RETRIEVAL OF TRADEMARK IMAGES BY SHAPE FEATURE

A PROJECT REPORT

Submitted by

DEEPIKA.L 71205104003

NIVEDITHA.K 71205104026

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY: CHENNAI 600025

APRIL 2009

BONAFIDE CERTIFICATE

Certified that this project report **"RETRIEVAL OF TRADEMARK IMAGES BY SHAPE FEATURE"** is the bonafide work of **DEEPIKA.L** and **NIVEDITHA.K** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.Thangasamy,	Mrs.D.Chandrakala,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	Assistant Professor
Dept of CSE,	Dept of CSE,
Kumaraguru College of Technology,	Kumaraguru College of Technology,
Coimbatore-641006	Coimbatore-641006

The candidates with University Register Nos. **71205104003, 71205104026** were examined by us in the project viva-voce examination held on 27.4.2009

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled **"Retrieval of Trademark Images using Shape Feature"** is a record of original work done by us and to the best of our knowledge, a similar work has not been submitted to Anna University or any Institutions, for fulfillment of the requirement of the course study.

The report is submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer Science and Engineering of Anna University, Chennai.

Place : Coimbatore

Date : 27-04-2009

L. Deepika.

(Deepika.L)

K. Niveditha.

(Niveditha.K)

ACKNOWLEDGEMENT

We are extremely grateful to **Prof. Annamalai, M.E,** Vice Principal, Kumaraguru College of Technology for having given us this opportunity to embark on this project.

We are deeply obliged to **Dr. S. Thangasamy, Ph.D.,** Head of the Department of Computer Science and Engineering for his valuable guidance and useful suggestions during the course of this project.

We wish to express our heartiest thanks to **Mrs.P.Devaki**, Assistant Professor, who helped us to overcome the perplexity while choosing the project.

We thank our guide **Mrs.D.Chandrakala**, Assistant professor, Department of Computer Science and Engineering, for her excellent guidance in each and every step of our project and been with us to complete the project.

We thank the **teaching and non-teaching staff** of our department for providing us the technical support during the duration of our project.

We also thank all our **parents and friends** who helped us to complete this project successfully.

ABSTRACT

Trademarks are an important part of a company's industrial property, and it is a crucial responsibility of the Trade Marks Registry to ensure that all new trademarks registered are sufficiently distinctive to avoid confusion with existing marks. Since its inception, the Trade Marks Registry has classified trademark images using an elaborate system of manually-assigned codes. These codes form the basis of their TRIMS (Trademark Image System) image retrieval system, which allows users to specify Boolean combinations of category codes and display all images meeting the search criteria. Many trademark images are intended to depict animate or inanimate objects, such as cats, stars, or flowers, and TRIMS works well in such cases. However, a sizeable fraction (now numbering well over 20 000) are made up of abstract geometric designs with little or no representational meaning. Current indexing practice is to code for the presence of recognizable geometric shapes such as circles, triangles, or squares. This provides a partial solution to the problem, but since there are now several thousand images in each category, Registry staff attempting to establish the novelty of a trademark based on an abstract design are faced with an almost unmanageable task. There is thus a need for a system which provides reliable (and if possible automatic) indexing and retrieval for this class of images.

Our system retrieves abstract trademark images by shape similarity. It extracts the global features and local features of the query image and based on the similarities among those features with the images in the database, the top n similar images are retrieved. The classification of the images is done using neural network.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

BP	Back propagation
RBF	Radial Basis Function
ANN	Artificial Neural Network
NN	Neural Network
LR	Learning Rate

1. INTRODUCTION

Trademark is a symbol in a form of an image used to publicize and to indicate the services or products of an organization or a company. Trademark symbols enable clients to identify good products. The trademark symbol is legally registered representing the specific company or the organization. A registered trademark is protected through legal proceedings from imitation and misuse. Based on these aspects, it is a stringent requirement for trademark symbols to be uniquely different from other trademarks for legal reasons and in order not to mistakenly identify the company's identity.

Trademark offices in several countries in the world strive to ensure the uniqueness of all registered trademarks. There is a very challenging task due to ever increasing number of trademarks. Up to now, the number of trademarks worldwide is over one million and is growing rapidly. The trademark registry offices worldwide classify trademark images into four categories as shown in the Figure 1 to ease a classification task. Word-in-mark consists of either a word or a character as in Figure 1.a. Device-mark consists of simple graphics or images as in figure 1.b.Composite-mark consists of word or character embedded in simple graphics or images as in Figure 1.c. Complex-mark consists of complex images without outstanding representation or shape as in Figure 1.d.

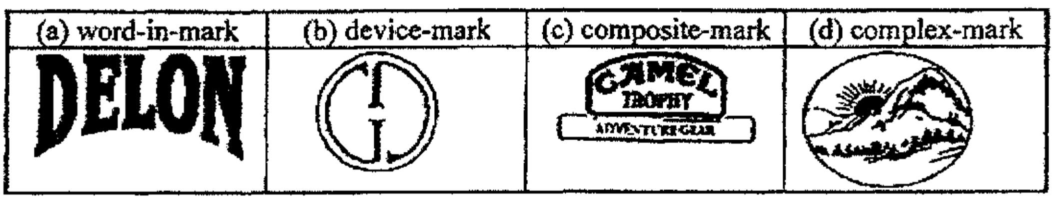


Figure 1 Example of trademark images

1.1 EXISTING SYSTEM

Traditional methods for trademark image retrieval like Vienna classi?cation developed by the World Intellectual Property Organization (Figure 2) use manually assigned classification codes to reflect image content. In principle, this solves the problem of image registration by ensuring that similar images will receive identical classi?cation codes.

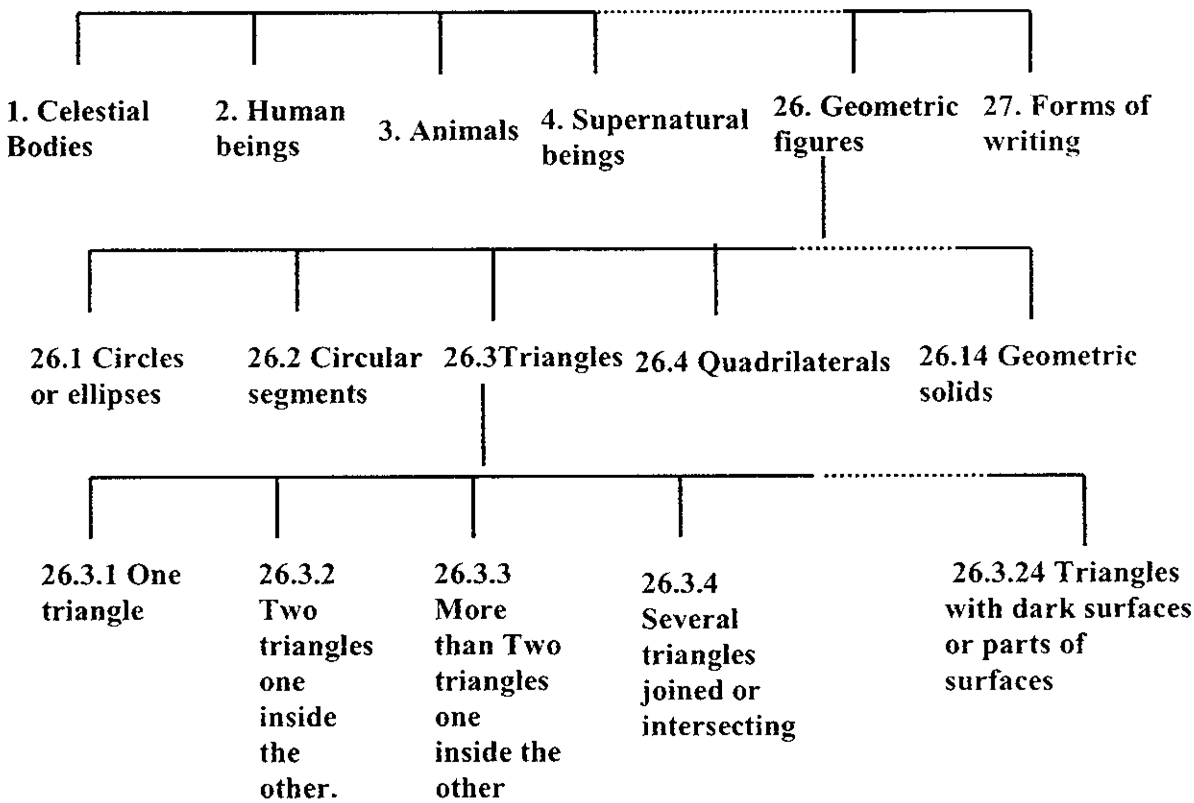


Figure 1.1 Trademark image classifications according to the existing system

However, it still suffers from two major drawbacks, both inherent in any retrieval system based on manual classification codes:

- The matching task is feasible provided the existing trademark is few in numbers, however when the numbers escalated to hundred thousands, the matching task becomes tedious and time consuming. Manual classification of images is potentially error-prone.
- Classification codes are not always helpful for retrieval, particularly for abstract images. Similarity judgments may be based on a number of criteria, including overall shape, the shapes of image components, and the spatial configuration of components. No current classification scheme can reflect the full range of such criteria.
- In addition, other problems arise resulting from using this method.Firstly, the decision on which class a trademark should be subjective, secondly the class becomes either too specific or too broad depending on how the users use the classes and finally there is no mechanism to handle the generation of new classes.
- In addition there is a large fraction of images with little or no representational meaning, making such a classification task extremely difficult.

1.2 PROPOSED SYSTEM:

To overcome this problem, and to make inroads into the field of Trademark image retrieval, many methods have been proposed that take into account certain features of images when making a comparison between any two of them. In the proposed method, one of the most fundamental attributes of an image, its shape, is considered for processing the images. Since shape is nothing but the outline that an object presents to the external world, two objects that present identical outlines could well be taken to belong to the same category. Here the external feature is the outline shape of the trademark image and inner features are those different shapes that are present within the outer boundary. These features are taken to be the defining attribute that is compared with that of other images for the retrieval of relevant results.

1.3 ABOUT THE IMPLEMENTATION

This project addresses a shape description approach for avoiding the retrieval of irrelevant results in trademark image retrieval and to increase the overall efficiency of the trademark image retrieval process. The proposed model is implemented and evaluated using MatLab Version 7.5 under Microsoft Windows Operating System. The images and the information used are collected from the Internet and other electronic and bibliographical information sources.

2. METHODOLOGY

2.1 PROPOSED SYSTEM – OVERALL DESIGN:

The proposed approach for image retrieval takes as input a query image from the user. The query image is first pre-processed and then the convex hull property is used in order to get the outer boundary of the region of interest. The inner features are also extracted from the query image. These features are represented in a form that is invariant to translation, scaling and rotation and this is done by calculating Hu's invariant moments. The outer edge of the region of interest is given as input to the neural network which is already trained using the edge values in the database. Using neural network, the subset of the database is obtained. Then each feature moment values for the corresponding query image is compared with the feature moment values of the images in the database and based on the threshold the top N images that are more similar are retrieved.

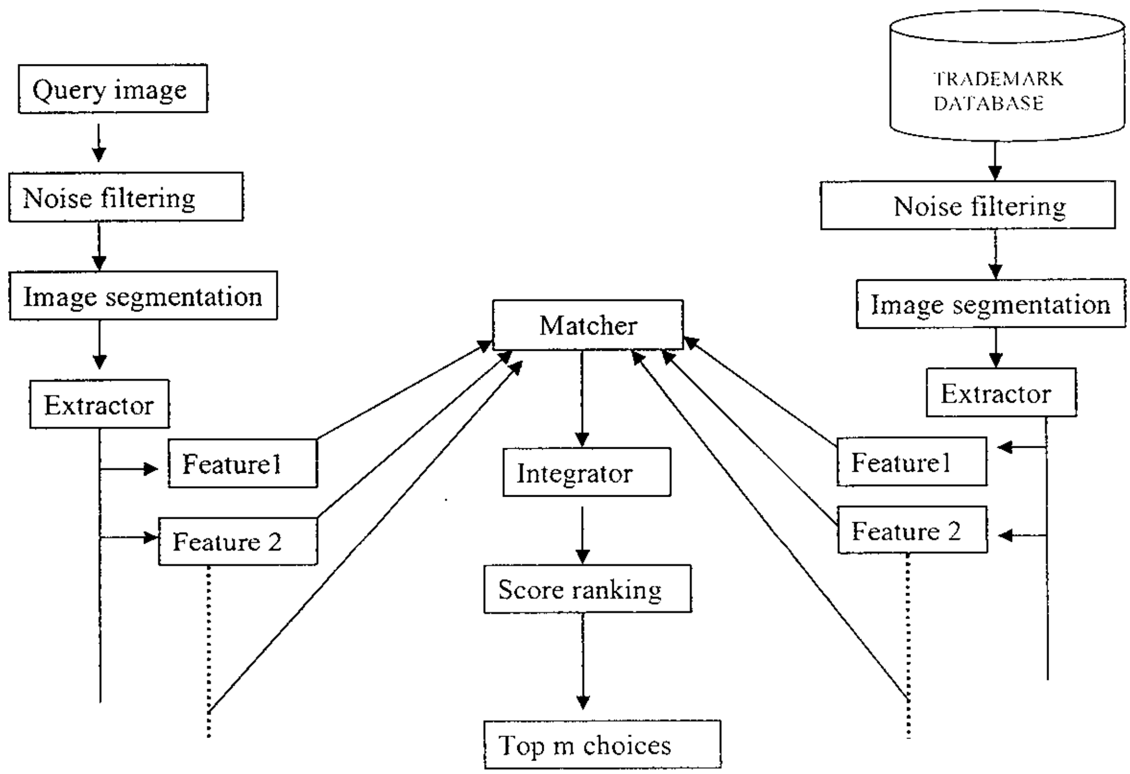


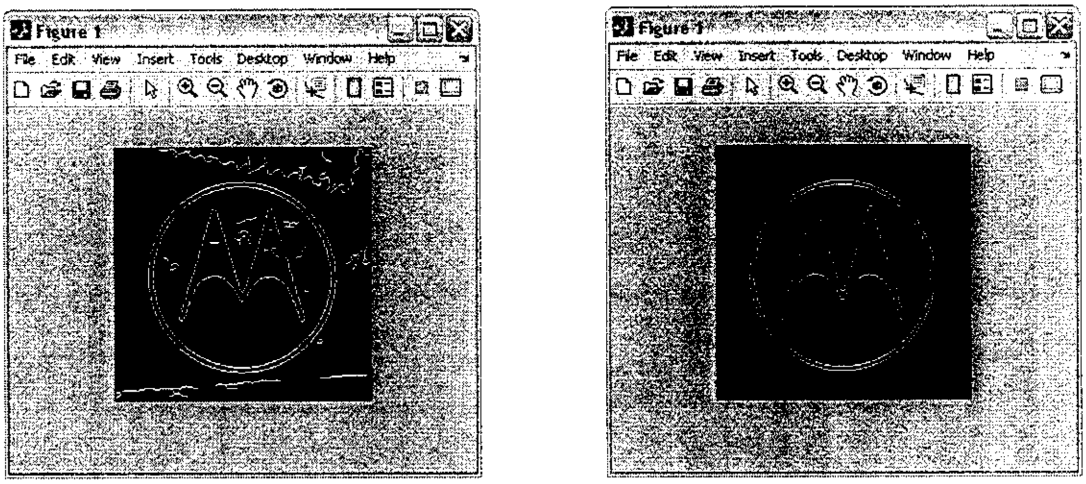
Figure 2.1 Proposed system architecture

2.2 IMAGE DATABASE MANAGEMENT:

The database has mainly UK registered trademark images. The United Kingdom Patent Office now holds more than 300,000 current trademarks in its registry, of which 40 percent contain some form of image data. The types of trademark images that are used as a test case is limited to the category of device-mark with a distinct geometric shape particularly circles and triangles. Rotated, translated and scaled images of each image are also stored along with the original image.

2.3. PREPROCESSING:

This pre-processing process includes the following. Image resizing and noise filtering. The trademark images that are considered are of different sizes. For further processing, all the images are to be of similar size and hence they are first resized to a standard size of 200 by 200 pixels. Spatial filters can be effectively used to remove various types of noise in digital images. There are three main categories of spatial filters namely; order filters, median filters and adaptive filters. The order filters are implemented by arranging the neighborhood pixel in order from the smallest to largest gray-level value and using this ordering to select the correct value.. The order filters are a non linear, thus the results are sometimes unpredictable. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high-frequency parts of an image. Median filtering is similar to using an averaging filter, in that each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called outliers). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image.



(a) (b)

Figure 2.3. (a) image before noise filtering (b) image after noise filtering

2.4 IMAGE SEGMENTATION:

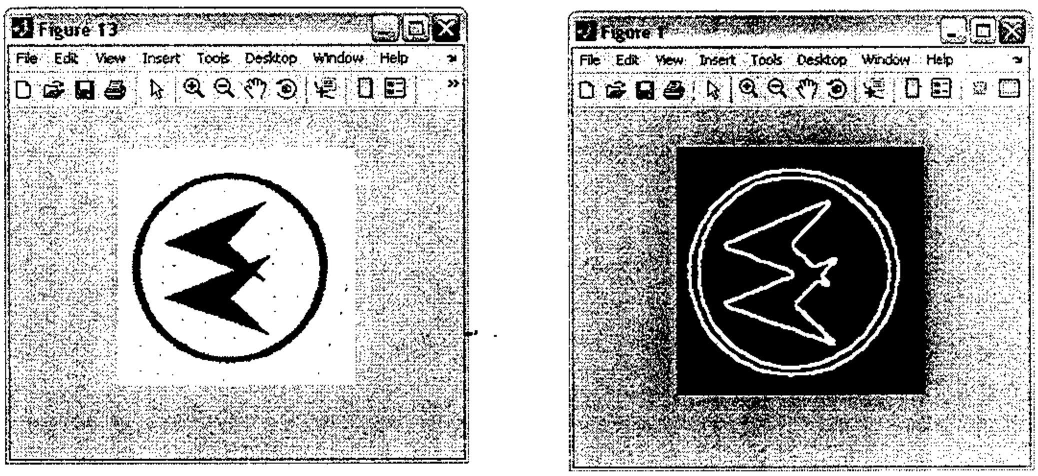
A central problem, called segmentation, is to distinguish objects from background. For intensity images (i.e., those represented by point-wise intensity levels) four popular approaches are: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods.

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc.

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Hybrid techniques using a mix of the methods above are also popular.

A connectivity-preserving relaxation-based segmentation method, usually referred to as the active contour model, was proposed recently. The main idea is to start with some initial boundary shapes represented in the form of spline curves, and iteratively modify it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an ``elastic" contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task.

Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. The edge detection technique using sobel operator is well suited and therefore used in our implementation.



(a) (b)

Figure 2.4. (a) image before segmentation (b) image after segmentation

2.5 FEATURE EXTRACTION:

After the segmentation process, each inner feature within a trademark image is to be extracted. This is done by extracting all the connected components. Each connected component represents a feature.

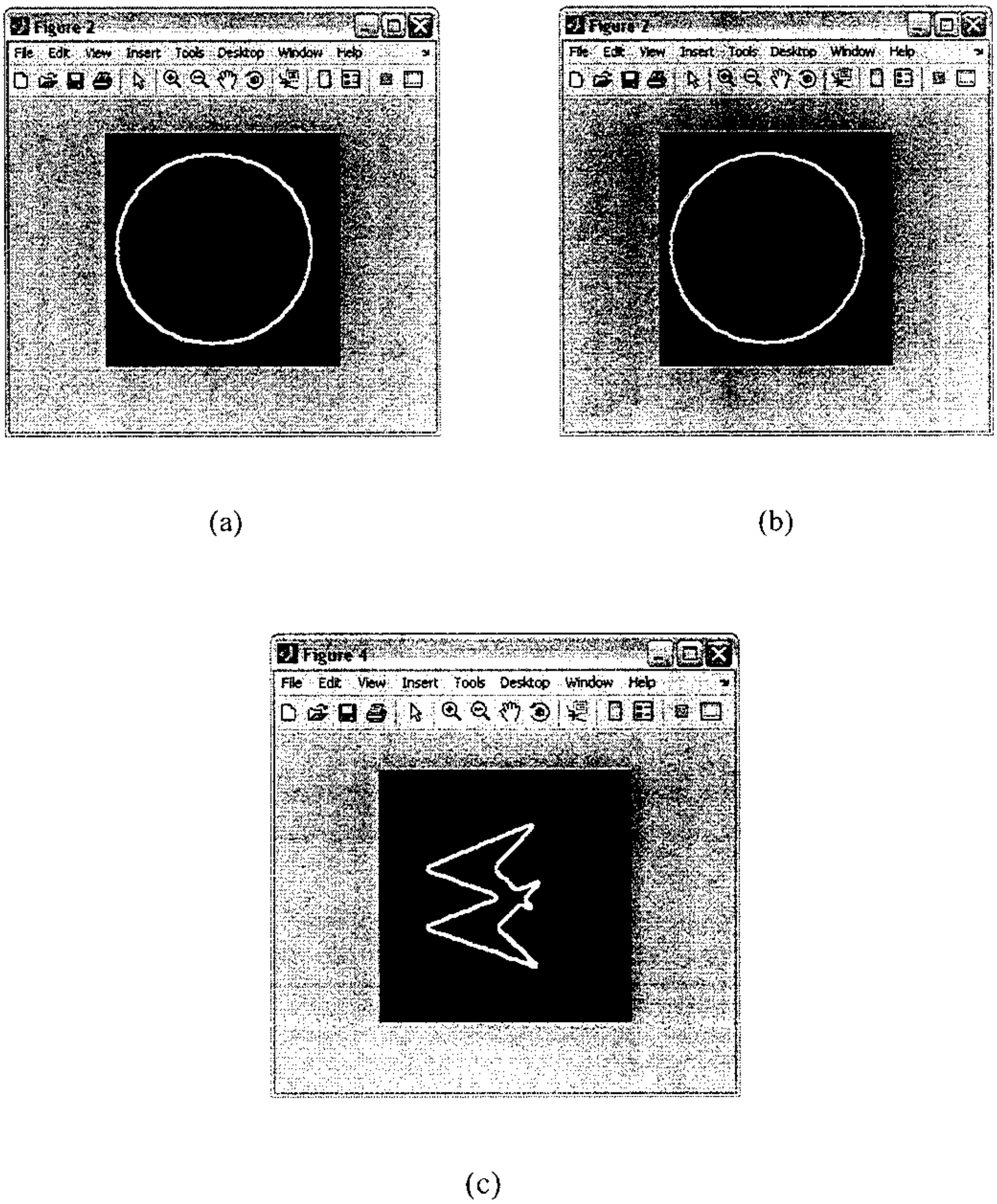
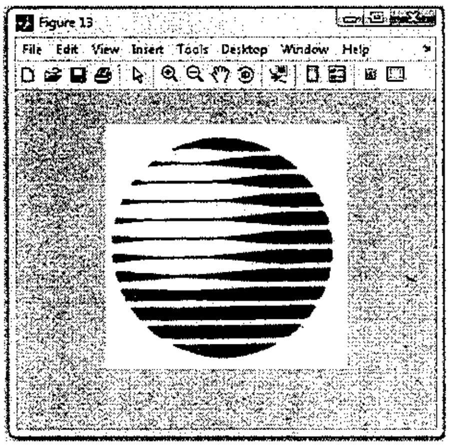


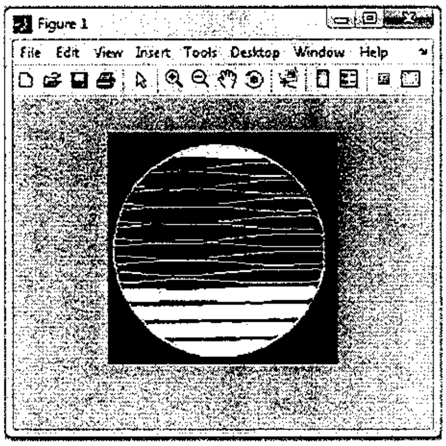
Figure 2.5.1 (a) feature1 (b) feature2 (c) feature 3 of the figure 2.4(a)

Generally extracting the outer boundary for the images with discontinuities is a difficult task. Hence in order to find out the outer boundary even for such images, we use the convex hull methodology. Hence by using this technique, even if there occurs discontinuities in the outer boundary or there is no discontinuity, the outer edge is detected and extracted.

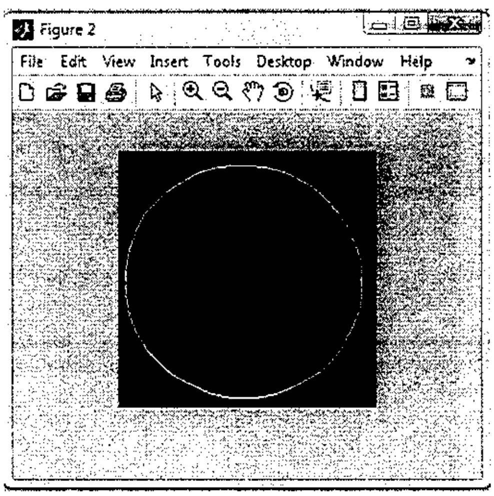
Following is an example where an image that has many discontinuities in its outer boundary is detected and the edge is extracted.



(a)



(b)



(c)

Figure 2.5.2 (a) original image with discontinuity (b) finding the complete edge (c) extracted edge

2.5.1 MOMENT CALCULATION:

Moment based feature extraction is followed because of the following reasons

- Moments represent the global characteristics of the image shape.
- It also provides a lot of information regarding the different types of geometrical features inherent in the image.
- It produces a set of features that are invariant
- The properties of the image moments resemble the phenomenon in statistics and mechanics that had been applied to solve real problems successfully.

There are many moment based feature extraction algorithm and the method that is considered in our implementation is geometric invariant moments. The main advantage of this is that moment values are invariant to translation, rotation and scaling.

The algorithm for the geometric invariant moments is as follows

a) generate the regular moments for the binary image using the following equation

$$m_{pq} = \sum_{i=1}^{L} x_i^p y_i^q$$

Where L : the number of pixels belonging to the object

p, q: the moment order(p=0,1,2,3 q=0,1,2,3)

b) Compute the image centroid using the following expression

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}}$$

c) Produce the central moments invariant to translation or displacement using the following expression,

$$\mu_{pq} = \sum_{x}\sum_{y}(x-\bar{x})^p (y-\bar{y})^q$$

d) Compute the moments invariant to scale or size based on the following equation,

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}$$

Where

$$\gamma = \frac{(p+q)}{2} + 1$$

e) Compute a set of 7 moment functions defined on $?_{pq}$ that are invariant to rotation, translation and scale changes based on the following equations

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{30})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{30})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2]$$

The above seven numerical values of geometric invariant moments or features are very small. To avoid precision problems the logarithms of the absolute values of the seven functions are selected as features representing the image.

2.6 CLASSIFICATION BASED ON NEURAL NETWORK:

2.6.1 INTRODUCTION

An artificial neural network consists of a large number of processing elements called neurons. Each neuron is connected to other neurons by a weighted links. The ANN can also be viewed as weighted directed graphs in which nodes represent neurons and weighted directed edges are connections from an input layer to output layer or vice versa. Weights are the primarily means of long term storage in neural network. Neurons in the input and output layers are normally connected to the external environment. Each neuron has a set of input links from other neurons, a set of output links to other neurons, a current activation level and an activation function. The activation function is a formula used to compute the neuron's activation level given its inputs and weights. There are different types of activation functions like step function, sign function, linear function and sigmoid function.

A neural network model is a structure that can be adjusted to produce a mapping from a given set of data to features of or relationships among the data. The model is adjusted, or trained, using a collection of data from a given source as input, typically referred to as the training set. After successful training, the neural network will be able to perform classification, estimation, prediction, or simulation on new data from the same or similar sources. The Neural Networks supports different types of training or learning algorithms.

There are different types of neural networks such as feed forward, radial basis function, SVM etc. A neural network model can be customized when the unknown function is known to have a special structure.

For example, in many situations the unknown function is recognized as more nonlinear in some inputs than in others.

The feed forward back propagation and radial basis function network are the two models that are taken into our study.

2.6.2 BACK PROPAGATION AND RADIAL BASIS FUNCTION NETWORK:

The network consists of input layer, one or more hidden layer and an output layer. The layers are connected in the form of a feed forward architecture. In classification or recognition problems, the input layer is defined by an array of nodes that constitute a sampled version of the input signal .the training input data is in the form of vectors of input variables or training patterns. Each element in an input vector corresponds to an input node in the input layer. Nodes in the hidden layer detect the pattern in the input layer and perform non-linear mapping between input and output. The output layer is defined by a set of nodes each corresponding to a class, pattern or category.

2.6.2.1 BP ALGORITHM:

The training of a neural network by BP involves three stages : the feed forward of the input training pattern,the calculation and backpropagation of the associated error and adjustment of the connection weights. After training, the generalization capability of the network is tested based only on the computation of the feed forward phase.

During feed forward computation, each input neuron receives an input signal and broadcasts the signal to each of the hidden neurons. Each hidden unit then computes its activation and sends its signal to each output neuron. Each output neuron computes its activation to form the response of the net for the given input pattern.

ARCHITECTURE OF BP NEURAL NETWORK

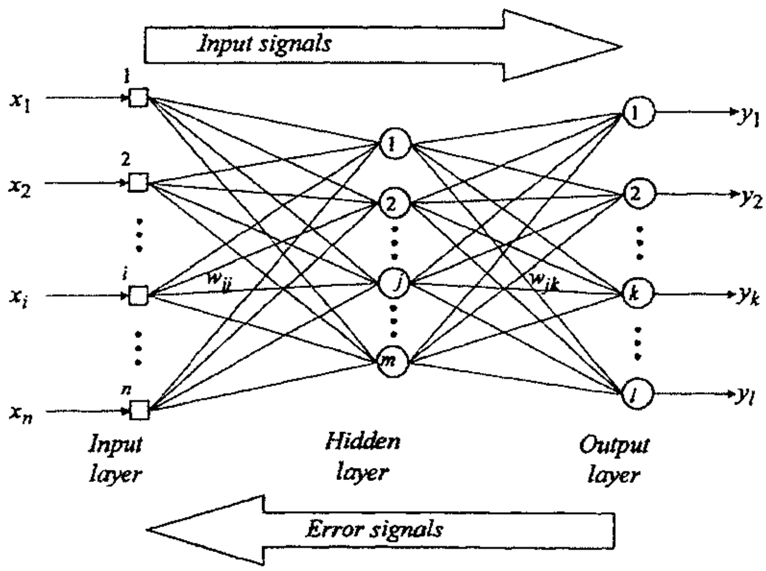


Figure 2.6.2.1. Architecture of BP network

TRAINING ALGORITHM OF BP NEURAL NETWORK:

Step 0. Initialize weights.(Set to small random values)

Step 1. While stopping condition is false do steps 2-9.

Step 2. For each training pair, do steps 3-8.

Feed forward:

Step 3. Each input neuron(X_i i=1,...n) receives input signal x_i and broadcasts this signal to all neurons in the hidden layer.

Step 4. Each hidden neuron (Z_j, j=1,...,p) sums its weighted input signals,

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$$

applies its activation function to compute its output signal. $z_j = f(z\_in_j)$, and sends this signal to all neurons in the output layer

Step 5. Each output neuron (Y_k, k=1,...,m) sums its weighted input signals

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$$

And applies its activation function to compute its output signal,

$y_k = f(y\_in_k)$

Back propagation of error:

Step 6. Each output neuron (Y_k, k=1,...,m) receives a target pattern corresponding to the input training pattern, computes its error information term,

$$d_k = (t_k - y_k)f(y\_in_k)$$

Calculation its weight correction term (used to update w_{jk} later),

$$w_{jk} = a d_j z_j$$

Calculates its bias correction term(used to update w_{0k} later),

$$w_{0k} = a d_k$$

and sends d_k to units in the hidden layer.

Step 7. Each hidden neuron (Z_j, j=1,...,p) sums its delta input (from the output layer above),

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$$

Multiplies by the derivative of its activation function to calculates its bias correction term (used to update v_{0j} later),

$$v_{0j} = ad_j,$$

Update weights and biases:

Step 8. Each output neuron (Y_k, k=1,...,m) updates its bias and weights (j=0,...,p);

$$w_{jk}\ (new) = w_{jk}\ (old) + \quad w_{jk}$$

Each hidden neuron (Z_j, j=1,...,p) updates its bias and weights (i=0,....,n);

$$v_{jk}\ (new) = v_{jk}\ (old) + \quad v_{jk}$$

Step 9. Test stopping condition

2.6.2.2 RADIAL BASIS FUNCTION:

The Radial Basis Function (RBF) networks performs similar function mapping with the multi-layer neural network, however its structure and function are much different. A RBF is a local network that is trained in a supervised manner. This contrast with a MLNN is a global network. The distinction between local and global is made through the extent of input surface covered by the function approximation. RBF performs a local mapping, meaning only inputs near a receptive field produces activation.

As is the case with most neural networks, the aim is to train the net to achieve the balance between the net's ability to respond and the ability to give reasonable responses to the input that is similar, but not to the one used in training.

A multi-layer net can learn only input patterns to an arbitrary accuracy. More than one hidden layer can also be used though one is sufficient. The weight in a neural network is a segment of the information about the input signal that has to be stored.

Architecture of Radial Basis Function Network (RBFN)

The input layer of this network is a set of n units, which accept the elements of an n -dimensional input feature vector. n elements of the input vector x_n is input to the l hidden function, the output of the hidden function, which is multiplied by the weighting factor w_{ij} , is input to the output layer of the network $y_j(x)$.

For each RBF unit k, k = 1, 2, 3, ... l, the center is selected as the mean value of the sample patterns belong to class k, i.e.,

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_k^i, k = 1,2,3,...,m$$

Where x_k^i is the eigenvector of the i^{th} image in the class k, and N_k is the total number of trained images in class k. For any class k, the Euclidean distance d_k from the mean value μ_k to the farthest sample pattern x_k^f belong to class k:

$$d_k^f = \left\| x_k^f - \mu_k \right\|, \quad k=1, 2, \ldots, m$$

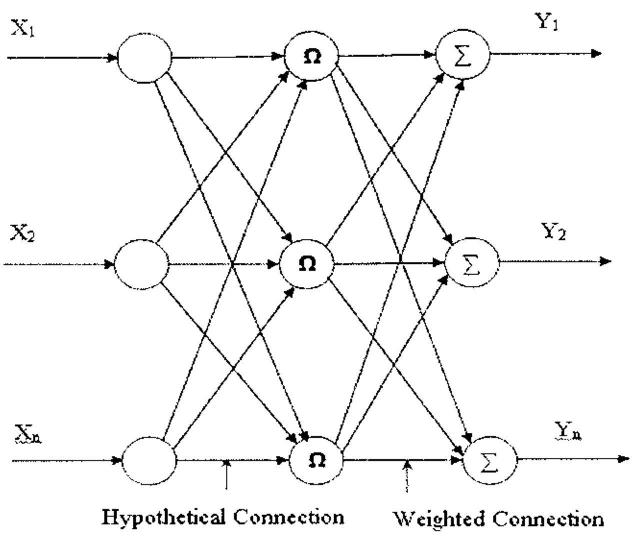


Figure 2.6.2.2 Architecture of RBF network

Only the neurons in the bounded distance of d_k in RBF are activated, and from them the optimized output is found. For designing a RBF neural network classifier, the number of the input data x_i is equal to the number of the feature vector elements.

Activation Function

The activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). Radial Basis Function uses Gaussian activation function. The response of such function is non – negative for all value of x. The function is defined as

$$f(x) = x$$

Training Algorithm for RBF Network

The training algorithm for the Radial Basis Function Network is given below. The important aspect of the Radial Basis Function Network is the usage of activation function for computing the output.

The training algorithm for the network is given a follows:

Step 1: Initialize the weights (set to small random values).

Step 2: While stopping condition is false do Steps 3 – 10.

Step 3: For each input do Steps 4 – 9.

Step 4: Each input unit (x_i, i=1,...n) receives input signals to all units in the layer above (hidden unit).

Step 5: Calculate the radial basis function.

Step 6: Choose the centers for the radial basis functions. The centers are chosen from the set of input vectors. A sufficient number of centers have to be selected in order to ensure adequate sampling of the input vector space.

Step 7: The output of i_m unit is given by $f_i(x_i)$ in the hidden layer.

$$f_i(x_i) = e\left(-\sum_{j=1}^{r}\left[x_{ji} - x_{ji}\right]^2 / \sigma_i^2\right)$$

Where

x_{ji} = Center of the RBF unit for input variables.

σ_i = Width of the ith RBF unit.

$x_{ji} = j_{th}$ variable of input pattern.

Step 8: Initialize the weights in the output layer of the network to some small random value.

Step 9: Calculate the output of the neural network

$$y_{net} = \sum_{i=1}^{H} w_{im} f_i(x_i) + w_0$$

where,

H – number of hidden layer nodes (RBF function)

y_{net} - output value of m_{th} node in output layer for the n_{th} incoming pattern.

w_{im} – Weight between i_{th} RBF unit and m_{th} output node.

w_o – Biasing term at n_{th} output node.

Step 10: Calculate error and test stopping condition.

The stopping condition may be the weight change, number of epochs, etc.,

2.7 SIMILARITY MEASUREMENT OF TRADEMARK IMAGES

The matching (similarity) score of the retrieved images is calculated after finding the percentage of match of all the features of the query image and all features of each images in the database. The subset of the trademark images is selected based on the value obtained from the neural network and similarity measurements are done only using the selected subset. For each image in the subset, the difference of the moment values of the ith feature and the corresponding moment values of the ith feature of the query image is calculated. From the obtained differences, percentage of similarity between the ith feature of query image and the images in the subset is calculated using the below formula

$$\frac{\text{Max - x}}{\text{Max - min}}$$

Where

x – the difference between the ith feature of the query image and the corresponding images in the subset.

max - maximum threshold limit, constant.

min – minimum threshold limit, constant.

For each image in the database subset, the average of the similarity percentage of all its features is calculated. The obtained average percentages for the images in the subset are sorted in descending order and stored.

3. IMAGE QUERYING AND RETRIEVAL

In the image querying phase, the user supplies a trademark image whose similar images are to be retrieved. The software finds the geometric invariant moment values to the query image and measures the similarity as described in the above section and from the values top n images which is similar to query image is displayed.

4. IMPLEMENTATION & RESULTS

4.1 MOMENT BASED FEATURE EXTRACTION:

After the images are preprocessed and individual features are extracted, the geometric invariant moments of each feature and edge (using convex hull) of the images are calculated and stored.

	m1	m2	m3	m4	m5	m6	m7
Edge	1.3259	4.5752	5.4884	7.5733	17.6648	10.6759	15.0769
feature1	1.3063	4.623	6.6855	6.4159	14.8135	9.3045	13
feature2	1.2291	5.8471	5.3102	7.3771	14.665	10.9445	13.8928
feature3	0.0684	3.1525	1.7748	8.0084	13.3627	10.1036	14.4258

Table 4.1.1 shows the moment values of the figure 2.4(a)

After the calculation of the seven geometric invariant moments for the shape feature of the trademark image, the next step is to minimize the entire database set. This is done by considering the outer edge of the trademark image. The moment values of the outer edge of all the images in the database are trained using neural network. 150 images are taken for the training of neural network. This database consisting of 150 images is further subdivided into four databases. These databases are trained using neural network to find out the efficiency.

We here compare the efficiency of the training of feed forward neural network and radial basis function network.

Learning rate = 0.03, epochs=120, mc=0.6, number of hidden nodes=5, goal = 0.

Training Set		Testing Set		No Of Images That Are Classified		No Of Images That Are Misclassified	
Circle	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**
15	15	60	60	47	46	13	14
30	30	45	45	35	37	10	8
45	45	30	30	26	27	4	3
60	60	15	15	12	13	3	2

Table 4.1.2 Training using feed forward network

Learning rate=0.1 epochs=120, mc=0.6, number of hidden nodes=5, goal = 0.

Training Set		Testing Set		No Of Images That Are Classified		No Of Images That Are Misclassified	
Circle	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**
15	15	60	60	48	45	12	15
30	30	45	45	36	39	9	6
45	45	30	30	25	26	5	4
60	60	15	15	11	13	4	2

Table 4.1.3 Training using feed forward network

Learning rate=0.2, epochs=120, mc=0.6, number of hidden nodes=5, goal=0

Training Set		**Testing Set**		**No Of Images That Are Classified**		**No Of Images That Are Misclassified**	
Circle	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**
15	15	60	60	44	48	16	12
30	30	45	45	35	40	10	5
45	45	30	30	24	26	6	4
60	60	15	15	11	13	4	2

Table 4.1.4 Training using feed forward network

Here in the above three tables the learning rate are varied and the performance of the network is analyzed. The feed forward network with learning rate 0.03 has better outcome and hence it is considered.

Epochs=120, Goal=1, Hidden nodes=5, Spread constant=500.

Training Set		**Testing Set**		**No Of Images That Are Classified**		**No Of Images That Are Misclassified**	
Circle	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**	**Circle**	**Triangle**
15	15	60	60	46	48	14	12
30	30	45	45	36	38	9	7
45	45	30	30	28	29	2	1
60	60	15	15	14	15	1	0

Table 4.1.5 Training using RBF network

From the table 4.1.2 and 4.1.5 , a graph is drawn as shown below

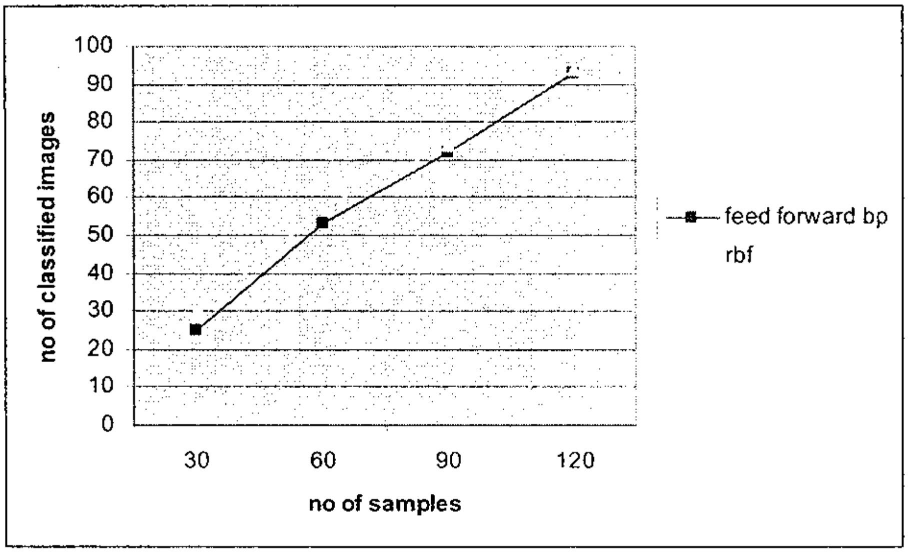


Figure 4.1 Graph – no of samples vs no of classified images in Feed forward backpropagation and RBF networks

Comparing the above results obtained from above tables and graph RBF network has better performance when compared to feed forward back propagation network. Hence we use RBF network for our classification.

5. PROGRAMMING ENVIRONMENT

5.1. HARDWARE REQUIREMENTS:

1. Processor : Intel Pentium IV
2. Processor Speed : 1.6 GHz
3. Memory : 256 MB

5.2 SOFTWARE REQUIREMENTS

1. Operating System : MS Windows XP
2. MATLAB 7.5

6. FUTURE ENHANCEMENTS

The application has been built with only basic functionalities, with the emphasis given on the working of the algorithm. Extended functionality can be provided to both the User Interface and to the algorithm used itself.

6.1. ENHANCEMENTS TO THE UI:

Some of the features that can be added to the User Interface include:

- Enable the user to select the level of accuracy for the retrieval of results.
- Generate reports that provide a graphical representation of the various parameters involved in the retrieval process in the form of graphs, charts etc.,

6.2. ENHANCEMENTS TO THE ALGORITHM:

Some of the enhancements that can be done to the algorithm in future are given below:

The algorithm can be modified in such a way that

- It works for complex images i.e., images which have a complex background.
- It works for colored images .
- It works for images that are other than geometric shapes.
- It works for 3-D images.

7. CONCLUSION

The "Retrieval Of Trademark Images Using Shape Feature" method was successfully implemented and evaluated using Matlab 7.5 on Windows XP operating system. The results arrived at are significant and accurate enough, fulfilling all the claims made. The system can help answer many real-world demands.

Unlike traditional information retrieval systems, in which implicit queries might return bad results or even no result, this system always gives the best possible solutions. The design is user-friendly and comprehensive so that even an untrained user can learn it with ease. The system is also easily specialized; the basic features of the system can be modified to suit various needs. By including information that is particular to a situation, the system can be made more specific and the efficiency improved.

The database used for the demonstration of the working of this algorithm was chosen carefully, so that the basic functionality of the algorithm is highlighted. Further research can strive to improve the scalability of the algorithm by having more built-in features that are customizable by the user.

Overall, the project was challenging to implement and provided deep insights into the field of image processing and image retrieval.

8. APPENDIX

MATLAB 7.5 SAMPLE CODE

SAMPLE CODE TO EXTRACT FEATURES

```
im=imread(filename);
figure(13),imshow(im);
j=imresize(im,[200 200]);
le = graythresh(j);
b = im2bw(j, le);
k=uint16(b);
im1=medfilt2(k,[3 3]);
BW = edge(im1,'sobel');
[imx,imy]=size(BW);
msk=[0 0 0 0 0;
   0 1 1 1 0;
   0 1 1 1 0;
   0 1 1 1 0;
   0 0 0 0 0;];
B=conv2(double(BW),double(msk),'valid');
L = bwlabel(B,8);
mx=max(max(L))
for j= 1:mx
  fprintf('feature=%d',j) ;
[r,c] = find(L==j);
rc = [r c];
[sx sy]=size(rc);
n1=zeros(imx,imy);
for i=1:sx
  x1=rc(i,1);
```

```
  y1=rc(i,2);
  n1(x1,y1)=255;
end
level = graythresh(n1);
Ibw = im2bw(n1, level);
r=invmoments(Ibw);
e=strcat('B',int2str(j));
  xlswrite('query.xls',r,'sheet1',e);
moment=r
end
```

SAMPLE CODE FOR CONVEX HULL

```
k=uint16(BW);
threshold = graythresh(k);
bw = im2bw(k,threshold)
bw = bwareaopen(bw,8);
se = strel('disk',0,0);
bw = imclose(bw,se);
bw = imfill(bw,'holes');
figure,imshow(bw)
rp = regionprops(double(bw), 'all');
xy = rp.ConvexHull;
line(xy(:,1),xy(:,2),'Color','yellow','LineWidth',2)
[r,c] = size(j);
mask = poly2mask(xy(:,1), xy(:,2), 200, 200);
figure,imshow(mask)
a=uint16(mask);
le = graythresh(a);
b = im2bw(a, le);
a=uint16(b);
```

```
e=edge(a,'sobel');
figure,imshow(e)
```

SAMPLE CODE TO FIND SIMILAR IMAGES

```
t=1;
a=1;
b=1;
x=1;
g=0;
g
for i=1:3
   cou=1;
   if(i==1)
   for k=1:78
      g=0;
      c=0;
   for j=2:8
      c(t)=abs(valf1(a,j)-qf(b,j));
      g=g+c(t);
      t=t+1;

   end
   v=max(c);
   h=g-v;
   a=a+1;
   name=valf1(k);
   dif1(cou,1)=name;
   dif1(cou,2)=h;
   cou=cou+1;

   end
```

```
if (nof==1)
   i=4;
end
end
a=1;
b=2;
t=1;

if(i==2)

   for k=1:78
    g1=0;
    c=0;
for j=2:8
    c(t)=abs(valf2(a,j)-qf(b,j));
    g1=g1+c(t);
    t=t+1;

end
v=max(c);
h=g1-v;
a=a+1;
name=valf2(k);
dif2(cou,1)=name;
dif2(cou,2)=h;
cou=cou+1;
   end
if(nof==2)
   i=4;
end
end
```

```
  a=1;
  b=3;
  t=1;

  if(i==3)

    for k=1:78
      g1=0;
      c=0;
  for j=2:8
      c(t)=abs(valf3(a,j)-qf(b,j));
      g1=g1+c(t);
      t=t+1;

  end
  v=max(c);
  h=g1-v;
  a=a+1;
  name=valf3(k);
  dif3(cou,1)=name;
  dif3(cou,2)=h;
  cou=cou+1;
   end
  end
end
per(nof,dif1,dif2,dif3);
function per(nof,pf1,pf2,pf3)

max=30;
min=0;
cou1=1;
```

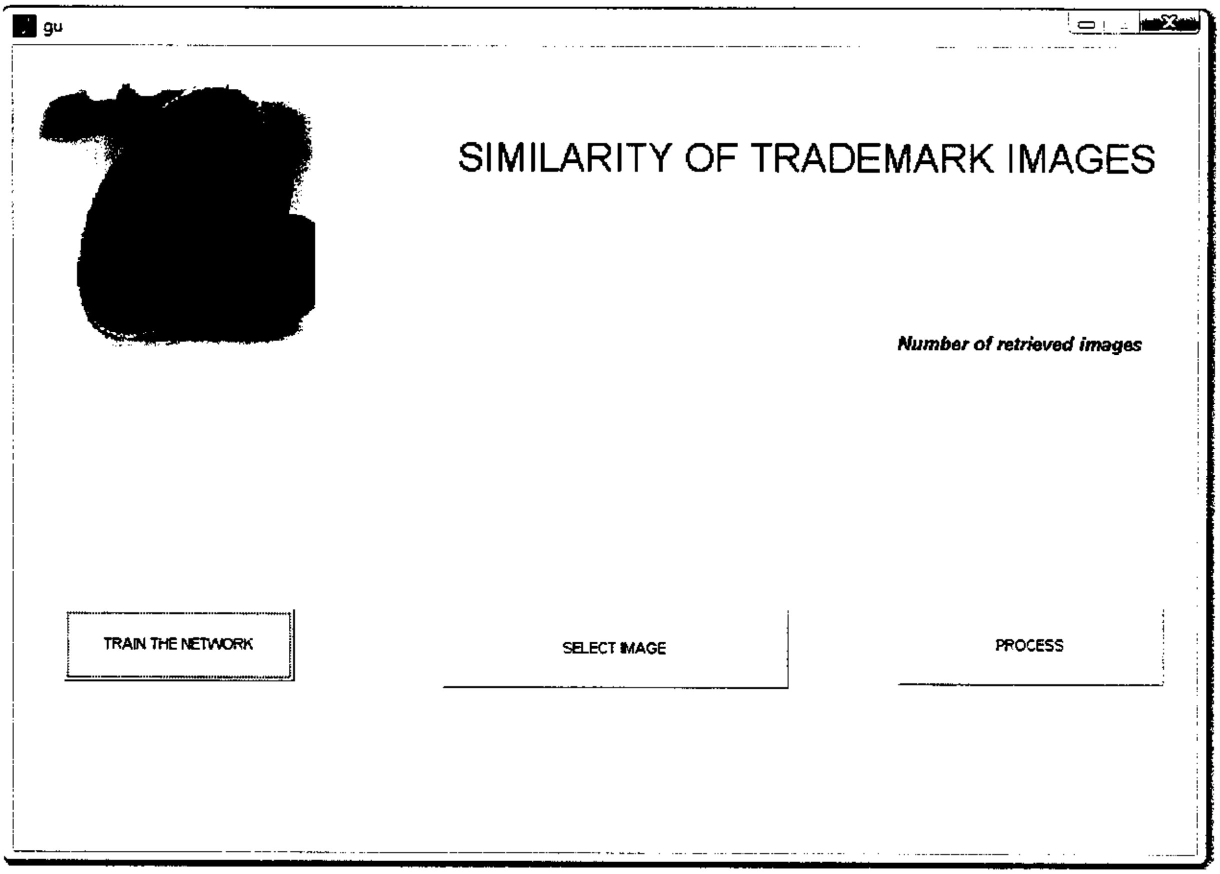
```
cou2=1;
cou3=1;
nof
for i=1:78
  c=pf1(i,2);
  if(c>30)
     u=0;
  else
  g=(max-c)/(max-min);
  u=g*100;
  end

  pf1(cou1,3)=u;
  cou1=cou1+1;
  if(nof==2)
   c1=pf2(i,2);


   if(c1>30)
     u1=0;
  else
  g1=(max-c1)/(max-min);
  u1=g1*100;
  end
  pf2(cou2,3)=u1;
  cou2=cou2+1;

  end



  if(nof>=3)
     c1=pf2(i,2);
```

```
  if(c1>30)
     u1=0;
  else
  g1=(max-c1)/(max-min);
  u1=g1*100;
  end
  pf2(cou2,3)=u1;
  cou2=cou2+1;

  c2=pf3(i,2);
  if(c2>30)
     u2=0;
  else
  g2=(max-c2)/(max-min);
  u2=g2*100;
  end
  pf3(cou3,3)=u2;
  cou3=cou3+1;
  end
end
pf1
pf2
pf3
ressort(nof,pf1,pf2,pf3);
function ressort(nof,avf1,avf2,avf3)
global fr;
cou=1;
avf1
for i=1:78
     if(nof>=3)
      av=(avf1(i,3)+avf2(i,3)+avf3(i,3))/3;
```

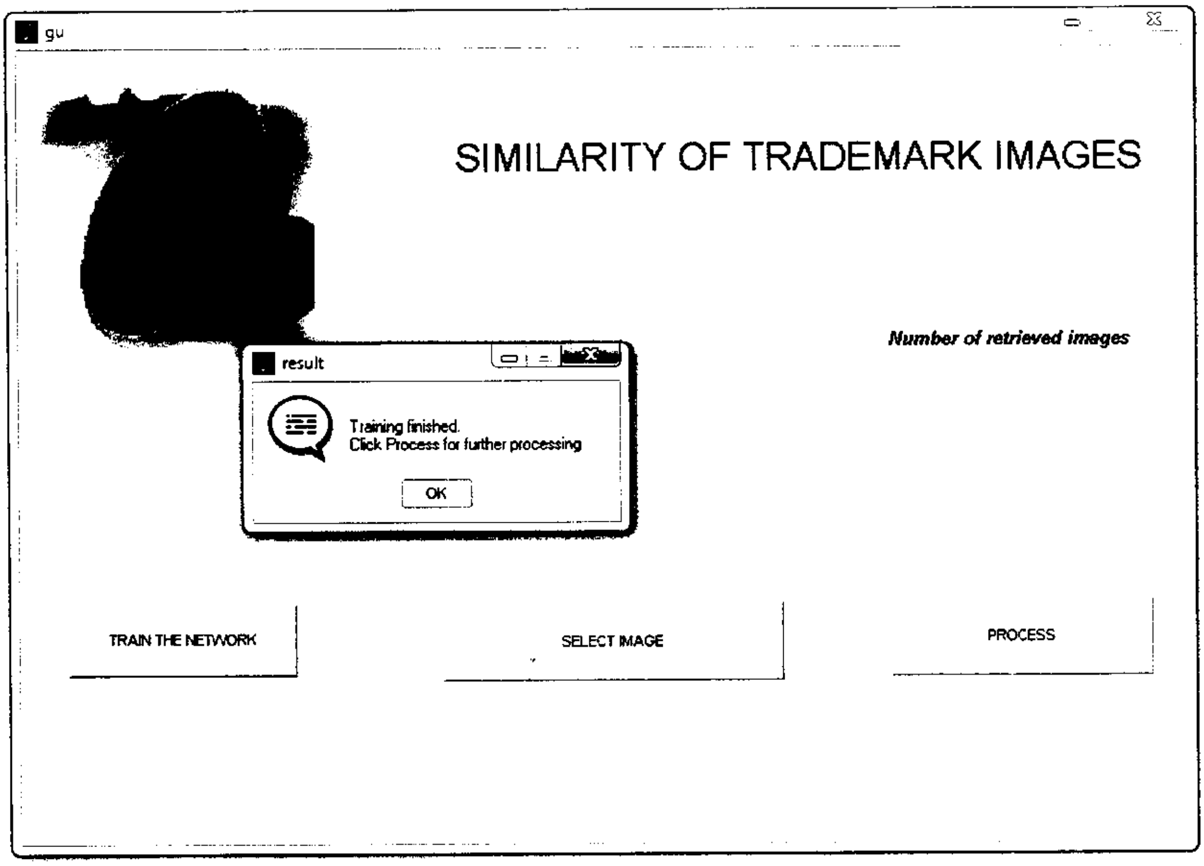
```
        nam=avf1(i);
        reval(cou,1)=nam;
        reval(cou,2)=av;
        cou=cou+1;

    elseif(nof==2)
    av=(avf1(i,3)+avf2(i,3))/2;
    nam=avf1(i);
    reval(cou,1)=nam;
        reval(cou,2)=av;
        cou=cou+1;

    else
        av=avf1(i,3);
        nam=avf1(i);
        reval(cou,1)=nam;
        reval(cou,2)=av;
        cou=cou+1;
    end
end
reval
fr=-sortrows(-reval,2);
fr
```

SNAPSHOTS:

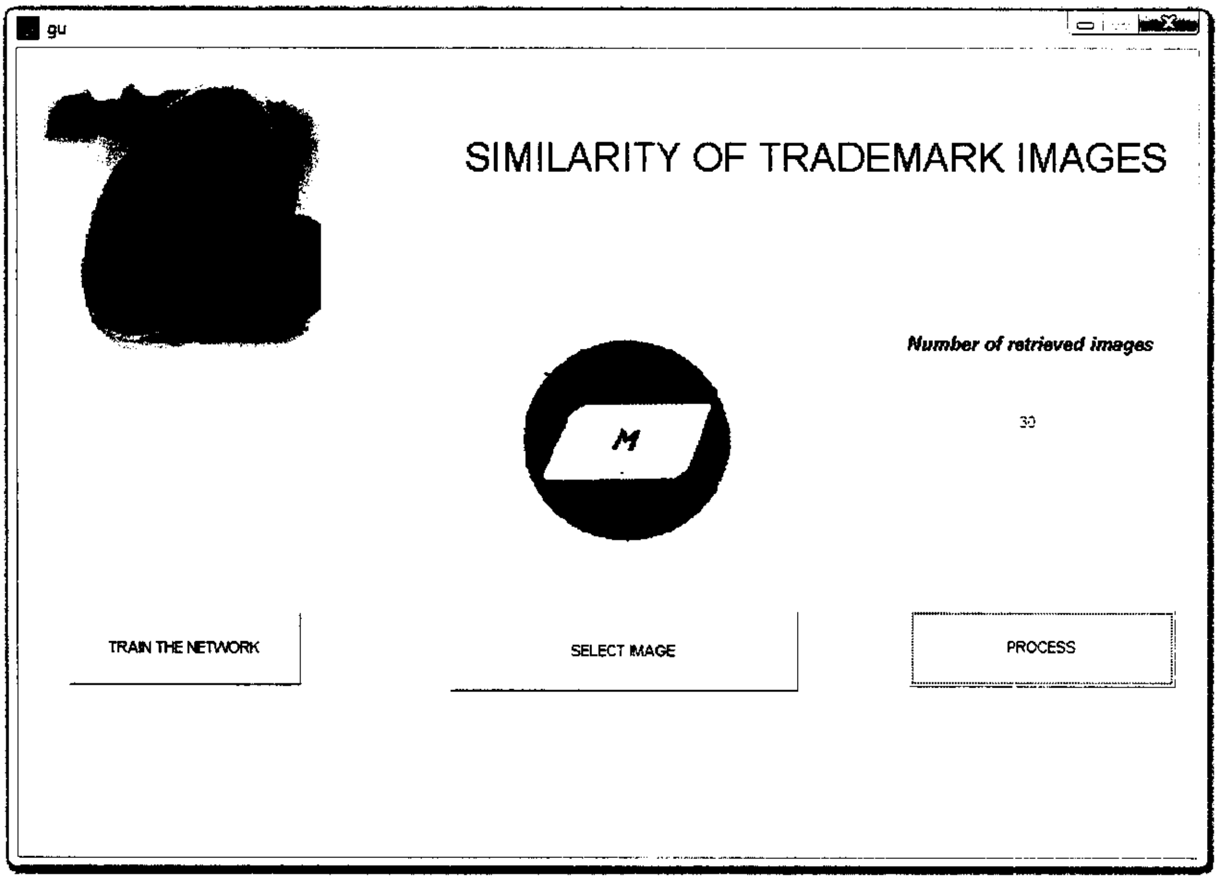
MAIN PAGE



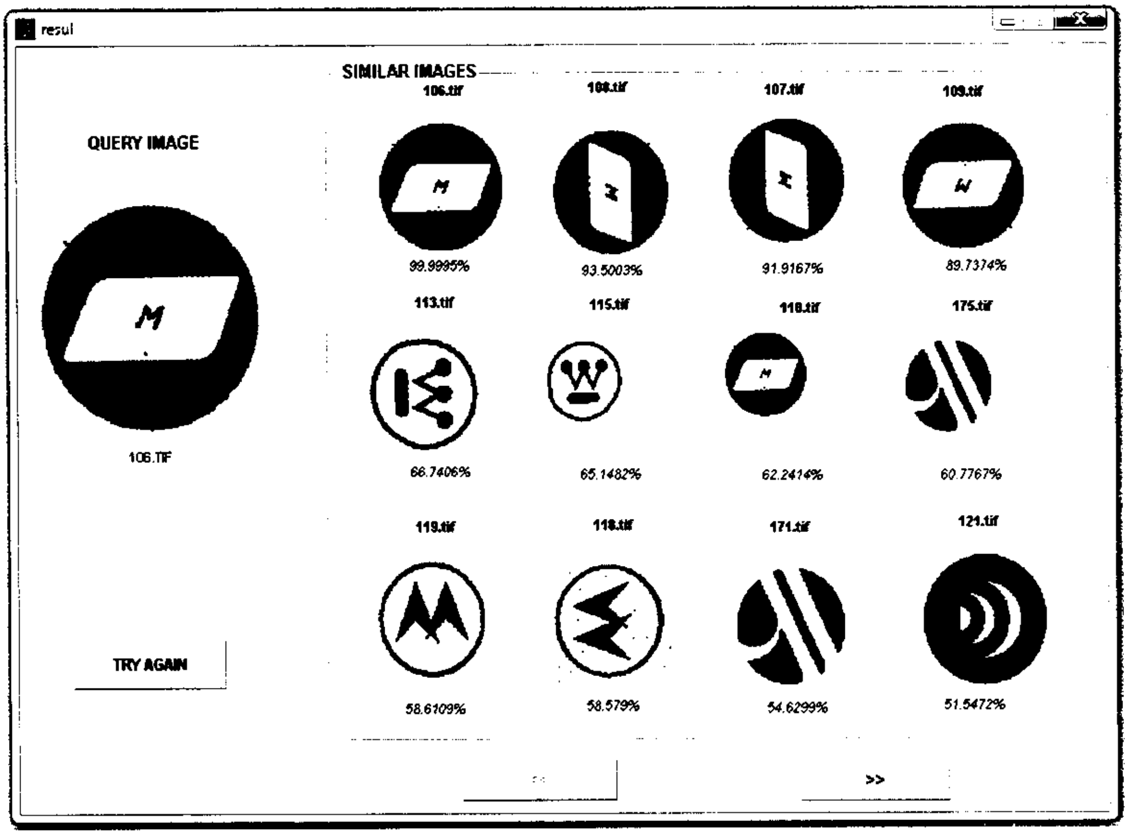
TRAINING THE NETWORK

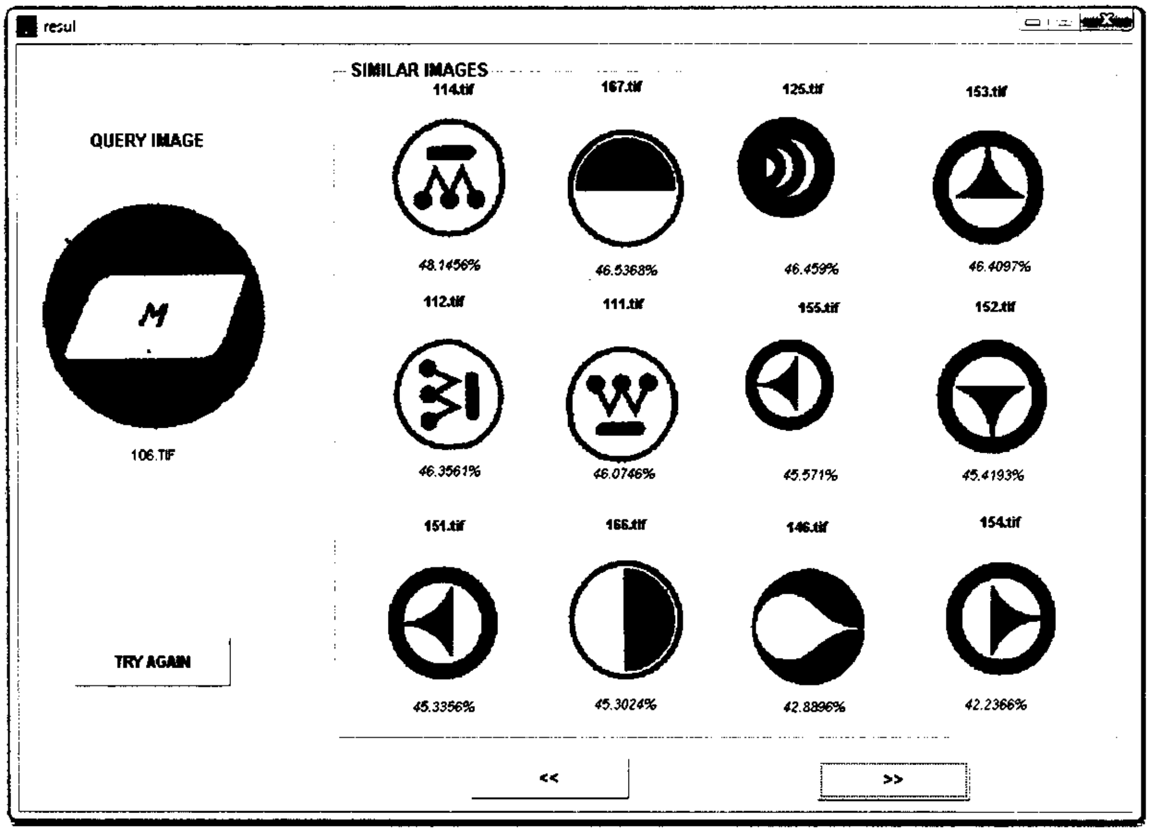


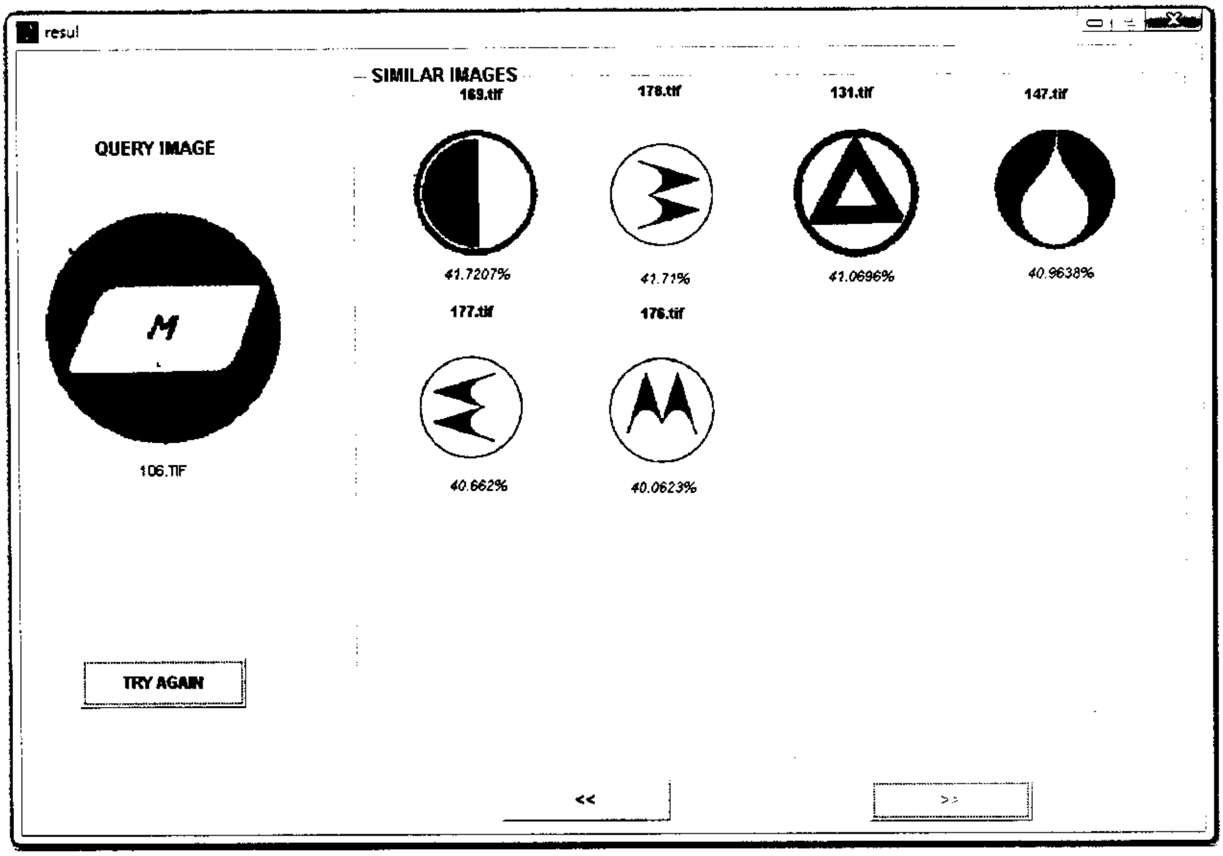
SELECTING THE QUERY IMAGE



DISPLAYING SIMILAR IMAGES




resul
SIMILAR IMAGES
QUERY IMAGE
106.TIF
TRY AGAIN
114.tif
167.tif
125.tif
153.tif
48.1456%
46.5368%
46.459%
46.4097%
112.tif
111.tif
155.tif
152.tif
46.3561%
46.0746%
45.571%
45.4193%
151.tif
166.tif
146.tif
154.tif
45.3356%
45.3024%
42.8896%
42.2366%
<<
>>


resul
SIMILAR IMAGES
QUERY IMAGE
106.TIF
169.tif
178.tif
131.tif
147.tif
41.7207%
41.71%
41.0696%
40.9638%
177.tif
176.tif
40.662%
40.0623%
TRY AGAIN
<<
>>

9. REFERENCES

[1] Jain A.K and Aditya Vailaya, **"Shape-based Retrieval: A Case Study with Trademark Image Databases"**, Pattern Recognition 31(9), 1369-1390 (1998).

[2] John P. Eakins, Jago M. Boardman, and Margaret E.Graham. "**Similarity retrieval of trademark images**".*IEEE MultiMedia*, 5(2):53–63, 1998.

[3]Alwis, S. and Austin, J. "**Trademark image retrieval using multiple features**". Presented at *CIR-99: The Challenge of Image Retrieval*, Newcastle-upon-Tyne, U.K., Feb. 1999.

Books:

1. Gonzalez R. and Woods B.E., "Digital Image Processing", Addison Wesley, 1993.
2. Donald Hearn, M. Pauline Baker, "Computer Graphics", Second Edition, Prentice-Hall of India, 1994.
3. Stephen J. Chapman, "MATLAB Programming for Engineers", Third Edition, Thomas Learning, 2004.
4. Rudra Pratap, "Getting Started with MATLAB 7 A Quick Introduction for Scientists and Engineers", Oxford University Press, 2006.

Websites:

1. www.imageprocessingplace.com
2. www.mathworks.com