Fuzzy logic based interpretation and fusion of color queries

B. Verma[a,*], S. Kulkarni[b]

[a] *School of Information Technology, Griffith University, Gold Coast Campus, QLD 9726, Australia*
[b] *Department of Computer Science and Mathematics, Nipissing University, North Bay, Canada ON P1B 8L7*

Received 23 August 2002; received in revised form 24 April 2003; accepted 15 May 2003

1. Introduction

Interest in the potential of digital images has increased enormously over the last few years, fuelled at least in part by the rapid growth of imaging on the World Wide Web (WWW). Creating and storing digital images nowadays is easy and getting cheaper all the time as the needed technologies are becoming available to the masses. There already exist a vast number of digital visual data sources, e.g. different kinds of sensors, digital cameras and scanners in addition to the various image collections and databases for all kinds of purposes. Furthermore, the fast development of computing hardware has enabled us to switch from text based computing to graphical user interfaces (GUIs) and multimedia applications.

At present, the most common technique for integrating images into a database is to store images together with some descriptive text or keywords assigned by human operators. Image retrievals

* Corresponding author. School of Information and Communication Technology, Griffith University, PMB 50, Gold Coast Mail Centre, QLD 9726, Australia. Tel.: +61-7-5552-8592; fax: +61-7-5552-8066.

E-mail addresses: b.verma@mailbox.gu.edu.au, b.verma@griffith.gu.edu.au (B. Verma), siddhik@nipissingu.ca (S. Kulkarni).

are performed by matching the query texts with the stored descriptive keywords. However, as this approach is exclusively text based and no visual properties of underlying data are employed, there are several problems inherent in the systems. First of all, as text descriptions of image contents are assigned and keyed in by human operators, not only is the process time consuming due to enormous volumes of image data, but it is also very subjective and incomplete. Retrieval may fail if the user forms the query based on a different set of keywords, or the query refers to the image contents that were not initially described. An ideal system should allow both a keyword and a concept search, in conjunction with a content-based search. However, most of them do not provide flexibility in formulating these queries in natural language. At the start, the user should be able to either issue a direct query or query by example. The system receives images that are similar to the users' query and ranks them on the basis of similarity. Many strategies and algorithms have been proposed for similarity-based retrieval from the high dimensional index structure. However, there has been little work on query processing based on natural language [14] and fusion of multiple queries.

The rest of the paper is organized as follows: Section 2 describes the background, Section 3 details the research methodology, experimental results are described in Section 4, these results are analyzed and compared in Section 5 and Section 6 concludes the paper.

2. Background

Traditional text-based image retrieval systems use keyword annotations as the retrieval paradigm [17]. This approach does, nevertheless, have some obvious shortcomings and difficulties. Annotating large databases takes a lot of effort as the annotations must be entered manually. Another problem is the possibility of different interpretations of the image content. Different people see different things in images and annotations cannot possibly cover them all. As a result, a fraction of potentially relevant images may not be included in the result of a query.

Problems with traditional methods of image indexing have led to the rise of interest in techniques for retrieving images on the basis of automatically derived features. To overcome this problem, several such systems have been developed [2]. After a decade of intensive research, CBIR technology is now beginning to move out of the laboratory and into the market place, in the form of commercial products like QBIC [6,7] and Virage [9,10]. However, the technology lacks maturity, and is not yet being used on a significant scale. In the absence of hard evidence on the effectiveness of CBIR techniques in practice, opinion is still sharply divided about their usefulness in handling real life queries in large and diverse image collections [8].

3. Research methodology

This section describes the proposed technique for retrieving images based on the color feature of the image. This section is further divided into two stages to describe the technique in detail. The first stage concentrates on preprocessing, which includes building an image database, feature extraction technique and preparing the feature database. The second stage describes database retrieval, which contains fuzzy interpretation of the queries, a neural network for learning those queries and fusion of multiple queries.

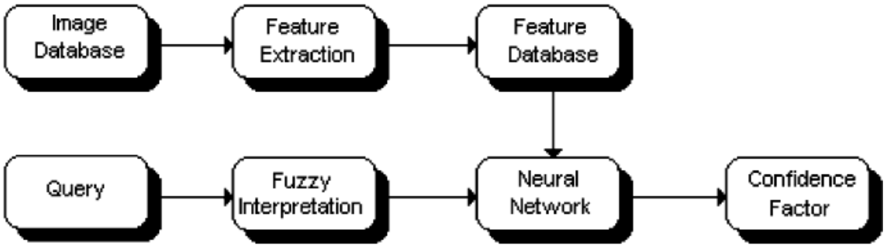

Fig. 1. Block diagram for color image retrieval.

Color is a very important cue in extracting information from images [13]. This feature is used in most CBIR systems. Before retrieving the images from the database based on the color feature, it is necessary to preprocess the image, extracting the color features and storing them in a database. Fig. 1 shows the different components used for retrieving images from a database. These include forming the image database, color feature extraction from each image, preparing the feature database, fuzzy interpretation of queries, use of a neural network to learn those queries and to display the images along with the confidence factor on the WWW.

3.1. Preprocessing

Different real world images were downloaded from the WWW and stored in an image database. This image database contains a variety of images of scenery, animals, people, flowers, etc. The majority of these images are in jpg format. For preprocessing, the images are converted from jpeg format to raw format.

Color feature extraction forms the basis of color image retrieval. The distribution of color is a useful feature for image representation. Color distribution, which is best represented as a histogram of intensity values, is more appropriate as a global property which does not require knowledge of how an image is composed of different objects. So this technique works extremely well to extract global color components from the images. A color histogram technique is proposed for extracting the colors from the images. The color histogram for an image is constructed by counting the number of pixels of each color. The color of any pixel may be represented in terms of the components of red, green and blue values. These histograms are invariant under translation and rotation about the view axis and change only under the change of angle of view, change in scale and occlusion. Therefore, the color histogram is a suitable quantitative representation of image content [1].

Let F_S denote the set of features used to represent color content, $F_S = \{\text{color}\}$. The feature representation set is motivated by the research results of Nepal, Ramakrishna and Thom [14] who identified nine colors that fall within the range of human perception. They used exactly nine colors out of 13 colors used by Carson and Ogle [3]. Also, nine colors are used in Berkeley's Digital Library project and we used the mapping program provided by Berkeley http : //elib.cs.berkeley.edu/src/cypress/meets.c. The feature representation set of colors is rep $(\text{color}) = \{$red, green, blue, white, black, yellow, orange, pink, purple$\}$.

An image histogram refers to the probability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three-color

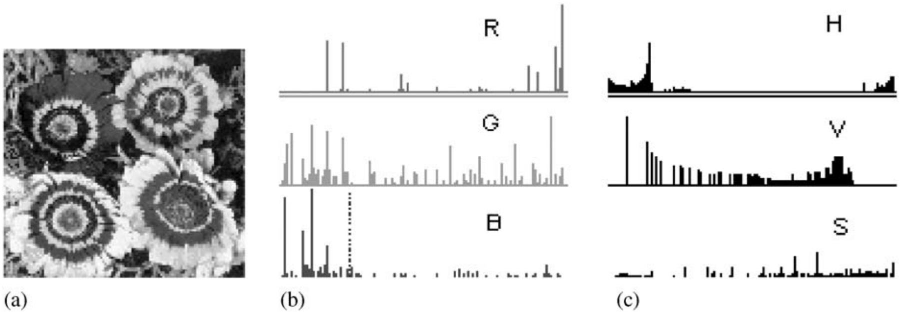

Fig. 2. Color histogram: (a) color flower image; (b) histogram for each channel in RGB; (c) HSV histogram.

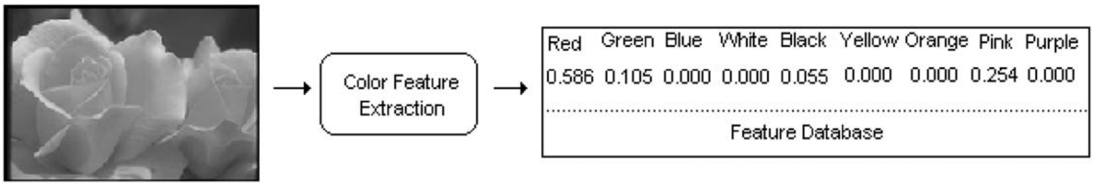

Fig. 3. Preparation of feature database.

channels. More formally, the color histogram is defined by $h_{rgb} = N\ prob\{R = r, G = g, B = b\}$, where R, G, B represent the three-color channels and N is the number of pixels in an image. These RGB values are converted into Hue [0, 360], Saturation [0, 1] and Value [0, 1]. Fig. 2 shows an image of a flower and the extracted histogram for each channel R, G, B and H, S and V.

The color feature is extracted from the image and stored in a database. N_p represents the total number of pixels in an image. For each color value (red, green, blue, etc.) The number of the pixels that belong to the value are recorded and denoted by N_f where $f \in F_u$. Due to the size of the image collection, it is necessary to use a database for managing the image content information. The chief priority is to store this data in such way as to facilitate the fastest possible retrieval time in order to make rapid browsing feasible.

Let N_p represent the number of pixels in each image. The number of the pixels are calculated for each color. The feature component F_R is calculated for each color in an image by the following formula:

$$F_R = \frac{N_R}{N_P}$$

N_R represents the number of red pixels in an image. Similarly, the feature component is calculated for all nine colors (Fig. 3).

3.2. Query interpretation

In most of the current CBIR systems, a query is submitted in the form of an example image and similar images are subsequently obtained from the database. The similarity can be computed on the basis of features of the images such as color, texture, etc. in most cases. These systems do not have

Fig. 4. Query interpretation (mostly, many and few).

the facility of asking for a query in terms of the real content of the image in the form of color, texture and shape, etc. For example, if the user wants images that contain a certain amount of red color and a certain amount of yellow color, the amount of color is asked in terms of percentages. The query can be posed to retrieve all the images that contain 70% of red color and 30% of yellow color. But in natural language, we do not speak in terms of percentages. Therefore, some fuzzy values to interpret queries were proposed in this research.

In some applications, fuzzy systems often perform better than traditional systems because of their capability to deal with non-linearity and uncertainty. One reason is that while traditional systems make precise decisions at every stage, fuzzy systems retain the information about uncertainty as long as possible and only draw a crisp decision at the last stage. Another advantage is that linguistic rules, when used in fuzzy systems, would not only make tools more intuitive, but also provide better understanding of the outcomes.

A relationship is defined to express the distribution of the truth of the variable. Theoretically, a fuzzy set F is the universe of discourse $X = \{x\}$ is the number in the range $[0, \alpha]$, indicating the extent to which x has the attribute F. Thus, if x is the amount of content of the image, "few" may be considered as a particular value of the fuzzy variable "content" and each x is assigned a number in the range 0 to α, $\mu_{\text{content}}(x) : [0, \alpha]$, that indicates the extent to which x is considered to be content: $\mu_{\text{content}}(x) : [0, \alpha]$ is called the membership function. When the membership function is normalized (i.e. $\alpha = 1$), then $\mu_{\text{content}}(x) : X \to [0, 1]$ and the fuzzy logic is called normal.

Referring to Fig. 4, the query to retrieve the images from the database is prepared in terms of natural language, such as *mostly*, *many* and *few* content of some specific color. This approach is to make CBIR systems intelligent so that they can interpret human language.

The first step is to define a simple syntax for queries. In the real world, users do not want to enter many numbers or percentages for colors, shapes, etc. Usually to retrieve images, users prefer to say that they would like to retrieve images that are "*mostly red*" or "*mostly green and a little bit red*". The technique describes these colors in three expressions such as mostly, many and few. So if the user wants to retrieve images which have a large amount of red color content, the query can be defined as "*mostly* red". In a similar fashion, if the user want to retrieve the images that have a small amount of red color content, the query can be posed as "*few* red". The user can able to pose a composite query in terms of colors and content types. For example, if the user is searching for the images that contain large amounts of red color and a little bit of green color, then the syntax for the query is "*mostly* red and *few* green". The general syntax for queries is as follows:

$$\text{QUERY} = \{\{\langle\text{Content}\rangle\langle\text{Color}\rangle\}\langle\&\rangle\{\langle\text{QUERY}\rangle|\langle\,\rangle\}\}$$

where

$$\text{Content} = \{\langle\text{mostly} \mid \text{many} \mid \text{few}\rangle\}$$
$$\text{Color} = \{\langle\text{red} \mid \text{green} \mid \text{blue} \mid \text{white} \mid \text{black} \mid \text{yellow} \mid \text{orange} \mid \text{pink} \mid \text{purple}\rangle\}$$

For example:
A color query to retrieve images can be defined as follows:

QUERY = {mostly red AND few green}

The query to retrieve images from a database is defined in terms of natural language such as mostly ⟨content⟩, many ⟨content⟩ and few ⟨content⟩ of some specific color. There is no better logic than fuzzy logic to interpret such queries.

The user will provide queries in terms of natural language such as *mostly*, *many* and *few*. Therefore, the particular color content for each image is assumed to be "*mostly*", "*many*" and "*few*". In this model, the interpretation domain is a fuzzy set [0, 1]. The ranges of the values used are [0.9, 1] for "*mostly*", [0.4, 0.5] for "*many*" and [0.15, 0.25] for "*few*". Also, the numeric weights such as 0.9 and 0.92 are so close to each other that they both indicate the particular feature is mostly present in the desired images.

$$\mu(content) = \begin{cases} mostly & if \;\; content \in \langle 0.9 \;\; 1 \rangle, \\ many & if \;\; content \in \langle 0.4 \;\; 0.5 \rangle, \\ few & if \;\; content \in \langle 0.15 \;\; 0.25 \rangle. \end{cases}$$

3.3. Learning of queries

To retrieve the required images from the database, it is necessary to learn the meaning of the query, which is in terms of nine colors and three content types such as mostly, many and few. A neural network-based technique is the best solution to learn those queries. Different neural network algorithms can be categorized by, for example, the learning method and architecture of the network. The supervised learning neural network is efficient to learn the colors and content types. The error back propagation neural network is proposed to learn the meaning of those queries.

The number of inputs indicates all nine colors that we have selected for the experiments. The number of outputs is three and it indicates the content type such as *mostly*, *many* and *few*. The number of hidden units is taken as five for our experiment. The training pairs are decided for each content type. There are 99 training pairs for each content, so the total training pairs obtained is 297. Learning rate (η) and momentum (α) are kept as 0.7 and 0.2 respectively for our initial experiments. We performed the experiments by taking 100 iterations initially. This approach is very novel and it overcomes the problem of re-training of neural networks in real world on-line applications where databases get larger everyday, e.g. databases on the Internet. The neural network is trained only once for the queries. For example, the color feature has nine colors. So the network has many inputs, e.g. color feature has nine inputs. The range of minimum and maximum values for the three attributes mostly, many and few are shown in Table 1.

3.4. Fusion of multiple queries

Most existing CBIR systems are based on a single query. Some systems that use more than one query are based on very primitive conventional methods for fusion [15]. Many of them simply treat the composite query as many separate queries and execute them one by one. For example, first the

Table 1
Minimum and maximum values

Mostly	Many	Few
$\text{Min} = 0.9 + \Delta$	$\text{min} = 0.4 + \Delta$	$\text{min} = 0.15 + \Delta$
$\text{max} = 1.0$	$\text{max} = 0.5$	$\text{max} = 0.25$
$\Delta = \Delta + 0.01$	$\Delta = \Delta + 0.01$	$\Delta = \Delta + 0.01$

Table 2
Final confidence factor using binary AND

Confidence factor for first query (mostly red)	Confidence factor for second query (few green)	Final confidence factor
1	1	1 (100%)
1	0	0 (0%)
0	1	0 (0%)
0	0	0 (0%)

Table 3
Final confidence factor using fuzzy AND (first case)

Confidence factor for first query (mostly red)	Confidence factor for second query (few green)	Final confidence factor
0.6	0.8	0.6 (60%)
0.85	0.9	0.85 (85%)

search is performed on all the images in the database for the first query and then the second query is used to search only on the images that are found by the first query. This search can be improved by using neural network fusion. The fusion of the two queries is described below; however, the approaches can be used for more than two queries.

3.4.1. *Fusion of queries using binary AND*

This is explained by taking samples of queries. For example, if the composite query is "*mostly* red AND *few* green", the output of this query is shown using a binary AND (Table 2).

This approach will work, if the possible output of the neural network is 1 or 0. But the output of the neural network is in the range of 0–1, for example 0.8 and 0.4 for the corresponding queries; this approach will fail to give the final confidence factor. Another approach may be considered to fuse the outputs of the neural network, i.e. by the use of Fuzzy Logic AND.

3.4.2. *Fusion of queries using fuzzy AND*

The possibility of getting the final confidence factor for Fuzzy AND on the example query, "*mostly* red AND *few* green" is shown in Table 3.

Table 4
Final confidence factor using fuzzy AND (second case)

Confidence factor for first query (mostly red)	Confidence factor for second query (few green)	Final confidence factor
0.9	0.3	0.3 (30%)
0.3	0.3	0.3 (30%)

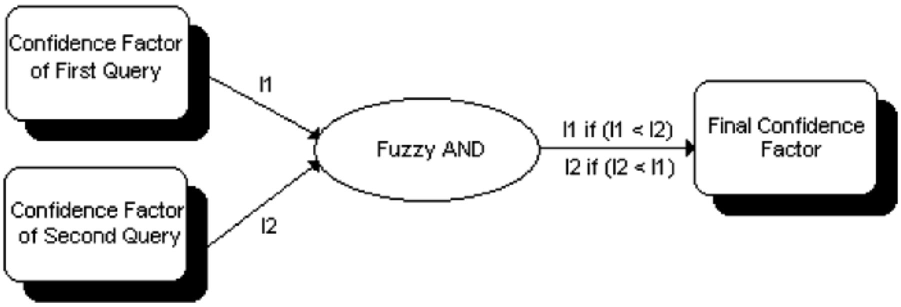

Fig. 5. Fuzzy AND technique.

This approach will work better than the binary AND if the outputs of the neural network for the two queries are similar or having very small difference. But if the difference between the neural network outputs is considerable, then this approach will fail to give an adequate final confidence factor (Table 4).

As shown in Fig. 5, the confidence factor for the first query is I_1 and for the second query is I_2, fuzzy AND approach selects the minimum value as the confidence factor.

From the above results, it is shown that in both cases, the final confidence factor is 0.3, though in first case the confidence factor for the first query is 0.9. This interpretation is not very accurate for different possible outputs of the neural network. The final approach is based on the fusion of the multiple queries using neural network.

3.4.3. *Fusion of queries using neuro-fuzzy AND*

The fusion algorithm is based on a two layer feed-forward neural network. The number of inputs to the neural network is equivalent to the number of queries and the number of outputs is equivalent to one that is the actual output of neural network. The actual output of the neural network is called the confidence factor. The neural network is trained using error back propagation algorithm and it decides the final confidence factor for the combination of the queries. This approach works better than the above-mentioned binary AND and fuzzy AND. Fig. 6 shows the approach using neuro-fuzzy AND technique. The confidence factor for the first query is I_1 and the confidence factor for the second query is I_2. These two factors are applied to neural network as input. The final confidence factor is indicated by O_1 (Table 5).

In the first case, if the confidence factors are 0.9 and 0.3, then the final confidence factor will be decided by the neural network, say *A*.

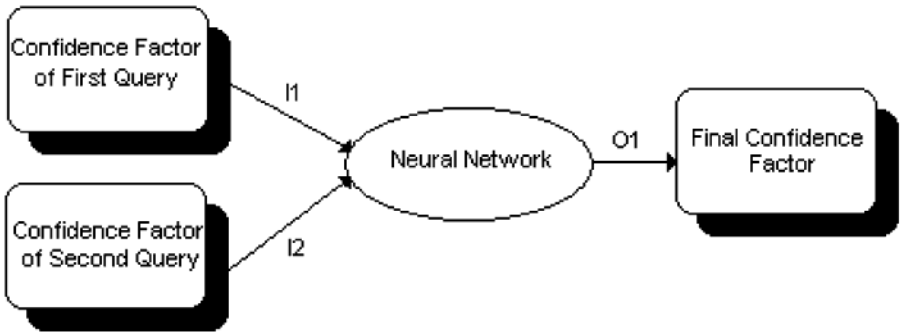

Fig. 6. Neuro-fuzzy AND technique.

Table 5
Final confidence factor using neuro-fuzzy AND

Confidence factor for first query (mostly red)	Confidence factor for second query (few green)	Final confidence factor
0.9	0.3	A
0.3	0.3	B (%A > %B)

In the second case, if the confidence factors are 0.3 for both queries then the final confidence factor is B. But the value of B must be smaller than the value of A. This neural network fusion technique has been implemented and the results are shown in the next section, experimental results.

4. Experimental results

This section outlines all relevant experimental results that were conducted using the proposed techniques from the previous section.

To test the effectiveness of this proposed system, the preliminary experiments were conducted on a single color feature and content type. These results were published in our previous paper [11]. Before retrieving the images based on the single color and content type, images were preprocessed. This preprocessing includes, downloading real world images from the WWW to form the image database. This image database contains a wide variety of images, like the images of flowers, scenery, animals, mountains, etc. Most of the images are in jpg format. All these images were converted into raw format for extracting the color feature from the image. The color feature was extracted from all the images and stored into a database. A separate feature database was prepared to have easy access for the images after posing a particular query.

Table 6
Results of the query using fuzzy AND technique

Query	Image #	Confidence factor for first query	Confidence factor for second query	Fuzzy AND
Mostly green and few black	149	0.832603	0.999992	0.832603
Mostly red and few yellow	137	0.493791	0.999999	0.493791

4.1. *Fusion of multiple queries*

To pose a query in terms of multiple colors and content types, the output of the individual queries were fused. This approach was examined by the fusion of queries using binary AND, fuzzy AND and neuro-fuzzy AND.

4.1.1. *Binary AND*

The confidence factor obtained from the individual queries are fused together to get the final confidence factor. The binary AND technique has two inputs that should be in terms of 1 and 0. This technique fails if the confidence factors have values other than 1 and 0. This technique cannot be used in real world applications. For example, if the confidence factor for the first query is 1 and for the second query is 0, it is not adequate to say that image satisfies that particular query. The technique of binary AND for different sets of queries was tested and found that it is not adequate to use this technique in real world applications.

4.1.2. *Fuzzy AND*

The confidence factors are fused using a fuzzy AND technique. The fuzzy AND is the minimum of two. For the query, mostly green and few black, the confidence factor for the image # 149 was 0.832603 for the first query and 0.999992 for the second query. The fuzzy AND technique gives the confidence factor as the lowest among the two, which is 0.832603. Table 6 shows the fuzzy AND confidence factor for the combination of the query. These results are compared later using a neuro-fuzzy AND technique.

Fig. 7 shows the images retrieved for the query *mostly* green and *few* red. The confidence factors are indicated by f1 and f2 for query *mostly* green and *few* red, respectively. The final confidence factor was indicated by Fuzzy AND. For the image # 124, the confidence factor for the query, mostly green and few red was 0.953675 and 0.899998 respectively. The final confidence factor was 0.899998. As there was not much difference for the two confidence factors, the final confidence factor was acceptable, but for the image # 209, the final confidence factor was 0.574968 although the confidence factor for the second query was almost 1. This technique was tested for different combinations of the queries.

4.1.3. *Neuro-fuzzy AND*

A number of experiments were conducted to test the effectiveness of the neuro-fuzzy AND technique. The neural network was designed for the queries to have two colors and two content types. The neuro-fuzzy AND technique is also suitable for more than two queries. Table 7 shows the

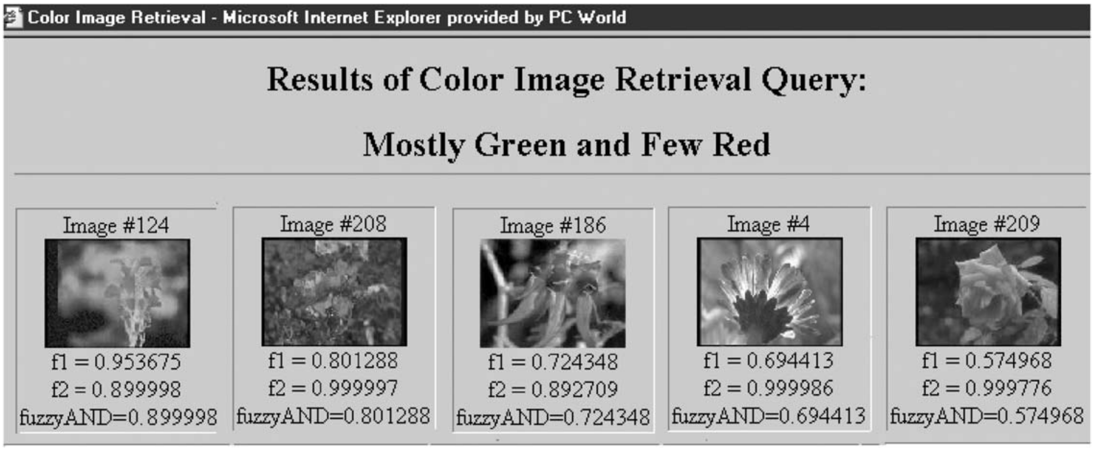

Fig. 7. Results of the query: *mostly* green and *few* red using fuzzy AND technique.

Table 7
Neural network parameters for fusion

Number of inputs	Number of outputs	Hidden units	Learning rate	Momentum	RMS error
2	1	4	0.7	0.6	0.003244

different parameters used to train neural network. The results obtained after posing different queries are described in the following sections.

4.1.3.1. Query: mostly and many contents. The query can be posed in terms of mostly and many contents of colors. The results displayed for the query: *mostly* green and *many* blue are shown in Fig. 8. The images are mainly of green trees/lawn at bottom and blue sky at the top of the image.

4.1.3.2. Query: mostly and few contents. If the user wants to retrieve images from database such as *mostly* red and *few* orange, the two colors are selected from the pull-down menus. Similarly content types such as Mostly, Few are selected from another pull-down menu. In Fig. 9, Image #96 contains a good combination of *mostly* red and *few* orange from all the images in the database, so this image has a maximum confidence factor of 0.967684. All other images are obtained in descending order with their confidence factor.

The results of the fuzzy AND and neuro-fuzzy AND techniques are compared in Table 8. It is seen that the confidence factor was improved significantly using the neuro-fuzzy AND technique. The proposed technique of neuro-fuzzy AND was tested for the same query used for fuzzy AND. The final confidence factor for each image was improved. In Fig. 7, the final confidence factor obtained for image # 4 using fuzzy AND technique was 0.694413. For the same query, mostly green and few red, the confidence factor was improved to 0.970758 using neuro-fuzzy AND technique. The images retrieved for this query are shown in Fig. 10. The improvement in the final confidence factor was observed for each image retrieved for the different combinations of the queries.

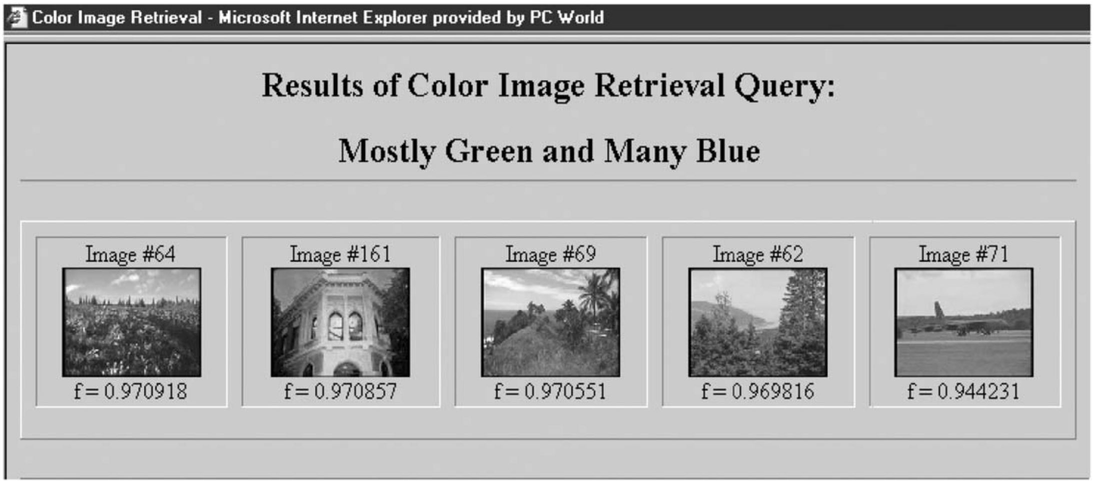

Fig. 8. Results of the query: *mostly* green and *many* blue.

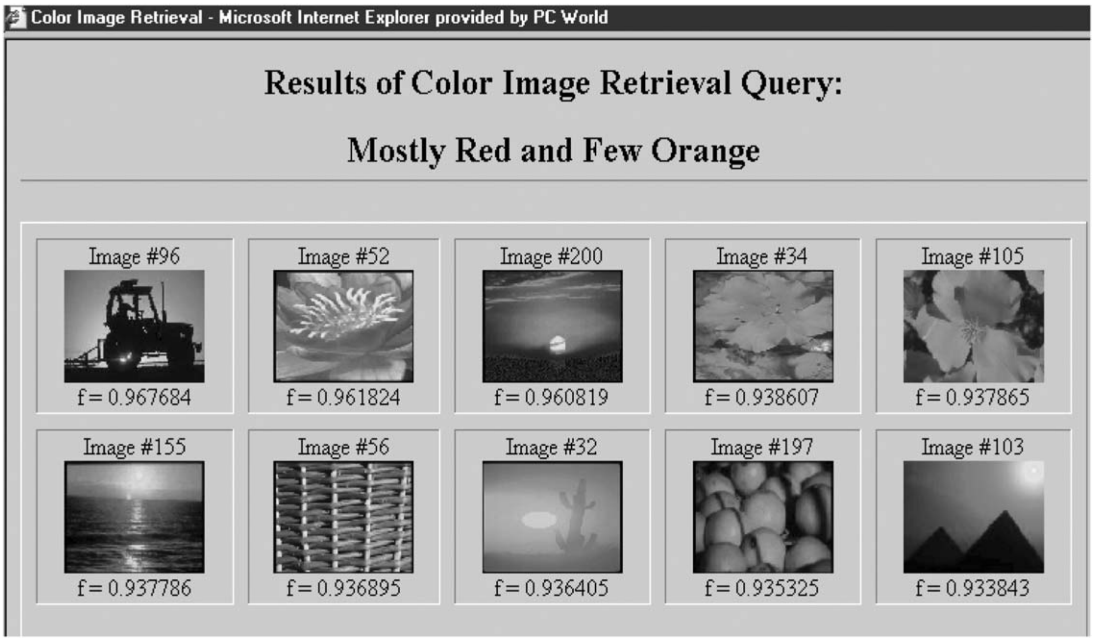

Fig. 9. Results of the query: *mostly* red and *few* orange.

4.1.3.3. Query: many and many contents. The results of the query, many contents of red color and many contents of green color are shown in Fig. 11. From the image database of 210 images, five images were shown those have the confidence factor greater than 0.9. The image #25 has the highest confidence factor and appeared first. All the images contain red and green color to certain extent. This type of the query showed that the user could pose the query to retrieve the images those contain certain amount of particular colors in an image.

Table 8
Results of the query using fuzzy AND and neuro-fuzzy AND techniques

Query	Image #	Confidence factor for first query	Confidence factor for second query	Confidence factor fuzzy AND	Confidence factor for neuro-fuzzy AND
Mostly green and few black	149	0.832603	0.999992	0.832603	0.965888
Many red and few yellow	137	0.493791	0.999999	0.493791	0.921801

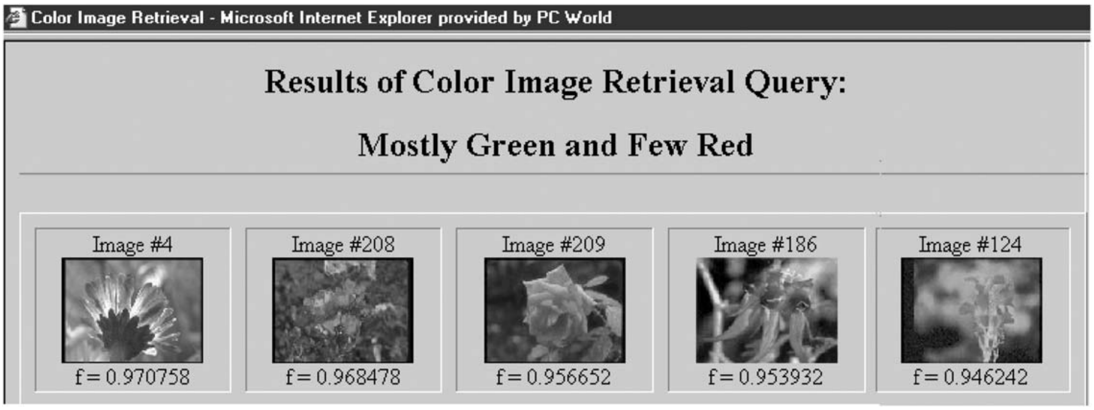

Fig. 10. Results of the query: *mostly* green and *few* red using neuro-fuzzy AND technique.

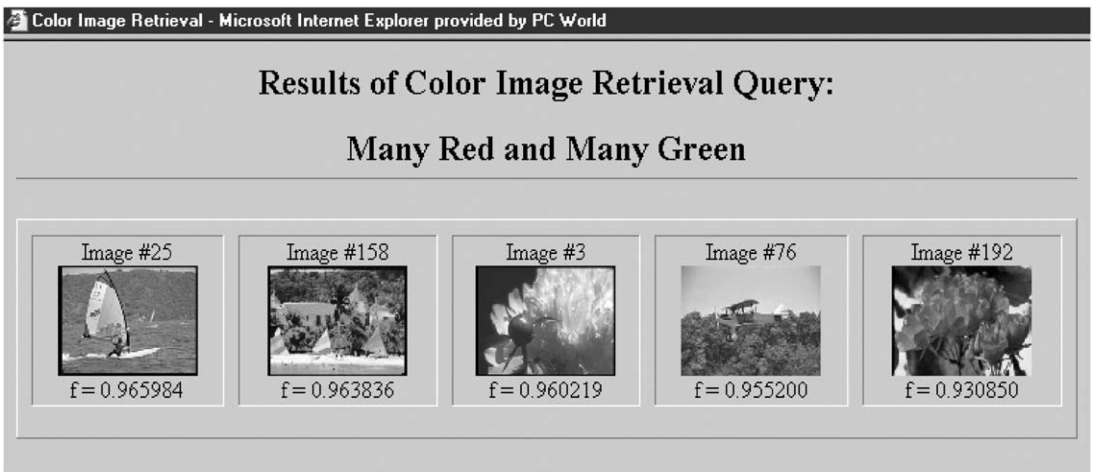

Fig. 11. Results of the query: *many* red and *many* green.

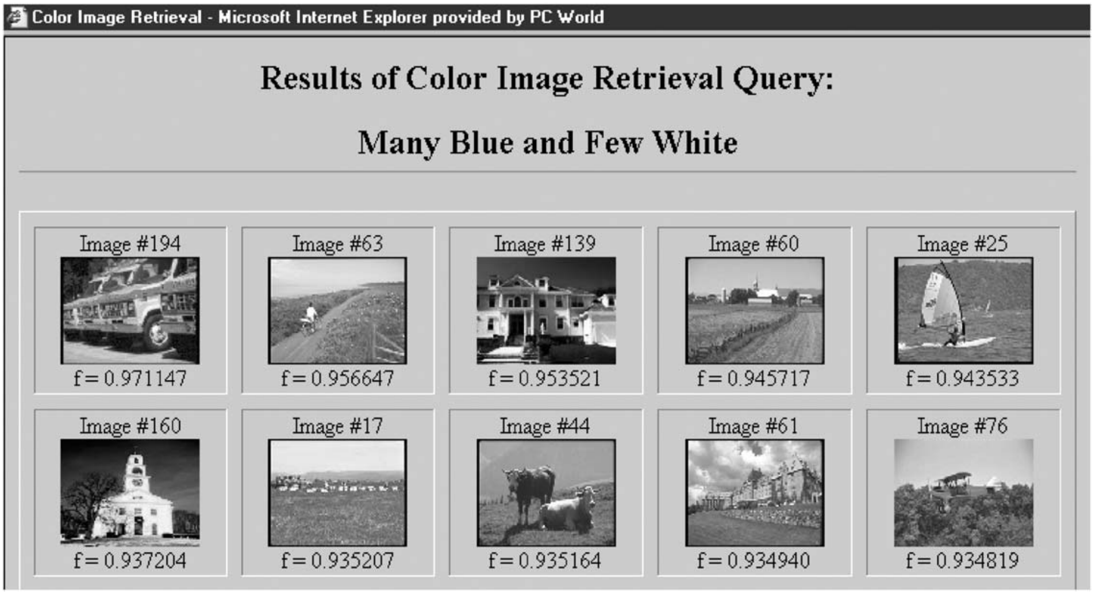

Fig. 12. Results of the query: *many* blue and *few* white.

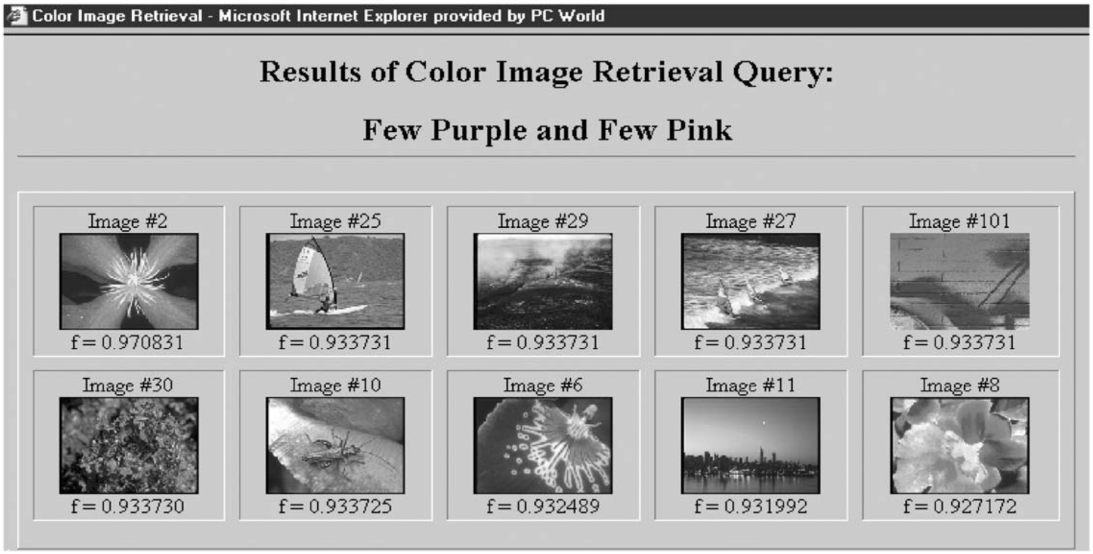

Fig. 13. Results of the query: *few* purple and *few* pink.

4.1.3.4. Query: many and few contents. The query can be posed in terms of many and few content of colors. Fig. 12 shows the results obtained after submitting the query *many* blue and *few* white.

4.1.3.5. Query: few and few contents. Fig. 13 shows the results obtained after submitting the query, *few* purple and *few* pink. Ten images retrieved which have the confidence factor greater than

0.9. Image # 2 is best suited for this particular query with a confidence factor of 0.970831. The experiments were conducted for all combinations of queries and the confidence factor was improved using the neuro-fuzzy AND technique.

5. Analysis and comparison

The purpose of this section is to analyze the experimental results obtained in this research and to discuss their significance. The results obtained for color image retrieval are discussed on overall findings and different issues. The experiments were conducted for all query combinations; some of the experimental results are shown in the previous section. The results are compared with IBM's Query By Image Content project [16].

5.1. Query methods

The use of natural language for posing the query in terms of content of the image colors produced promising results. In most of the CBIR systems, the query is posed in the form of an example image. The task of the system is then to retrieve the images most similar to the given references. This approach is known as Query by Pictorial Example (QBPE) [4]. One drawback with the QBPE is that the success of query considerably depends on the initial set of images. The image retrieval systems may also allow the user to provide a separate example image outside the database. This requires the system to index the example image on-line in order to be able to find the similarities between that image and the images in the database. Queries for retrieving specific objects in an image were also posed by a number of the researchers [18], such as "find red cars" or "show the images of sunset", however image retrieval entirely depends on the keywords [20].

In a visual feature query, the user specifies certain values in percentage for the color feature using sliding bars such as "retrieve images that contain 75% red color and 25% green color". However posing a query in terms of natural language is more user friendly than posing it in terms of numerical values. The proposed fuzzy approach helped the user to pose a query in terms of natural language such as mostly, many and few content of a particular color in an image.

CBIR systems such as VisualSEEK [19], Netra [12] obtained results by posing spatial region queries of an image. In most of the cases, this technique works well but in some cases it fails. If the user wants to retrieve images of sunset, the query was formulated to have yellow and orange color at the top, it retrieves some images of sunset along with some images with yellow and orange colored flowers.

Fig. 14 shows a screen shot using IBM's Query By Image Content project [5]. The query is asked in terms of percentage of two colors. This screen shot shows a query to retrieve images that contain 80% green color and 20% red color. The colors can be selected from a color wheel. This system has a good facility of posing queries for multiple colors chosen from the color wheel (Fig. 15).

The query was posed in terms of percentage of colors and not in terms of natural language. Most of the images retrieved contain green and red colors but some of the images did not contain green color while some of the images contained very small amount of green color. The system does not have the facility for providing numerical values, which could provide more information about the confidence of that image for a particular query.

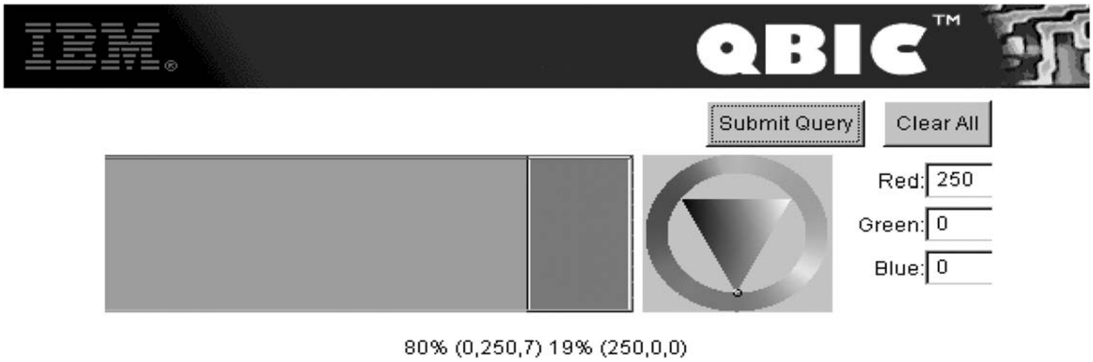

Fig. 14. Screen shot of IBM's QBIC.

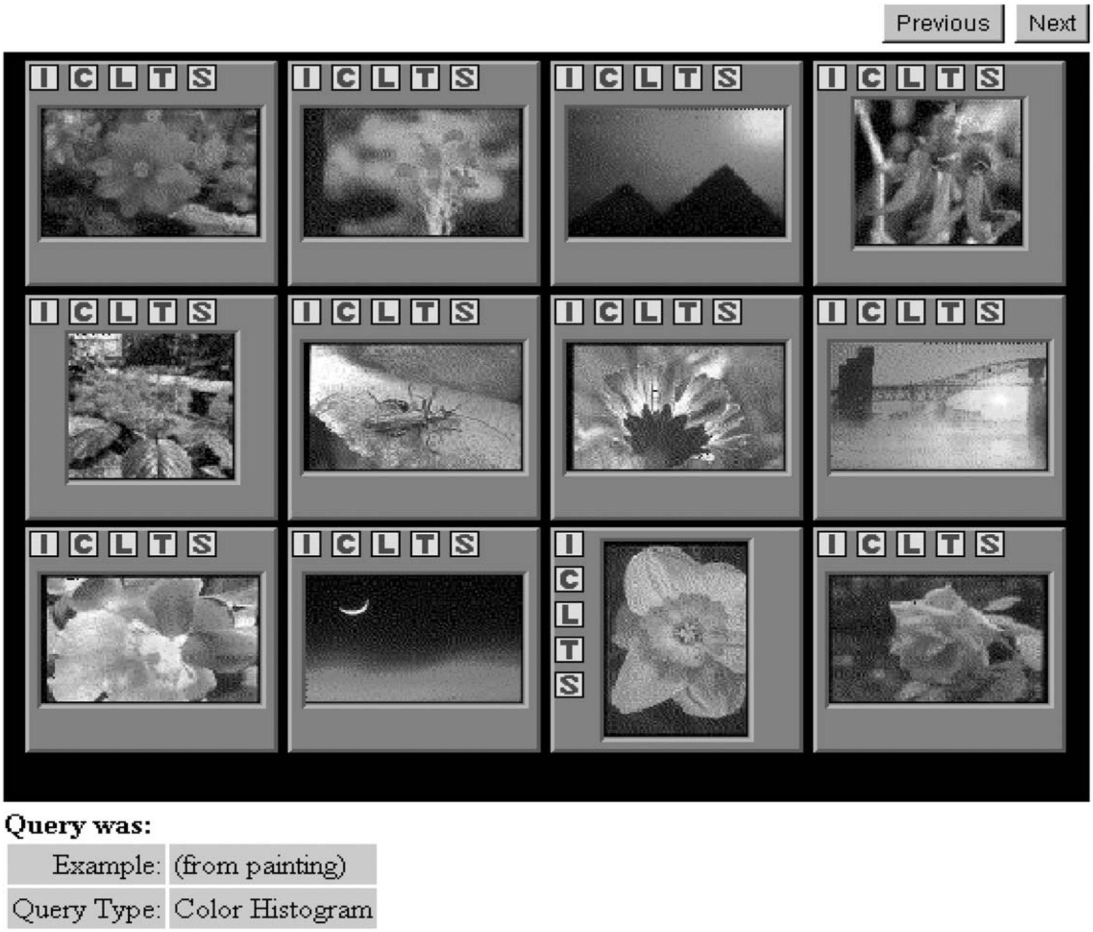

Fig. 15. Results of the query: 80% green color and 19% red color.

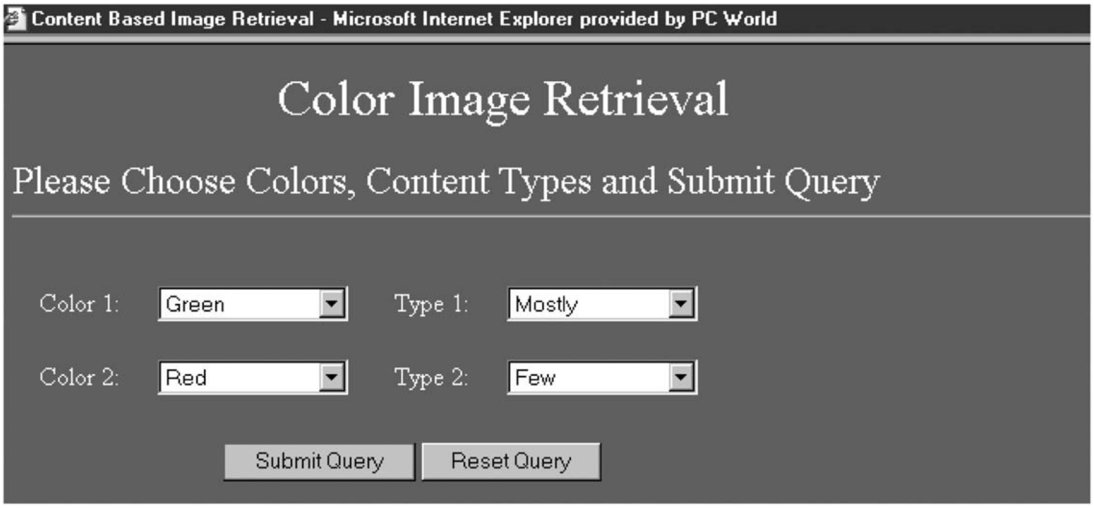

Fig. 16. Screen shot for query: *mostly* green and *few* red.

Fig. 16 shows the screen shot developed for the proposed system to pose the query in terms of natural language. Multiple colors and content types can be chosen from a pull-down menus and the query can be submitted. It is very easy for the user to select the colors and the content types rather than adjusting the colors in terms of percentage values and the combination of primary colors such as red, green and blue. The query can be requested for maximum six colors in case of QBIC, but in the proposed system the query can be formulated for two colors. But this technique can be extended for more than two colors using fusion of the queries.

Fig. 17 shows the results obtained after posing the query, *Mostly* green and *Few* red. All the images have mostly green and few red color in them. There are total eight images which have the confidence factor greater than 0.9. The first image has a confidence factor of 0.970758 and it is best suited for this query. The images retrieved by QBIC do not have any numerical value or any factor that may suggest that how that image is confidence for that particular query.

5.2. Fusion of multiple queries

Most CBIR systems allowed a user to pose the query in terms of only a single color. In a few systems, the user can pose a query in terms of multiple colors. The search was performed on the first query and the second query was posed on the images that were retrieved as a result of the first query. By incorporating the fusion of multiple queries, it was seen that there was a prominent increase in the confidence factor for the images. The neuro-fuzzy technique was used to fuse the multiple queries and the results were discussed and compared with fuzzy AND and binary AND techniques for fusion.

The binary AND technique is suitable to fuse queries if the confidence factor for the queries are 0 or 1, but it was seen that the actual output of a neural network is a value in between 0 and 1. Therefore, this technique is not suitable to use for real world color images. Fuzzy AND technique worked satisfactorily if there was not a significant difference between the two confidence factors.

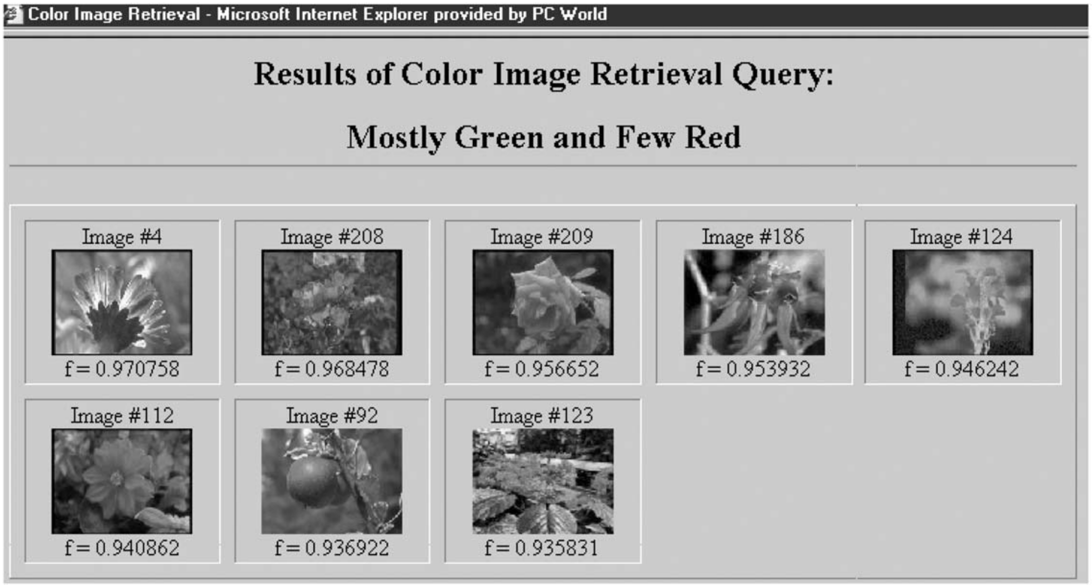

Fig. 17. Results of the query: *mostly* green and *few* red.

Table 9
Comparison of fuzzy AND and neuro-fuzzy AND techniques

Image #	Confidence factor for the first query	Confidence factor for the second query	Final confidence factor using fuzzy AND	Final confidence factor using neuro-fuzzy AND
124	0.953675	0.899998	0.899998	0.946242
208	0.801288	0.999997	0.801288	0.968478
186	0.724348	0.892709	0.724348	0.953932
4	0.694413	0.999986	0.694413	0.970758
209	0.574968	0.999766	0.574968	0.956652

By proposing the neuro-fuzzy technique, it was seen that there was improvement in the confidence factors for the images of that particular query. Table 9 shows the results obtained after fusing the confidence factor using fuzzy and neuro-fuzzy techniques for the query: mostly green and few red.

The confidence factors for the top five images were compared and it was observed that there was significant improvement in the final confidence factor using the neuro-fuzzy AND technique. Similarly, different combinations of queries were posed using fuzzy AND and neuro-fuzzy AND techniques. The proposed neuro-fuzzy technique performed better than fuzzy AND technique.

6. Conclusion

This paper has proposed and investigated a novel approach for retrieving the images based on their contents. The system was implemented using CGI script in C language. The performance of the system was evaluated on real world images.

For color image retrieval, the single color approach allowed the user to ask the query in terms of specific color and content type. The results for the single color and content type worked well and achieved very promising results. The user has the facility to ask the query in terms of natural language such as mostly, many and few color content of an image. Only one neural network was trained on the query for color and content type. It solved the problem of re-training of the network if there is a change in database. This was one of the unique features of proposed research. This technique has been extended to ask the query in terms of multiple colors and multiple content types. The new technique for fusion of the queries was introduced and it was implemented for two colors and their content types. One more neural network was trained on possible confidence factors obtained from the two queries to get the final confidence factor. The results of neuro-fuzzy technique were compared with fuzzy AND and binary AND technique. It was observed from the results that neuro-fuzzy AND technique has improved the confidence factor compared to fuzzy AND technique. A prototype image retrieval system that enables the search for specific colors and content types has been built for natural color images and demonstrated the promising research results. One de-merit of the proposed technique is that user cannot pose a query in terms of two mostly content. The future work will incorporate fusion of multiple queries for color and texture features of an image.

References

[1] R. Brunelli, O. Mich, On the use of histograms for image retrieval, Proc. IEEE Internat. Conf. on Multimedia Computing and Systems, June 1999.

[2] C. Carson, S. Belongie, H. Greenspan, J. Malik, Region-based image querying, Proc. IEEE Workshop on Content-based Access of Images and Video Libraries, June 1997, pp. 42–49.

[3] C. Carson, V.E. Ogle, Storage and retrieval of feature data for a very large online image collections, Bull. IEEE Comput. Soc. Tech. Committee Data Eng. 19 (4) (1996) 19–27.

[4] N.S. Chang, K.S. Fu, Query-by-pictorial-example, IEEE Trans. Software Eng. 6 (6) (1980) 519–524.

[5] W. Cody, L. Haas, W. Arya, M. Carey, R. Fagin, M. Flickner, D. Lee, D. Petkovic, P. Schwarz, J. Thomas, M. Roth, Querying multimedia data from multiple repositories by content: the garlic project, Proc. Third Working Conf. on Visual Database Systems, Lausanne, Switzerland, March 1995, pp. 17–35.

[6] D. Dannels, D. Campenhout, W. Niblack, W. Equitz, R. Barber, E. Bellon, F. Fierens, Interactive outlining: an improved approach using active conyours, Proc. Storage and Retrieval for Image and Video Databases, 1993, pp. 226–233.

[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, IEEE Comput. 28 (9) (1995) 23–32.

[8] V. Gudivada, V. Raghavan, Content-based image retrieval systems, IEEE Comput. Mag. 9 (28) (1995) 18–22.

[9] A. Gupta, Visual information retrieval: a virage perspective, Technical Report Revision 4, Virage Inc., San Diego, CA, 1996, URL: http://www.virage.com/wpaper.

[10] A. Gupta, R. Jain, Visual information retrieval, Commun. ACM 40 (5) (1996) 70–79.

[11] S. Kulkarni, B. Verma, P. Sharma, H. Selvaraj, Content based image retrieval using a neuro-fuzzy technique, Proc. IEEE Internat. Joint Conf. on Neural Networks, Washington, USA, 1999, pp. 846–850.

[12] W. Ma, B. Manjunath, Netra: a toolbox for navigating large image databases, Proc. IEEE Internat. Conf. on Image Processing, 1997, pp. 568–571.

[13] B. Mehtre, M. Kankanhalli, A. Narasimhalu, G. Man, Color matching for image retrieval, Pattern Recognition Lett. 16 (3) (1995) 325–331.

[14] S. Nepal, M. Ramakrishna, J. Thom, A fuzzy system for content-based image retrieval, Proc. IEEE Internat. Conf. on Intelligent Processing Systems, 1999, pp. 335–339.

[15] S. Nepal, M. Ramakrishna, J. Thom, A Fuzzy Object Query Language (FOQL) for image database, Proc. Sixth Internat. Conf. on Database Systems for Advanced Applications, April 1999, pp. 117–124.

[16] W. Niblack, The QBIC Project: querying images by content using color, texture and shape, Proc. Storage and Retrieval for Color and Image Video Databases, 1993, pp. 173–187.
[17] A. Ono, M. Amano, M. Hakaridani, T. Satou, M. Sakauchi, A flexible content-based image retrieval system with combined scene description keyword, Proc. IEEE Computer Society, Internat. Conf. on Multimedia Computing and Systems, 1996, pp. 201–208.
[18] J. Smith, S. Chang, An image and video search engine for the World Wide Web, Proc. Storage and Retrieval for Image and Video Databases V, 1997, pp. 84–95.
[19] J. Smith, S. Chang, Querying by image regions using the VisualSEEK content-based visual query system, J. Intelligent Multimedia Inform. Retrieval (1997) 23–41.
[20] L. Zhu, C. Tang, A. Zhang, Using keyblock statistics to model image retrieval, Proc. IEEE Pacific-Rim Conf. on Multimedia, 2001, pp. 522–529.