as query by image content (QBIC), and content-based visual information retrieval (CBVIR), was introduced. The term "content-based" alludes that the contents of the image are used in the retrieval process, rather than the metadata such as keywords, or captions associated with the image which are traditionally used.

Most CBIR systems rely on low-level image features, i.e., color, texture and shape which are extracted from each image. These features are arranged so that they form an appropriate feature vector (FV). When the system receives a query image, the feature vector is extracted from it, and then this image can be compared to the images stored in the system. The images with the minimum distance to the query image are displayed as the result. In order to increase the overall efficiency of the system, the low-level features are usually stored in a database.

The main advantage of the CBIR systems is that the retrieval process is automatic, and it is not necessary to manually provide any metadata for images. However, one of the problems the CBIR systems have to face is that images are complex to manage. Also, image indexing is not a trivial task. Furthermore, CBIR systems require a mapping from the high-level user perceptions into low-level image features, which is known as the “semantic gap” problem. A survey on this concept is presented in [3].

In order to reduce the semantic gap, many different approaches such as image semantic classification, short-term learning (STL), and long-term learning (LTL) have been proposed, which are based on the relevance feedback technique. The relevance feedback technique was used in the CBIR for the first time by Rui in 1998 [4].

In the relevance feedback technique the user interacts with the system until the desired results are achieved. This interaction helps the system to establish a meaningful relation between the low-level features and the high-level perceptions, which reduces the semantic gap. Indeed, learning by means of relevance feedback is done in two ways, namely: short-term learning, and long-term learning. Short-term learning is done according to a query from the user in order to lead the system towards the user’s desired results. In every interaction with the user, the system tries to find out how the user thinks. Short-term learning can be categorized into four different approaches:

- Classification-based methods;
- Query refinement methods;
- Similarity refinement methods;
- Multi-query methods.

2 Short-Term Learning Methods

Considering short-term learning and long-term learning we have put our focus on the former in this paper; therefore, in this section, the four sub-categories of short-term learning are briefly explained along with some researches conducted based on each method.

2.1 Classification Methods

The methods belonging to this category train a classifier using the training set, which is provided by the user and contains some relevant and irrelevant images regarding the query images. This classifier is then used to distinguish between the relevant and irrelevant images regarding the unseen query images. One of the most widely used classifiers to classify the images as described, is the Support Vector Machine (SVM). For instance, in [5] fuzzy SVM has been used in their proposed method in order to separate the relevant and irrelevant images. Indeed, in this method that combines the short-term and long-term learnings, the short term learning technique splits an image into several regions, then the fuzzy support vector machine learning is applied to these regions. Subsequently, the long-term learning approach tries to adaptively learn the semantic concepts represented in each image by means of relevance feedback and semantic clustering.

Djordjevic et al. [6] proposed a method whereby the images are partitioned into small blocks and the low-level features are extracted from each block. Then, the blocks are clustered, and some representatives are achieved for each image. After receiving the relevance feedbacks from the users, the relevant and irrelevant images are used by a SVM classifier.

2.2 Similarity Refinement Methods

The similarity refinement process adjusts the similarity function during the feedback iterations in order to refine the query response. Each CBIR system uses a similarity function to retrieve the most similar images to the query image. This similarity function is typically based on various low-level features, each of which may have different weights. The weight of these features are adjusted in the similarity refinement process.

The similarity of two images Q and T can be computed using the following formula:

$$D(Q, T | W) = \sum_{i=1}^n w_i d(q_i, t_i) \quad (1)$$

where $W = [w_1, w_2, \dots, w_n]$ is the weight vector which determines the weight of each low-level feature used in the similarity measure; q_i and t_i are the values of the i^{th} feature from the feature vectors representing the images Q and T respectively, n is the length of the feature vectors, and $d(\cdot)$ is a distance function.

An approach to refine the similarity function is to use the variance of features of the relevant and irrelevant images. In [4] after each feedback iteration, where the relevant and irrelevant images are specified by the user, the weight of each feature

of the feature vector is adjusted by the inverse of the standard deviation of that feature over the relevant features.

Cheng *et al.* [7] proposed a method wherein the users provide a ranking for the images in the feedback rounds. Subsequently, the system computes the ranking of exactly the same images based on the current weight vector for the features. Based on the similarity between these two kinds of rankings, the weights of the similarity function are adjusted.

In [8] an optimization function is defined and the weights of the features have been adjusted using the Laplace method. In this method some pre-stored images are used for the optimization task.

Another approach to similarity function refinement is to define a cost function and use the Gradient-descent method to adjust the weights of the features [9]. The main aim of the methods that take this approach is to find a better cost-function. The idea behind this approach is borrowed from the supervised learning [10].

2.3 Multi-Query Methods

In these methods the retrieval is performed using several queries. There are several ways to create these query vectors based on a given query. One of the most widely used approaches is clustering the images that are recognized to be relevant according to the user's feedbacks and to use the centers of these clusters as new queries. Salvador *et al.* proposed a method to estimate the proper number of clusters [11].

Kim *et al.* [12] have used the Q-clustering method in order to create several queries from the primary query. In their method, first the query image is given to the system, then the system retrieves M images from the database which have the least distance to the query image. These images are categorized by the user as the relevant or irrelevant images. At the first iteration, the relevant images are classified and during the rest of iterations the images are categorized into these pre-defined categories. The Bayesian classifier is used for the classification task. Also, the authors proposed another method which uses hierarchical clustering instead of the Q-clustering [13].

Herraez *et al.* [14] used genetic algorithms to create new query vectors by applying the genetic operators, i.e. mutation and crossover, on the relevant images. The new query vectors are created by mapping each child into one of the images from the image database.

2.4 Query Refinement Methods

This method was first proposed in 1997 in a system called MARS [15], where the average of all the images that were considered to be relevant to the query was computed and the result was used as the new query.

In [15] both the relevant and irrelevant images have been used to improve the query vector. Indeed, this method, which is known as Ricchio's method, tries to move the query vector closer to the relevant images and move it away from the irrelevant ones during each step.

3 Architecture of CBIR Systems

As mentioned before, a CBIR system receives a query image and returns the most similar images to the query image among the images stored in the images database. The queries are processed based on the low-level features of the images. The query-processing module is responsible for extracting the features of the query image and comparing the feature vector to the stored images. Once the similarities between the query image and the images stored in the database are computed, the stored images are ranked based on their distance to the query image, and the most similar ones are retrieved.

In order to increase the efficiency of the system, the features vectors representing each image are extracted and stored in the database. This process, which is done by data insertion subsystem, is usually performed off-line. It is worth mentioning that like other forms of databases, the image databases are typically indexed too. The image indexing is performed based on the feature vectors extracted from each image, and utilizes structures such as M-tree [16] or Slim-tree [17] to accelerate the similarity computation and the retrieval processes. Last but not least, it should be noted that various CBIR systems may vary this architecture or the actions carried out in each phase according to their needs. For instance, the method proposed by Santhosh et al. [18] includes the conversion of images to gray scale and image segmentation through a clustering method.

4 Proposed Method

Firstly, we should mention that in the proposed system, some general low-level features are extracted from the images contained in the database of images. The database is represented by $X = \{X_1, X_2, \dots, X_N\}$ where, X_i refers to the i^{th} image. For every image X_i there is a feature vector F_i . Therefore, as shown in Figure 1, the image database contains a database of feature vectors

$F = \{F_1, F_2, \dots, F_N\}$ corresponding to X . We extracted four types of low-level visual features, three of which are color features and one of them is a texture feature.

4.1 Features

Color Moments

The mean, variance and skewness of an image which are known as the color moments (respectively, known as the first, second, and third color moments) characterize the color distribution in an image.

The first moment can be calculated by using the following formula:

$$\mu_i = \sum_{j=1}^N \frac{1}{N} P_{ij} \quad (2)$$

where N is the number of pixels in the image and p_{ij} denotes the value of the j^{th} pixel of the image at the i^{th} color channel.

The second color moment can be calculated by using the following formula:

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - \mu_i)^2\right)} \quad (3)$$

The third color moment can be calculated by using the following formula:

$$\gamma_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - \mu_i)^3\right)} \quad (4)$$

Color histogram

A color histogram denotes a representation of the statistical distribution of the colors in an image, and disregards the spatial location of the colors. Each bin of the histogram corresponds to a color in the quantized color space. The number of the bins is specified by the number of colors in the image.

The color histogram can be measured for any kind of color space. The number of elements in a histogram is specified by the number of bits in each pixel of the image. Indeed, an n -bit image has 2^n values (or intensities) lying in the interval $[0, 2^n - 1]$.

Although the color histogram is relatively invariant under the translation and rotation about the viewing axis and varies just slightly with the angle of the view,

it is significantly affected by the changes in the illumination, shading, highlights, and the inter reflections. To overcome these drawbacks, Gevers et al. [19] proposed a robust histogram for object recognition that is invariant to the aforementioned changes. Domke and Aloimonos [20] proposed a method to create a color histogram which makes the histogram invariant under any mapping of the surface that is locally affine, which includes a very wide range of the viewpoint changes or deformations. The main idea is that the pixels in the two images can be weighted using the gradients in different color channels. Therefore, a deformation of a region of the image will change both the image and the weights equally. We have used the method proposed in [20] in this study.

Edge histogram

Edges are among the important features of images. In the MPEG-7 standard the edge histogram is used in order to capture the edge distribution of the image. However, its drawback is that it contains only the local edge distribution with 80 bins. In this paper, we have used the method proposed in [21], which generates two other types of the edge histograms: the global, and semi-global histogram bins.

Gabor features

The texture features of each image are captured using the Gabor filter. The Gabor filter provides optimal joint resolution in both frequency and spatial domains [22]. We have used 30 filters with five different scales and six different orientations [23]. We first create a grayscale equivalent for each image and normalize it so as to have dimensions of 256×256 to speed up the computation using the FFT method. After filtering the images, their means and standard deviations are computed which provide 60 features for each image.

5.2 Weight Adjustment

In our proposed method, the query vector is broken into several multi-query vectors in regard to every feature space to perform the short-term learning. Initially, the user feeds the system with a query image Q , then, the features of the query image are extracted. These features are illustrated in the following section. The extracted feature vector is then used by the similarity function to retrieve P relevant images from the stored images. The images are then partitioned into two categories by the user: relevant images, and irrelevant images. At the next stage, the multi-query approach is used. This is done by clustering the images that are categorized as relevant, and using each cluster center as a new query. Subsequently, the minimum distance of each image from the cluster centers is considered as its similarity to the query. We have used the WPGMC¹ algorithm to perform the clustering.

¹ Weighted Pair Group Method Centroid

In order to combine the results of the retrieval based on each feature i , the ranking of each image in the database is computed when images are retrieved based solely on that feature. This ranking is then used to adjust the weight of each feature. Suppose we have N images in the database, and we offer the user only the M first retrieved images. First, the rankings of each of these M images are added together. Then according to the difference between the summation achieved at stage t and stage $t+1$ the new weights are computed. It should be noted that initially the weight of all the features are equal. In order to shed more light on this process, we bring an example. Suppose the database contains 8 images, i.e., $\{A, B, C, D, E, F, G, H\}$, and the system shows the user the first 4 images, i.e., $\{I_1, I_2, I_3, I_4\}$. Furthermore, suppose we use three features f_1, f_2, f_3 . Also, the first four features retrieved at stage t are A,B,C, and D, and the first four features retrieved at stage $t+1$ are E,F,G, and H. The orders of the images retrieved using each feature separately at stages t and $t+1$ are represented in Table1 and Table2 respectively.

Table 1

The orders of the images retrieved using each feature separately at stage t

Retrieval based on each feature		
f_1	f_2	f_3
C	A	F
E	F	D
F	B	E
H	G	G
B	C	C
D	D	B
A	E	H
G	H	A

Table 2

The orders of the images retrieved using each feature separately at stage $t+1$

Retrieval based on each feature		
f_1	f_2	f_3
A	C	G
B	D	H
H	A	E
C	F	B
G	B	A
F	G	F
E	E	C
D	H	D

Considering Table 1, Table 2, and the aforementioned assumptions, we can deduce the results shown in Table 3.

It is worth mentioning that we call the sum of feature-wise ranking of the selected images the *feature-wise sum of the rankings*. The *normalized feature-wise sum of the rankings* for each feature is computed by dividing the feature-wise sum of the rankings for that feature by the summation of the feature-wise sum of the rankings.

Table 3
The rankings of the first 4 images according to each feature

Stage	Image	The ranking based on each feature			
		f_1	f_2	f_3	
t	I_1	A	7	1	8
	I_2	B	5	3	6
	I_3	C	1	5	5
	I_4	D	6	6	2
	The feature-wise sum of the rankings		19	15	21
	The normalized feature-wise sum of the rankings		19/55	15/55	21/55
$t + 1$	I_1	E	7	7	3
	I_2	F	6	4	6
	I_3	G	5	6	1
	I_4	H	3	8	2
	The feature-wise sum of the rankings		21	25	12
	The normalized feature-wise sum of the rankings		21/58	25/58	12/58

As implied from Table 3, the normalized total ranking of the first 4 images considering only feature f_1 has decayed from 0.34 to 0.36 so its corresponding weight should decrease accordingly. On the other hand, the normalized total ranking of the first 4 images considering only feature f_3 has improved from 0.38 to 0.2, therefore, its corresponding weight should increase accordingly. It is clear from this example that the values of the feature-based normalized rankings always sum up to 1. This is exactly the case for the weights of each feature in the similarity function. Therefore, considering the sign of the changes in these values, the total changes will always be equal to 0. Consequently, the difference between the normalized feature-wise sum of rankings of that feature at stages t and $t+1$ is added the weight of each feature. That is, the weight of the feature f_i at stage $t+1$ is computed using the following formula:

$$w_i^{t+1} = w_i^t + \left(\sum_{j=1}^M r_{ij}^t - \sum_{j=1}^M r_{ij}^{t+1} \right) \quad (5)$$

where r_{ij}^t is the ranking of the j^{th} relevant image when it is retrieved using only the i^{th} feature, and M is the number of the images that are given to the user. It should be reminded that the relevance of the images is determined by the user.

5.3 Similarity Distance

Thanks to its simplicity, the Euclidean distance is the most commonly used distance metric among all the image distance metrics.

Given two M by N images, $I = (i^1, i^2, \dots, i^{MN})$ and $J = (j^1, j^2, \dots, j^{MN})$, the Euclidean distance is computed by the following equation:

$$d_E^2(I, J) = \sum_{k=1}^{MN} (i^k - j^k)^2 \quad (6)$$

6 Experiments and Results

In order to evaluate the proposed method, Matlab running on a PC with an Intel CORE 2 Due 2.20 GHz processor and 4 GBs of RAM was implemented. A collection composed of 20000 JPEG formatted images borrowed from the Hamshahri2 dataset was used in our experiments [24]. Furthermore, 50 query images chosen from 50 different semantic groups were examined in the experiments. Each of these 50 images was used in a scenario wherein the image was given to the user as the query image, then the system offered the user a predefined number of images as the relevant images. Thereafter, the user was asked to provide a feedback and to decide which images were relevant and which ones were not. The relevant images were used for updating the similarity function according to the proposed method. The updated similarity function was used to retrieve images again. This loop was repeated 4 times for each image. We have compared our proposed method with the following methods:

- CC: The Classifier Combination, which is based on the multi-query approach.
- RR: This method uses the Rocchio's method [16] for the query vector refinement and uses the Rui's method [4] for the distance function refinement;
- RGG: This method uses the Rocchio's method [16] for the query vector refinement and uses the method proposed by Guldogan & Gabouj [25] for the distance function refinement.

The precision of the proposed method is compared to the precisions of the other methods and the results are depicted in Figure 2. It is worth mentioning that the results achieved at each iteration are averaged over all the categories. The equation 7 is used to compute the precision of each method.

$$\text{Precision} = \frac{\text{The number of the relevant images in the first } j \text{ positions of the retrieved images}}{j} \quad (7)$$

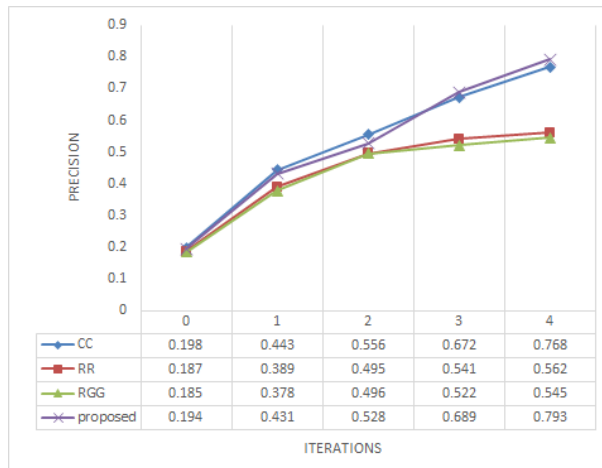


Figure 1

The average precision graph depicted for the top 20 retrieved images

The average time for the retrieval of images by means of each method is shown in Figure 3.

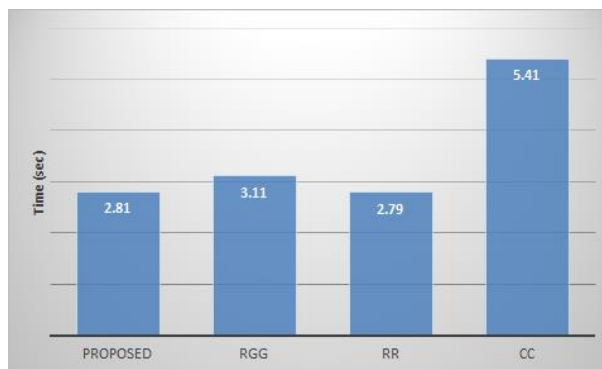


Figure 2

The average time for the retrieval of images by different methods

It can be alluded from the above results that the proposed method not only produces more accurate results, but also achieves the results in a reasonable time.

Furthermore, the precisions of each method for some randomly selected semantic groups are shown in Table 4. The best result is shown by bold face for each case.

Table 4
Comparison of the precision of the proposed method with other methods for the top 20 retrieved images

Semantic group	CC	RR	RGG	proposed
Sports	0.8185	0.5643	0.5912	0.9261
Buildings	0.7203	0.6485	0.7943	0.8527
Animals	0.9918	0.9561	0.8924	0.9840
Weather Forecast	0.5470	0.7511	0.7519	0.7455
Accidents	0.4149	0.2206	0.2418	0.5613
Cars	0.8651	0.8769	0.8133	0.8774

According to Table 4 the proposed method outperforms the other well-known methods in most cases.

Conclusion and Future Work

In this paper we developed a relevance feedback based approach to content-based image retrieval system. The proposed method combines the color, spatial, and texture features and adjusts the similarity function in order to retrieve the most relevant images from an image database in response to a query image. Our approach takes advantage of the multi-query methods as well, which results in better precision while the method is still fast. For the future studies we will try to combine our approach with some supervised learning methods. Also, we will try to use and evaluate other distance measures to compute the distance between the feature vectors.

Acknowledgement

This work was supported in part by the project VKSZ_14-1-2015-0072, SCOPIA: Development of diagnostic tools based on endoscope technology supported by the European Union, co-financed by the European Social Fund.

References

- [1] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from Relational Database of Images. *IEEE Computer*, 28 (9): 40-48, Sept. 1995
- [2] H. Lieberman, E. Rosenzweig, and P. Singh. Aria: An Agent for Annotating and Retrieving Images. *IEEE Computer*, 34 (7): 57-62, 2001
- [3] Ying Liu, Dengsheng Zhang, Guojun Lu. Regionbased Image Retrieval with High-Level Semantics using Decision Tree Learning. *Pattern Recognition* 41, pp. 2554-2570, 2008

-
- [4] Rui, Y., Huang, S., Ortega, M., Mehrotra, S., Relevance Feedback: a Power Tool for Interactive Content-based Image Retrieval, *IEEE Transactions on Circuits and Video Technology*, Vol. 8, No. 5, pp. 25-36, 1998
 - [5] Barrett, S., Content-based Image Retrieval: a Short Term and Long-Term Learning Approach, 2007
 - [6] Djordjevic, D., Izquierdo, E., An Object- and User-driven System for Semantic-based Image Annotation and Retrieval, *IEEE transactions on circuits and systems for technology*, Vol. 17, No. 3, pp. 313-323, 2007
 - [7] Cheng, P. C., Chien, B. C., Ke, H. R., Yang, W. P., A Two-Level Relevance Feedback Mechanism for Image Retrieval, *Expert Systems with Applications*, Vol. 34, pp. 2193-2200, 2008
 - [8] Hoi, S. C. H., Liu, W., Chang, S. F., SemiSupervised Distance Metric Learning for Collaborative Image Retrieval, *ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP)*, Vol. 6, No. 3, pp. 1-26, 2010
 - [9] Deselaers, T., Paredes, R., Vidal, E., Ney, H., Learning Weighted Distances for Relevance Feedback in Image Retrieval, 19th International Conference on Pattern Recognition (ICPR), 2008
 - [10] Paredes, R., Vidal, E., Learning Weighted Metrics to Minimize Nearest Neighbor Classification Error, *PAMI*, Vol. 28 (7), pp. 1100-1110, 2006
 - [11] Salvador, S., Chan, P., Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms, Technical Report CS-2003-18, Florida Institute of Technology, 2003
 - [12] Kim, D. H., Chung, C. W., Qcluster: Relevance Feedback Using Adaptive Clustering for Content-Based Image Retrieval”, *Proc. ACM SIGMOD*, pp. 599-610, 2003
 - [13] Kim, D. H., Chung, C. W., Barnard, K., Relevance Feedback using Adaptive Clustering for Image Similarity Retrieval, *The Journal of Systems and Software*, Vol. 78, pp. 9-23, 2005
 - [14] Herraiez, M. A., Ferri, F. J., Picot, S. M., Distancebased Relevance Feedback using a Hybrid Interactive Genetic Algorithm for Image Retrieval”, *Applied Soft Computing*, Vol. 11, pp. 1782-1791, 2011
 - [15] J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313-323. PrenticeHall, Englewood Cliffs, NJ, USA, 1971
 - [16] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pp. 426-435, Athens, Greece, 1997

- [17] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14 (2): 244-260, March/April 2002
- [18] Mathew S., Balas V., Zachariah P., and Samue P., "A Content-based Image Retrieval System Based on Polar Raster Edge Sampling Signature," *Acta Polytechnica Hungarica*, Vol. 11, No. 3, pp. 25-36, 2014
- [19] Theo Gevers and Harro Stokman. Robust Histogram Construction from Color Invariants for Object Recognition. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 26, No. 1, pp. 113-118, 2004
- [20] J. Domke, Y. Aloimonos, "Deformation and Viewpoint Invariant Color Histograms", *Proc. BMVC (British Machine Vision Conference)*, September 2006, Edinburgh, UK
- [21] Park DK, Jeon YS, .Won CS, Efficient Use of Local Edge Histogram Descriptor. *Proceedings of the 2000 ACM workshops on Multimedia*: 51-54, 2000
- [22] B. S. Manjunath, et al., *Introduction to MPEG-7*, Wiley, New York, 2002
- [23] Manjunath BS, Ma WY, Texture Feature for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18 (8): 837-842, 1996
- [24] Abolfazl AleAhmad, Hadi Amiri, Ehsan Darrudi, Masoud Rahgozar, & Farhad Oroumchian. (2009) *Hamshahri: A Standard Persian Text Collection*. *Knowledge-based Systems*, 22 (5), 382-387
- [25] Guldogan, E. & Gabbouj, M., Dynamic Feature Weights with Relevance Feedback in Content-based Image Retrieval. In *24th International Symposium on Computer and Information Science*. IEEE, pp. 56-59, 2009