An Intelligent Distributed Algorithm for Efficient Web Image Retrieval

Alaa M. Riad, Ahmed Atwan, Hazem M. El-Bakry, Sameh Abd EL-Ghany, Nikos Mastorakis

Abstract: - Web image retrieval is a challenging task that requires efforts from image processing, web text retrieval, and link structure analysis. General web image retrieval engines such as Google and Yahoo retrieve images according to the meta-data or the context in which the images are found. Hence, the results search engines are far from expected regarding the semantics of the images and the user can be overwhelmed by thousands of results, of which few are valuable. In order to become an effective complement to traditional Web-scale text-based image retrieval, content-based image retrieval must address scalability and efficiency issues. This paper presents a framework for distributed content based web image retrieval. The proposed framework is based on multi-agent system where agent autonomously searches for images over the internet to find images matching a given example. During image retrieval, the system can decide an optimized retrieval strategy by automatically analyzing the query input, thus a search agent is routed only to the image sources that maintaining categories of images similar to the query in content (color and shape). The mobility enables the search agent to perform local image match at image sources, which is expected to improve retrieval speed and reliability by minimizing network traffic and dependence on network status.

Keywords: - Image Retrieval, Mobile Agent, Distributed System

I. Introduction

Search engines are the most powerful resources for finding images on the rapidly expanding World-Wide Web. These search engines use manual annotation for describing the content of an image and rely on text retrieval techniques for searching particular images [1]. However, such annotation has three major drawbacks. First, the annotation depends on the person who adds it. Different users may perceive different meanings from the same image. Even if the two users do have the same perception of an image, they may use different keywords to annotate the image, depending on the individual vocabularies [2,3]. Naturally the result may vary from person to person and furthermore may depend on the context. The second drawback with manual annotation is that it is very time consuming. While it may be worthwhile for commercial image collections, it is prohibitive for indexing of images within the World Wide Web. One could not even keep up with the growth of available image data. Third drawback is that the user of a Text Based Image Retrieval must describe an image using nearly the same keywords that were used by the annotator in order to retrieve that image. Due to all these drawbacks, Content Based Image Retrieval is introduced [4].

The main idea of content based image retrieval (CBIR) is to retrieve within large collections images matching a given query to their visual content analysis. Visual features, such as color, texture or shape, are extracted from the images and indexed [5]. A basic way to perform retrieval consists in computing a similarity function for comparing the query feature the images features in the collection. With the increasing scale of image databases, centralized content based image retrieval system no longer provide sufficient quick search. By extending the centralized system model, it can increase the size of image collections and overcome the scalability bottleneck problem by distributing the process of image retrieval among different nodes [6-9].

In the distributed image retrieval scheme, images are spread into several well known collections. This is advantage since the processing of every image could be naturally paralleled. However, there are two challenges for distributed CBIR [10,11]; the first challenge is image source selection when searching large numbers of distributed image sources. It is likely that only a relatively small number of image sources may contain image relevant to the query. It is obvious (but wasteful) to query all image sources for a list of similar images. The second challenges in

Manuscript received December 31, 2008.
Alaa M. Riad, Ahmed Atwan, Hazem M. El-Bakry, Sameh Abd EL-Ghany, Faculty of Computer Science & Information Systems, Mansoura University, EGYPT.
Nikos Mastorakis, Technical University of Sofia, BULGARIA.

distributed CBIR is result merging, where a query Q to find top k best results is sent to n image sources which returns the k best image results. Thus, a total of $k * n$ results will be returned. The results are merged from all image sources and the top k results are return to the user. It is too expensive to get $k * n$ images and re-process them to select the globally top k results, in particular when n or image sizes are large.

This paper presents an approach for distributed Content based web Image Retrieval. The proposed approach is based on agents technology where agents autonomously search for images over the internet, then the image features (color and shape features) are extracted from the images and stored in the database. The extracted features are used as the basis for grouping the images servers that contain similar images categories together. The proposed approach aims at reducing the network traffic by avoiding the transfer of huge intermediate data through the network. The proposed approach is based on a cluster algorithm by which the similar images at each image source are categorized together into a cluster. The centroids of these clusters are sent to a central directory server. Fortunately, the proposed approach reduces message transfer as there is no need to update the cluster centroids with each new image. Instead, the need for a new calculation of the clusters arises after a significant number of updates. During the search process, a mobile agent is routed only to the image sources that maintaining categories of images that similar to the query in color and shape features. On each image source, mobile agent will match the query features against the features of images that are stored in the database. After obtaining the result from the image sources, the retrieved results are filtered to remove the irrelevant results using self organizing map neural network. The results containing the URL, as well as, the thumbnail of the similar images are transmitted to the user and the results are ranked according to their similarity to the submitted query.

II. CBIR over WWW using Mobile Agents

A. *System architecture*

Figure 1 shows the components of the proposed system. The proposed system consists of three main components: image source, directory server, client machine.

1) *Image source*

It composed of web crawler, Feature agent, Image Clustering Agent, images database.

a) *Web Crawler*

The Web Crawling process is conducted by a different autonomous web crawlers run on different computers. Its major tasks are:

- Traverses the Web by parsing the hyperlinks.
- Extracts the URLs of the images.
- Download images and store them as well as its related information in the images database.

b) *Feature Agent*

This agent resides in the image source. The major tasks it performs are:

- Read Image from images database.
- Extract features from the image.
- Store the feature vectors in the images database.
- Generate a thumbnail icon that sufficiently compacts and represents the visual information.

c) *Image Clustering Agent*

This agent resides in image source to perform these tasks:

- Carries out image clustering to speed up the retrieval of images.
- Sends the centroid of each cluster to the directory server to be stored in the centroids database.

d) *Images Database*

Images database contains information about the collected images from the internet. The images database contains image table that contains the following columns:

- ImageID: image source manages the images using unique ID which is the primary key.
- URLSite: the URL of site that contain the image.
- Image_ Name: it is the name of image in the web site.
- Image: in this column, the original image will be stored.
- Size: this is usually useful for those who wish to screen out images smaller than a certain size.
- Thumbnail: it is an icon for the image (resized image to small size). It will be returned to the user as initial result. If the user wants to retrieve the full image size, he/she will be redirected to the site that holds this image.
- ColorFeature: it is a vector that represent color feature.
- ShapeFeature: it is a vector that represent shape feature.

2) *Directory Sever*

Directory server serves as a common gateway, linking users to multiple image sources. The directory server receives the user query, forwards the user query to appropriate images servers. The directory server contains the following components:

server clustering agent, centroids database, feature agent, search agent, routing agent, and ranker agent.

a) Server Clustering Agent

The server clustering agent groups the images servers that hold the similar images categories together. The server clustering agent reads the centroids of each image category from centroids database and perform k-means clustering algorithm on those centroids.

b) Centroids database

The centroids database Contain the following tables:

1. Server Information Table: this table contains information about the image sources. The server information table contains these columns:
   - *ServerURI*: it represents the Unified Resource Identifier of the image source. This column represents the primary key of this table.
   - *PortNumber*: The port number represents TCP channel that *image source* is listening to.
2. Image_Centroids Table: this table contains information about the categories of images on each image source. This table contains the following columns:
   - *ServerURI*: it is a foreign key for the ServerURI that perform relationship with server information table.
   - *Image_ClusterID: Since each image sources may contains more than one image cluster, then each cluster will be identified using unique Image_ClusterID. Both ServerURI and ClusterID values composes the primary key.*
   - *ImageCentroid*: is the centroid vector of the image cluster.
3. CentroidGroup Table: in this table the centroid of each server group will be stored as well as the GroupID that is the primary key of this table. The columns of this table are:
   - *GroupID.*
   - *CentroidGroupVector.*
4. ServerGroup Table: this table contains information about the group that each image source belonging to. This table contains the following columns:
   - *ServerURI*: it is a foreign key for the ServerURI that perform relationship with server information table.
   - *GroupID*: the group that image sources belonging to.

c) Feature Agent

Extract the features from the query image using specific feature extraction method which selected by the user and pass the extracted feature vector to the routing agent.

d) Routing Agent

The main task of the routing agent is to retrieve the image sources URIs records where image sources are available and contain image clusters that match user query. The routing agent will perform the following steps to identify the list of servers to be visited.

- Receive the feature vector from the feature agent
- Connect to the Centroids database.
- Calculate the distance between the query feature vector and the CentroidGroupVector that are stored in the centroid Group table.
- Select the Server GroupID that represents minimum distance between query and centroid group
- Select the URIs of the image sources in the specified group
- Rank the image sources according to the similarity between image cluster centroid and the query and add those URI of the ranked servers in a queue.
- Generate and dispatch the search agent to the first image sources in the routing list.

e) Search Agent

It is a mobile agent which moves to one or more image sources to search for images that match the query. The search agent performs the following tasks:

- Execute matching code on the image's feature vectors and retrieves similar images and merges previously collected results with the results of its local search.
- Update the query parameters (number if required images and similarity threshold) according to the collected results.
- Finally, Transports the result set back to the ranker agent.

f) Ranker Agent

The main task ranker agent performs are:

- Merges the result from different search agents.
- Ranks the result according to their similarity to the query.
- Finally it passes the ranked results to the user interface agent.

3) Client Machine

The client machine contains a User Interface Agent which major tasks are:

- Enable user to specify the query by means of an example image with a number of top ranked required results.
- Enable user to select the preferred feature extraction methods.
- Send the query to the directory server.

- Display the results to the user.

B. *Processes Workflow*

The proposed system is split into the following processes:

1. Web Crawling Process.
2. Feature Extraction Process.
3. Image Clustering Process.
4. Server Clustering Process.
5. Search Process.

The Web Crawling, feature extraction and clustering processes are performed initially as a preprocessing phase while the search process is performed when the user is online. Figure 2, shows the preprocessing phase.

The next section explains each process in details.

1) Web Crawling Process

The Web Crawling process is conducted by a multiple web crawlers run on a set of computers. The Crawler agent performs a depth-first search for image links on potential websites (initially only URL's ending with .edu, .gov and .net are used in the implementation, crawler for each sites extension). When an agent receives a URL to crawl as input, it looks for all potential URL's in the page. However since it does a depth first search, it recursively keeps crawling down every link that it traverses. A depth value is used as a cutoff point to stop the crawling process. At this point, it has a collection of all links that it just crawled. Figure 3, describes the basic components of the web crawler.

The web crawler consists of these components:

1. *Web Page Downloader*: This component reads a list of URLs and makes HTTP requests to get those web pages.
2. URL Extractor: This component is responsible for extracting URLs from a *downloaded* web page.
3. URL Parser: this component parser the URLs, pass the image URL to image *downloader* to be downloaded and pass the HTML URL to the duplicate checker.
4. Duplicate Checker: this component prevents the same URL being crawled *again* by checking the list of URLs that have already been crawled. If the URL page is not crawled yet, the duplicate checker adds it back to the URL Queue which is a list of URLs of Web pages that will be crawled.
5. Image Downloader: download and store the image and information about image such as *URL* of the image site and image type in the images database.

2) Feature Extraction Process

In feature extraction process, both color and shape features are extracted from the images. Using just one feature extraction method may cause inaccuracy compared with other systems that use more than one feature extraction methods.

a) Color Feature Extraction

The color feature is extracted using color histogram method. Color histogram is popular because they are trivial to compute, and tend to be robust against small changes in object rotation and camera viewpoint [12,13]. The color histogram represents an image by breaking down the various color components of an image and extracts the three histograms of RGB colors; Red *(HR),Green (HG),and Blue(HB)*, one for each color channel by computing the occurrences of each color (histogram). After computing the histogram, the histogram of each color is normalized because the images are downloaded from different sites which maintain images with different size. The value of each color component is normalized as follows:

$$Color\_normalized=H_x/M*N \qquad (1)$$

Where:

- Hx is the histogram for specific color channel (x: R, G, and B).
- M, N are the dimensions of the image. Dividing the histogram by M*N normalizes the values of the histogram so that they can be compared among images of different size.

After getting the histogram of the image, the extraction agent will store it in the images database.

b) Shape Feature Extraction

A good shape representation of an image should be handle changes in translation, rotation, scaling; In this process, the *Mountain-Climbing Sequence (MCS)* [14], is used as a shape feature extraction because it is invariant to translation, rotation, and scale.

This section details the step-by-step that are required to perform shape feature extraction method using MCS:

1. Connect to the images database.
2. Read image.
3. Transfer image from RGB color to Binary Image: This step is performed to ignore the background information of the image.
4. Boundary extraction: In this step, the Contour based method is used to extract the boundary of object.
5. *Determine the central point of the object*: To permit invariance to translation, rotation and scaling, the geometric center of the object is selected as a reference point. The center

coordinates of the object is determined as:

$$xc = \sum_{i=1}^{n} x_i / n \tag{2}$$

$$yc = \sum_{i=1}^{n} y_i / n \tag{3}$$

Where x_i is the X coordinate and y_i is Y coordinate in each object's boundary point and n is total number of points in the boundary of the object.

6. *Re-sampling the boundary of shape:* By varying the number of sampled points, the accuracy of the shape representation can be adjusted [15]. The larger the number, the more details the shape is represented, consequently, the matching result will be more accurate. In contrast, a smaller number of sampled points reduces the accuracy of the matching results, but improves the computational efficiency. There are generally two methods of normalization (i) equal points sampling; and (ii) equal angle sampling; assuming k is the total number of candidate points to be sampled along the shape boundary. The equal angle sampling selects candidate points spaced at equal angle $\theta = 2\pi/k$. Figure 4.a and Figure 4.b illustrate the distance and angle of the contour point relative to the center. The equal points sampling method selects candidate points spaced at equal number of points along the shape boundary. The space between two consecutive candidate points is given by z/k, where *z* is the total boundary points.

7. *Extract the shape features:* Features of image are extracted by tracing boundary of the object to determine the set of different distance between the center of the object and the boundary where distance is given by:

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2} \tag{4}$$

Where d is the distance between the corresponding boundary point and the central point of the object.

- *To provide scale invariance*, the maximum distance is computed and all distances are normalized to it (divided by it). Thus, all values fall between 0 and 1 regardless of how much the objects are scaled.
- *To achieve rotation-invariance*, determine the index, s, of the distance function having the smallest value. The distances in the sequence D are then shifted forward for s positions to yield a new sequence D= ($d_s, d_{s+1}, \ldots, d_{s+n-1}$).

8. Store the extracted features in the database.

9. Repeat steps from 2 to 8 for all images present in the images database.

3) Image Clustering Process

Clustering means partitioning data into a set of clusters so that data items in a certain cluster are more similar to each other than to data items in other clusters. After clustering, each cluster is represented by its centroid and the query feature vector is compared to the centroids or the clusters. The best cluster or clusters, according to the used similarity measure, are then selected and the images belonging to those clusters are evaluated and k nearest neighbors is returned. One of the most popular and widely clustering methods that minimize the clustering error for points in Euclidean space is K-means clustering algorithm. The k-means algorithm is used to perform the clustering process. This choice was mainly motivated by the comparably fast processing of the k-means algorithm compared to other unsupervised clustering methods [16].

K-means can be described as a partitioning method. It is an unsupervised clustering method that provides k clusters, where k is fixed a priori. K-means treats each observation in data as an object having a location in space. It finds a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. First k points are chosen as centroids (one for each cluster). The next step is to assign every point from data set to the nearest centroid. After that, new centroids of the clusters resulting from the previous step are calculated. Several iterations are done, in each the data set is assigned to the nearest new centroid. The Image Clustering Agent on each image source will perform image clustering and compute the centroid of each cluster, which is the arithmetic mean of the feature vectors which stored in this particular server. After clustering the images, the Image Clustering Agent will send the centroid, ClusterID and ServerURL to the server clustering agent which stores it in the centroids database.

4) Server Clustering Process

In this process, the server clustering agent performs k-means clustering algorithm on the centroids of image categories to group the image sources that holds the similar image categories together. The centroid of each servers group will be stored in the Server-Group table. After server clustering process, each image source will belongs to one group or more

5) Search Process

In the search process, the user can then select query image and submit it to the *directory server* which acts as a gateway between *user* and different *image sources.* Once the *directory server* receives the query, the feature agent will extract the color and shape features from the query then it will pass the extracted features to the routing agent which will select the image sources group that contain images related to the query. After that the routing agent will dispatch a *search agent* to the set of *image sources* starting from the servers that contains images most similar to the query. On each *image source*, the *search agent* will match the extracted features if the query image against the stored features in the images database. The search agent will pass the results that containing the URL, as well as, the thumbnail of the similar images to the ranker agent which is responsible for filtering the result to remove the irrelevant images and ranking the results according to their similarity to the submitted image, finally the result are transmitted to the user. The next section explains steps of the search process in details.

a) Query Feature Extraction

When a user initiates a query for requesting similar images to a given an example image, the feature agent on the directory server will extract the feature vector from the example image and pass the feature vectors of query to the routing agent which is responsible for image source selection method. The Feature Extraction method was explained in Section 3.2.

b) Image source Selection

The main problem with Distributed Content based image retrieval is the image source selection and how to optimize the resource utilization by selecting image sources that are most likely to contain the relevant images for which a user is looking. The goal of image source selection method is to select the group of image sources that contain image categories relevant to the query. The square error is used to obtain the distance between a Query Image feature vector and the centroids vector of the server group:

$$Sim(\vec{C},\vec{Q}) = \min \sum_{i=1}^{k} \left\|\vec{Q}_i - \vec{C}_i\right\|^2 \quad (5)$$

Where the *C* and *Q are* centroid vector and query feature vector respectively.

c) Image sources Ranking

Image sources in the selected group are ranked based on the similarity between their centroids of image categories and the query feature vector. The euclidean distance between the query and the centroid of each image category in the servers group is calculated, and the servers are sorted in ascending order to determine which server will be searched first.

d) Similarity Measure

The similarity of the images is integrated by means of two components: the color similarity denoted *CS*, and shape similarity, denoted *SS*. The similarity measure between two images is evaluated by using the euclidean distance between the two images feature vectors.

$$\text{Recall} = \frac{\text{Relevant images retrieved}}{\text{Total number of relevant images}}$$

(1) Similarity Measure of Color

The similarity of two images according to the color is based on the similarity of their color histograms. The similarity of the *extracted histograms* between Query image *Q*, and an image *M* is defined as:

$$d_x = \sqrt{\sum_{i=o}^{255} (H_{X,i}^{Q} - H_{X,i}^{M})^2} \quad (6)$$

$$\text{Precision} = \frac{\text{Relevant images retrieved}}{\text{Total number of retrieved images}}$$

Where H_x is the normalized color histogram for a specific color (Red, Green and Blue).
The similarity measure between Query *image Q, and an image M* will be determined using the following formula for all colors:

$$CS(Q,D) = \sum dx(Q,M,H_X) \quad (7)$$

(2) Similarity Measure of Shape

The Euclidean distance was used to measure the similarity between Query shape feature vector and an image which is defined as:

$$SS(Q,M) = \sqrt{\sum_{i=0}^{n-1} (q_i - m_i)^2} \quad (8)$$

Where *Q* and *M* are the query and image M from images database, *qi* and *mi* are their *ith* features, respectively, and *n* is the dimension of the feature space which is equal in all images according to the re-sampling step in the shape feature extraction method.

e) Result Merging

In the retrieval process, the routing agent dispatches several mobile agents to search for similar images, each one functioning independently in computing the similarity between the query image and each candidate image based on its specialized similarity criterion and matching algorithm (color or shape features).

f) Result Ranking

To combine the results from different search agents, the ranker agent performs a simpler technique to combine multiple search agents' result together, and ranking them.
The final similarity measure is linear combination of the similarity measure of the two features (color and shape) with different weights.

$$Sfinal = w1 * Sshape + w2 * Scolor \quad (9)$$

Where f_i is the similarity grade for a specific image using similarity measure S_i with weight w_i. The $\Sigma w_i=1$.

III. System Performance

The retrieval performance is measured using recall and precision, as standard in all CBIR systems [17]. Recall measures the ability of retrieving all relevant or similar items in the database. It is defined as the ratio between the number of relevant or perceptually similar items retrieved and the total relevant items in the database.

Precision measures the retrieval accuracy and is defined as the ratio between the number of relevant or perceptually similar images and the total number of images that retrieved.

A. *Shape Feature Extraction Method*

For image retrieval on the basis of shapes, the tests were conducted on the Corel image database as a general database that used in most content based image retrieval systems to test the quality of the results. It is noticed that the shape measure is very effective in retrieving rotated images and scaled images. Figure 5 shows screenshots of the results by using shape feature extraction method.

B. *Color Feature Extraction Method*

Figure 6 shows screenshots of the results by using the color extraction method.

C. *Integration of Color and Shape Feature Extraction Methods*

The shape-based and color-based methods return two independent lists of images with different weights. These two lists should be combined in a meaningful way to give the user a combined image list. During the combining process, the user preference weighting for shape and color should be incorporated. The default weighting is 50% each for both methods. But the user can choose other weighting. Figure 7 shows a set of the result that was retrieved using the shape and color feature extraction methods.

IV. Performance Measure of Feature Methods

In this section, the images database was queried with a set of query images. The returned ranking images are checked to determine the relevant images. Based on the number of relevant images, the precision and recall rates are measured. Figure 8 and 9 present the approximately recall and precision respectively of the results.

V. Conclusion

Due to scalability issue of centralized databases and their failure to handle the growing retrieval requests and a larger image database, the Distributed CBIR systems were presented but doing this in a distributed environment is a great challenge. This paper presents an approach for distributed CBIR using mobile agent. The proposed approach is based on a multi-agent paradigm run on several image servers. The agents autonomously search for images over the Internet, and then convert the images to a vector used for searching and retrieving images according to its contents. This paper shows the quality of results according to the feature extraction method. The results show that retrieval with color only retrieves more irrelevant images and demonstrates while the retrieval by color with shape methods outperforms the result with color feature method. It is concluded that the color only is not enough to represent image feature and to get better result, so the system used integration between color and shape features to represent images. Future work will concern on Image retrieval based on the regional features which has attracted a lot of interest in the last few years.

References

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age," ACM Computing Surveys, vol. 40, no. 2, pp. 1-60, 2008.
[2] King, C. H. Ng, and K. C. Sia, "Distributed Content-Based Visual Information Retrieval System on Peer-To-Peer Networks," ACM TOIS, vol. 22, pp. 477-501, 2004
[3] H. Q. Nguyen, and Q. T. Ngo, "A Novel Method for Content Based Image Retrieval Using Color Features," IJCSES International Journal of Computer Sciences and Engineering Systems, vol. 3, no. 1, 2009.
[4] H. B. Kekre and S.D. Thepade,"Image Retrieval using Augmented Block Truncation Coding

Techniques”, International Conference on Advances in Computing, Communication and Control (ICAC3’09), India. ACM 2009

[5] S. Deb, and Y. Zhang, “An Overview of Content-Based Image Retrieval Techniques,” 18th International Conference on Advanced Information Networking and Applications, AINA, IEEE CS, 2004.

[6] A. M. Riad, A. Atwan, and S. A. El-Ghany, “Analysis of Performance of Mobile Agents in Distributed Content Based Image Retrieval,” International Conference on Computer Engineering & Systems (ICCES), IEEE, Cairo, Egypt, pp. 349-354, 2008.

[7] H. M. El-bakry, A. M. Riad, A. Atwan, S. A. El-Ghany, and N. Mastorakis “A New Automated Information Retrieval System by using Intelligent Mobile Agent, " Proc. of Recent Advances in Artificial Intelligence, Koweledge Engineering and Databases, Cambridge, UK, February 20-22, 2010, pp. 339-351.

[8] K. Jarrah, and L. Guan, “Content-Based Image Retrieval via Distributed Databases,” Proceedings of the international conference on Content-based image and video retrieval, Canada. pp. 389-394, 2008

[9] Roth, U. Pinsdorf, and J. Peters, "A Distributed Content Based Search Engine Based on Mobile Code," In 20th ACM Symposium on Applied Computing, Special Track on Agents, Interactions, Mobility, and Systems (SAC/AIMS), USA, pp. 66-73, 2005

[10] Picard, M. Cord, and A. Revel,”Image Retrieval Over Networks: Active Learning Using Ant Algorithm” IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 10, NO. 7, NOVEMBER 2008

[11] D. Picard, A. Revel , and M. Cord, “Long term learning for image retrieval over networks”. *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on* , vol., no., pp.929-932, 12-15 Oct. 2008

[12] L. Wen1, and G. Tan, “Image Retrieval using Spatial Multi-Color Coherence Vectors Mixing Location Information,” ISECS International Colloquium on Computing, Communication, Control, and Management, IEEE, vol. 1, pp. 299-302, 2008.

[13] T. Deselaers, D. Keysers, and H. Ney, “Features for image retrieval: an experimental comparison,” Information Retrieval, vol. 11, no. 2, p. 77-107, 2008.

[14] H. Lin, Y. Kao, S. Yen, and C. Wang, “A Study of shape-Based Image Retrieval,” Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW’04), IEEE, vol. 7, pp. 118- 123, 2004.

[15] T. V. Gopal, and V. K. Prasad, “A Novel Approach to Shape Based Image Retrieval Integrating Adapted Fourier Descriptors and Freeman Code,” IJCSNS International Journal of Computer Science and Network Security, vol. 8, no. 6, 2008.

[16] W. Li, “Modified K-means Clustering Algorithm,” Congress on Image and Signal Processing, CISP, IEEE, vol. 4, pp. 618-621, 2008.

[17] Katare, S. K. Mitra, and A. Banerjee “Content Based Image Retrieval System for Multi Object Images using Combined Features” Proceedings of the International Conference on Computing: Theory and Application, Kolkata, India, Pages 595-599 IEEE 2007

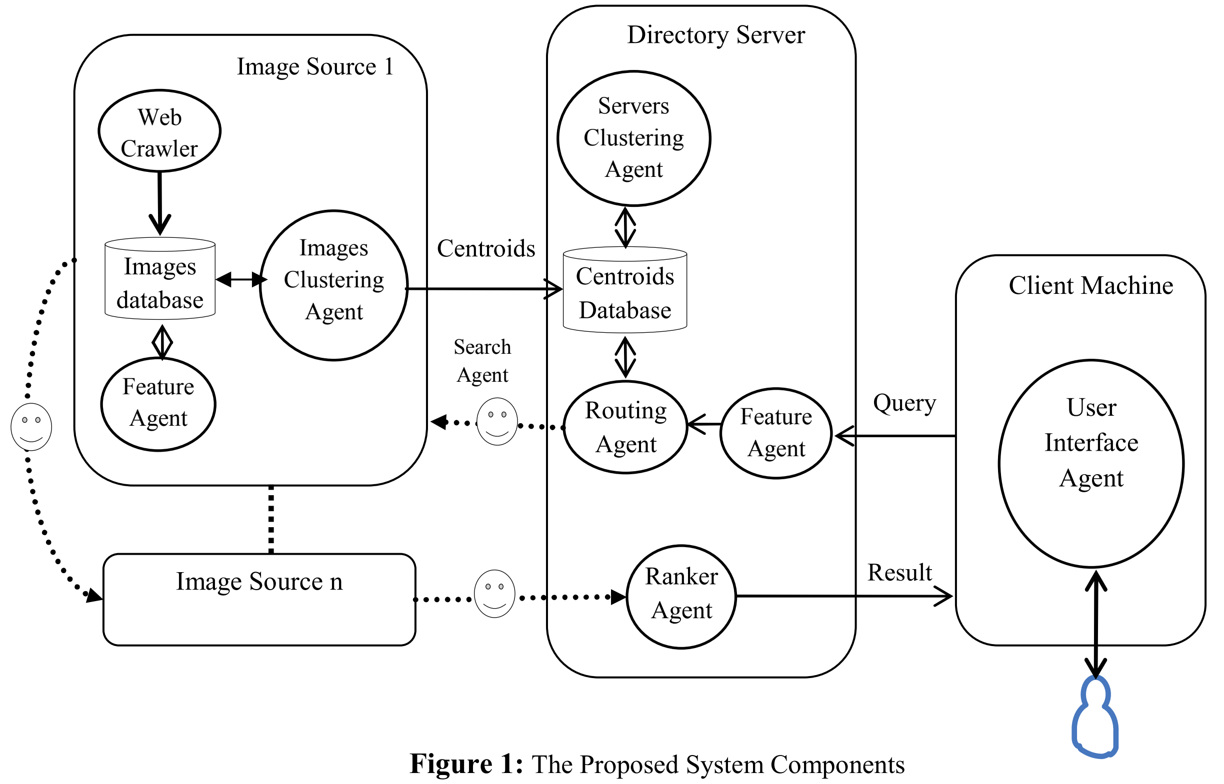

Figure 1: The Proposed System Components

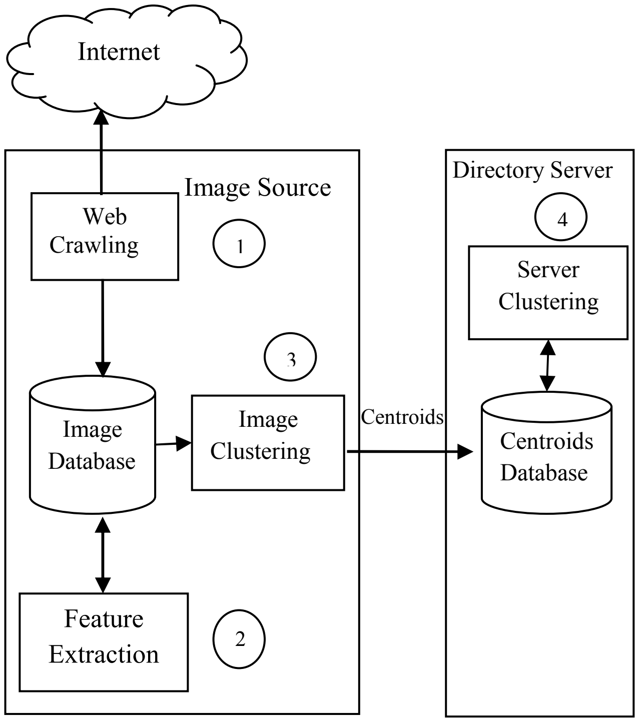

Figure 2: The Preprocessing Phase

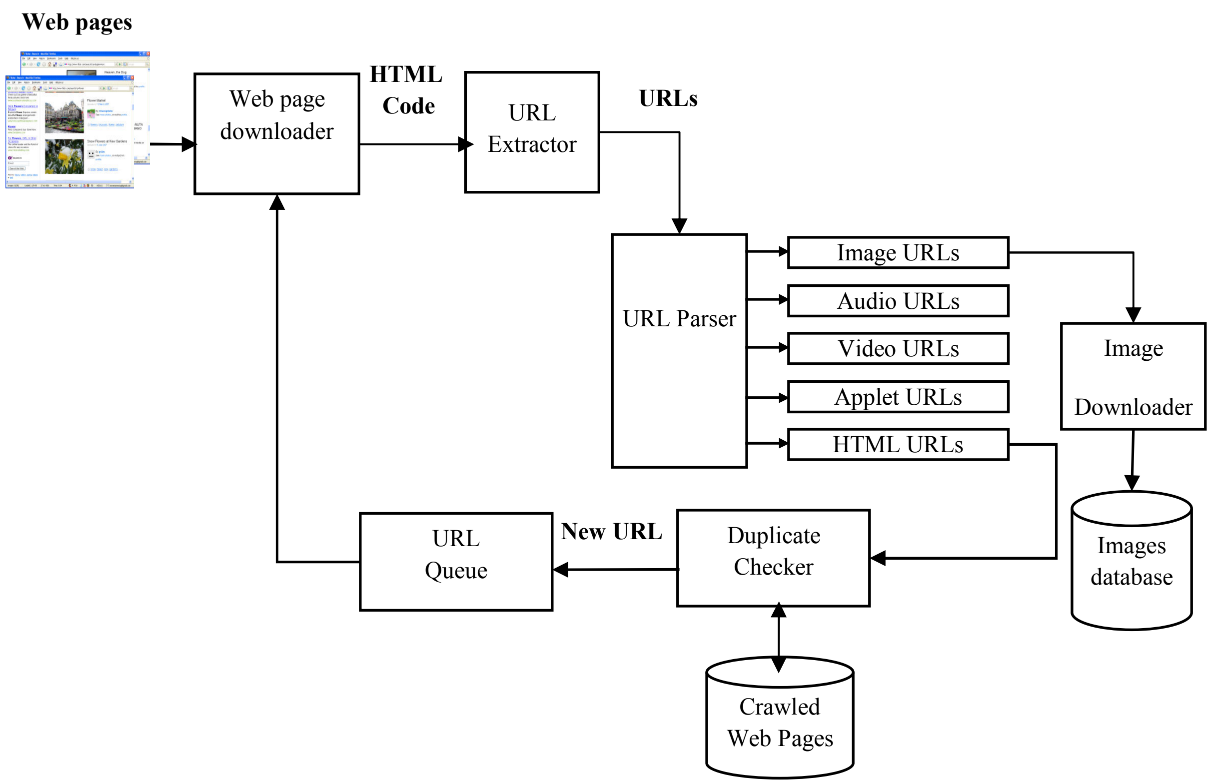

Figure 3: Web Crawling Process

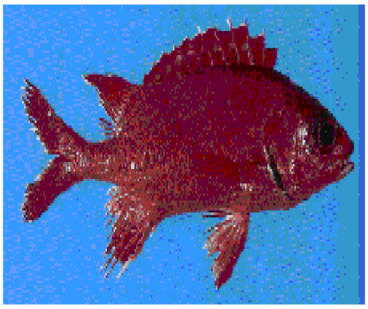

Figure 4a: Original Image

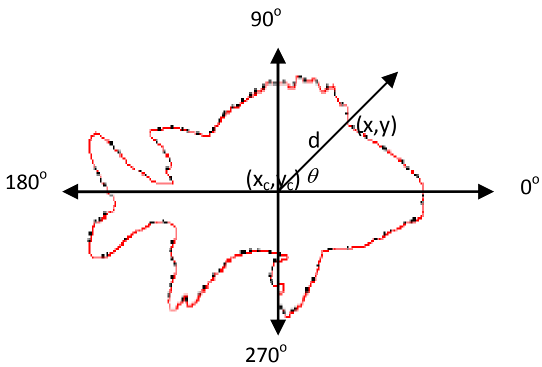

Figure 4b: Distance and Angle of the Contour Points Relative to Centroid

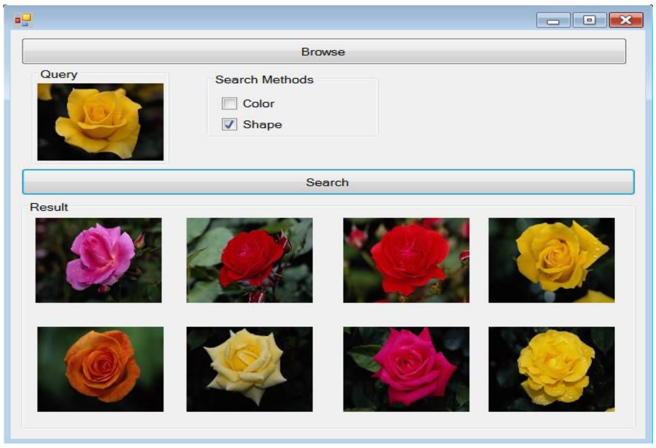

Figure 5: A Screen Shot For the Retrieved Result Using the Shape Feature Extraction Method

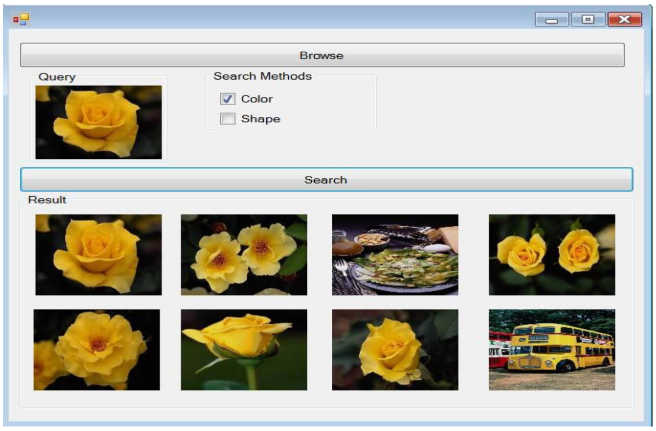

Figure 6: A Screen Shot For the Retrieved Results Using the Color Feature Extraction Method

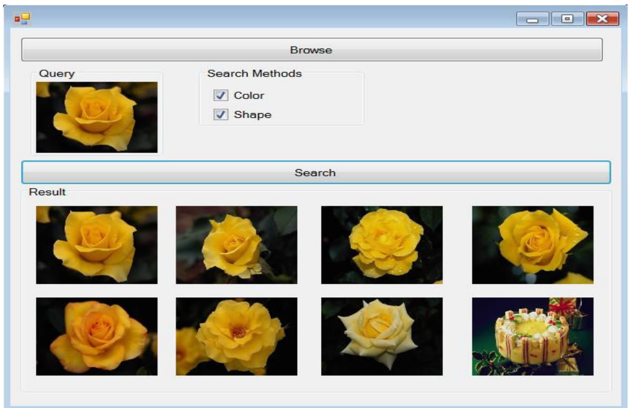

Figure 7: A Screen Shot For a Set of the Retrieved Result Using the Color and Shape Feature Extraction Method

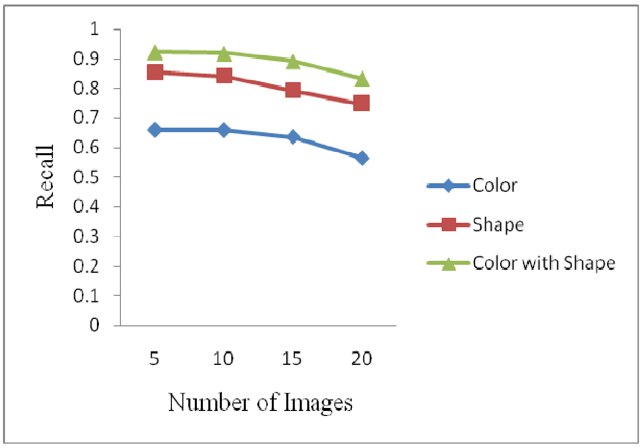

Figure 8: The Recall against the Number of Images with Color and Shape Feature Extraction Methods.

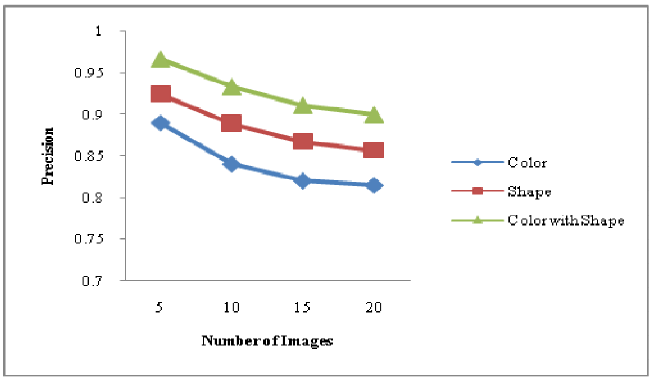

Figure 9: The Precision vs. the Number of Returned Images with Color and Shape Feature Extraction Methods.