**M.Tech. (Computer Science) Dissertation Series**

# Studies on Content Based Image Retrieval System (CBIR) with relevance feedback using Neural Networks and MPEG-7 features

M.Tech. Dissertation Report

*a dissertation submitted in partial fulfillment of the requirements for the M.Tech.(computer Science) degree of the Indian Statistical Institute*

**By**

**Nagendar G**
M.Tech-Computer Science
Roll No: CS0728

under the supervision of

**Prof. Malay K Kundu**
Machine Intelligence Unit
ISI, Kolkata
AND

**Dr. Pradipta Maji**
Machine Intelligence Unit
ISI, Kolkata

**INDIAN STATISTICAL INSTITUTE**
203, Barrackpore Trunk Road,
Kolkata-700108, West Bengal, India.

# Indian Statistical Institute

## Certificate of Approval

This is to certify that the thesis entitled **Studies on Content Based Image Retrieval System (CBIR) with relevance feedback using Neural Networks and MPEG-7 features** by Nagendar G towards partial fulfillment for the degree of M.Tech. in Computer Science at Indian Statistical Institute, Kolkata, embodies the work done under my supervision.

(Prof. Malay K Kundu)
Date: MIU - Indian Statistical Institute
Kolkata.

# ACKNOWLEDGEMENT

I would like to take this opportunity to thank **Prof. Malay K Kundu and Dr P. Maji**, Machine Intelligence Unit, ISI-Kolkata for their valuable guidance and inspiration. Their plesant and encouraging words have always kept my spirits up.

I also would like to thank all my friends, classmates and my family members for their support and motivation to complete this project.

Date:

Nagendar G
M.Tech(CS)
Indian Statistical Institute
Kolkata

## abstract

Effective image retrieval from a large database is a difficult problem and is still far from being solved. Hence, the retrieval of relevant images, based on measuring the similarity between automatically derived features(color, texture and shape, etc) of the query image and that of the images stored in the database, aproblem popularly known as content based image retrieval is a highly challenging task.. In a conventional CBIR approches, an image is usually represented bya set of features, where the feature vector is a point in a multidimensional space. It also has numerous applications in areas like Biomedicine (X ray, Pathology, CT, MRI,..), Crime Management, Commercial (Fashion, Catalogue, Design, Journalism,..) and it is widely used in medical services. Although extensive research has been performed on the CBIR over several decades, we have yet to acheive the accuracy from a fully automated CBIR system. In this approach we have used MPEG 7 features to extract the image information with relevance feedback using neural networks. The algorithm implemented in c.

# Contents

# Chapter 1

# Introduction

Due to revolutionary growth of digital imaging and internet technology, image retrieval from a large image collection became an important research issue.

Search engines such as Google retrieves images based on the text associated with them and do not look at the content of the image. Observe that as you move on from page to page, the number of irrelevant images you see get increased.

To overcome this we will use CBIR, which retrieves the images using its content.
Why CBIR is Important
If we continue with image retrieval based on the text annotated with images, there are a lot of drawbacks.
Text annotation is subjective and varies across different cross sections of people.
Annotating a huge database manually is a very humungous task.

CBIR retrieves images using its content, So we will get good results using cbir

## 1.1 Defination is CBIR

For a given query image the process of retrieving similar images from a large collection of Image databases on the basis of its features (such as colour, texture and shape)
Query means
An image we already have
Rough sketch we Draw
A Symbolic description of what we want

CBIR techniques are generally different from conventional information retrieval techniques due to the following reasons
Unstructured nature of image databases
Contains pixel intensities with no inherent meaning
Any kind of reasoning about image content is possible only after the extraction of some

useful image information(Features)

Two important steps in Content Based Image Retrieval are
1. Feature Extraction
2. Similarity Measure

We will represent an image by the features extracted from that and similarity measures are used to calculate the similarity between the images (By calculating the distance between the extracted features). We can use different similarity measures like eucliden, earth movers ...

**Features:**
Features gives the description about the image. They are
Visual Features:
Color, texture, shape, ....
Statistical features:
Histograms,

## 1.2 Working of CBIR System

For each image in the image database and for the Query it extracts the features using some algorithms and forms a signature ( Signature describes the content of an entire image), Using the similarity measures it calculates the distance between the signature of the Query and all the images in the database. It retrieves the images which are less distance (say up to 20 images) to the Query image. All these retrieved images are relevant images to the Query.

It may happen that all the retrieved images may not be similar to the query image, to get more similar images we will use Relevance Feedback

## 1.3 Relevance Feedback

Relevance feedback, is a supervised active learning technique used to improve the effectiveness of information retrieval systems.
It works as follows
For a given query, the sy stem first retrieves alist of ranked images (say up to 20) according to some similarity measure, then the user marks the retrieved images as relevant (positive examples) or irrelevant (negative examples) to the query

The system will refine the query based on the feedback, retrieve a new list of images, and

present them to the user. Again user wil mark relevant and irrelevant images then the system refine the query again. User can continue this iterative process up to their satisfaction of results.
There are two methods in the Relevance Feedback
1. Human Controlled Relevance Feedback
2. Machine Controlled Relevance Feedback

In Human controlled Relevance Feedback, user have to mark relevant and irrelevant images in each iteration. According to this feedback the system will retrieve the images. In this method we can use Neural Networks ( MLP, RBF, ..) for the feedback. In the first iteration user will mark relevant and irrelevant images (Two classes), using this class information the network will update its weights.

## 1.4 Image Retrieval Process

There are two kinds of Image Retrieval process which are
1. Image Retrieval without Relevance Feedback

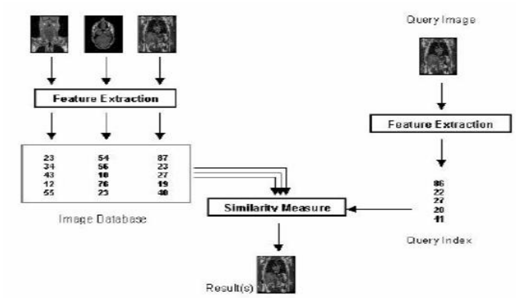



Image Retrieval without Relevance Feedback

2. Image Retrieval with Relevance Feedback

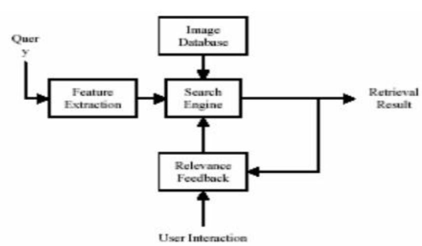



Image Retrieval with Relevance Feedback

## 1.5 Previous Techniques in Image Retrieval

In this section we have mentioned some other techniques in image retrieval. An image retrieval system consists of 2 parts, forming the signature of the image and the similarity measure between a pair of images. The mathematical description of an image for retrieving purpose is the signature of an image. Different techniques are available for forming the signature of an image. Signature is formed by the different features of the image. There is some great work going on region based visual signature, which is formed by the segmentation of the image. The main step in region based visual signature is to segment the image properly. The widely used some segmentation approches are K means, KD tree, normalized cuts criterion where image segmentation is mapped to a weighted graph partitioning problem etc ..... Another approch is to use Gabor Filters for forming the signature.

Several similarity measures are available for similarity matching, some of them are eucliden distance, earth movers, mahalanabois distance, l1 distance measure .....

In this paper we have used histograms with l1 distance measure for image retrieval then we have used Neural Networks for Relevance feedback. By using color structure descriptor we obtained the histogram for color, which gives the information about color. Unlike color histogram, it captures both color content and information about the structure of this content. By using Edge histogram descriptor we obtained the histogram for edges and texture, which gives the information about local, semi global and global distribution of edges also gives the information about texture. For the relevance feedback we have used the neural networks. In the neural networks we used Multi Layer Perceptron and Radial Basis Function network, in which radial basis function network giving slightly good results compared to multilayer perceptron network.

## 1.6 Organization of the report

The following chapters gives the information about image retrieval. Chapter 2 covers the MPLEG 7 color and edge descriptors. Chapter 3 covers neural networks(MLP/ RBF) which we have used in relevance feed back mechanism. Chapter 4 covers our proposed technique. Finally chapter 5 covers the conclusion and future work about our technique.

# Chapter 2

# MPEG 7 FEATURES

The MPEG-7 standard, formally named "Multimedia Content Description Interface", provides a rich set of standardized tools to describe multimedia content. The main objective of the MPEG-7 visual standard is to provide standardized descriptions of streamed or stored images (visual low-level descriptors) that help users or applications to identify, categorize or filter images or video. These low-level descriptors can be used to compare, filter or browse images purely on the basis of non-textual visual descriptions of the content

## 2.1 Color Descriptors

Color is one of the most widely used visual features in image and video retrieval. Color features are relatively robust to viewing angle, translation and rotation of the regions of interest. The currently standard six color descriptors represent different aspects of the color features, including color distribution, spatial layout of color and spatial structure of color. One of them is color structure descriptor.

## 2.2 Color Structure Descriptor

The Color structure descriptor is a color feature descriptor that captures both color content (similar to a color histogram) and information about the structure of this content. Its main functionality is image-to-image matching and it is very useful in image retrieval, where an image may consist of either a single rectangular frame or arbitrarily shaped, possibly disconnected, regions. The extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of 8x8 pixels that slides over the image, instead of considering each pixel separately.

Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. Here color values are represented in the

HMMD color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins.

CSD is characterized by a color structure histogram for M quantized colors $c_m$ and is expressed as
h( m) m= 1,..., M
where M 256, 128, 64, 32 and the bin value h(m) is the number of structuring elements containing one or more pixels with color cm . Unlike the conventional histogram, the color structure histogram is extracted from an image by accumulation using an 8×8-structuring window. The structuring element scans the image and counts the number of times a particular color is contained with in the structuring element.

The final value of h(m) is determined by the number of positions at which the structuring element contains $c_m$ .
**ALGORITHM:**

**Step 1:** Transform the RGB space in to 3 dimensional HMMD space(Hue, sum and difference)
**Step 2:** Quantize the HMMD space in to one of the following cells. 256, 128, 64, 32 as follows
Divide the difference axis in to 5 sub intervals [0,6), [6, 20), [20, 60), [60, 110) and [110, 255) and using the below table partion the HMMD space.
**Step 3:** Take a 8× 8 structuring element and scan the image through every pixel and counts the number of times a particular color is contained with in the structuring element. Update the histogram correspondingly

## 2.3 HMMD color space quantization

The HMMD (HueMaxMinDiff) color space is closer to a perceptually uniform color space. The double cone shape confines this color space as shown in Figure . The component names, 'Max', 'Min' and 'Diff are according to the following transform equations between RGB and HMMD:

$$\text{if( Max == Min ) Hue=0;}$$

$$otherwise : if(Max == R \ and \ G >= B) \ then \ Hue = 60 * (G - B)/(Max - Min)$$

$$elseif(Max == R \ and \ G < B)then \ Hue = 360 + 60 * (G - B)/(Max - Min)$$

$$elseif(G == Max) \ Hue = 60 * (2.0 + (B - R)/(Max - Min))$$

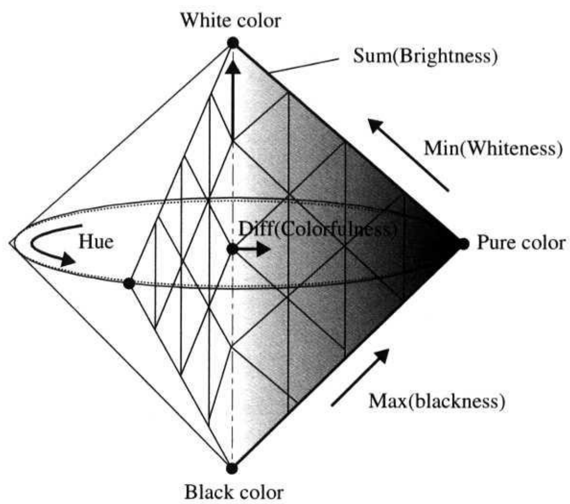
$$elseHue = 60 * (4.0 + (R - G)/(Max - Min))$$

$$Max = max(R, G, B); Min = min(R, G, B); Diff = Max - Min;$$

Even though the four components are identified in the name of the HMMD color space, one more component, Sum, can be defined.

$$Sum = (Max + Min)/2;$$

Therefore, a total of five components are identified in this color space. However, a set of three components, H, Max, Min or H, Diff, Sum, is sufficient to form the HMMD color space and specify a color point. The semantics of each component is distinct and described as follows. Hue (H = [0, 360]) represented by the angle from 0 to 360 specifies one color family from another, as red from yellow, green, blue or purple. Max (=[0,1]) specifies how much black color is present. Min (=[0,1]) specifies how much white color is present. Diff (=[0,1]) specifies how much a color is close to pure colors. Finally, Sum (=[0,1]) specifies the brightness of the color.




color structure descriptor uses the nonuniform quantization of the HMMD color space. HMMD space can be defined in three dimensions using sum- and diff-axes as well as hue angle. Four nonuniform quantizations of HMMD space are defined in the MPEG-7 Standard. The four quantizations partition the space into 256, 128, 64 and 32 cells respectively. These quantizations of HMMD space are defined via five subspaces as follows
The diff-axis, itself defined on the interval [0, 255], is cut into five subintervals: [0,6), [6, 20), [20, 60), [60, 110) and [110, 255). This ID partition of the diff-axis implicitly defines five subspaces numbered 0, 1 , . . . , 4, respectively.

A partition of HMMD is obtained by partitioning the ranges of hue and sum into uniform intervals within each subspace according to the following table.

<table>
<tr><th>No. of cells</th><th colspan="2">256</th><th colspan="2">128</th><th colspan="2">64</th><th colspan="2">32</th></tr>
<tr><th>Subspace</th><th>Hue</th><th>Sum</th><th>Hue</th><th>Sum</th><th>Hue</th><th>Sum</th><th>Hue</th><th>Sum</th></tr>
<tr><td>0</td><td>1</td><td>32</td><td>1</td><td>16</td><td>1</td><td>8</td><td>1</td><td>8</td></tr>
<tr><td>1</td><td>4</td><td>8</td><td>4</td><td>4</td><td>4</td><td>4</td><td rowspan="2">4</td><td rowspan="2">4</td></tr>
<tr><td>2</td><td>16</td><td>4</td><td>8</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>3</td><td>16</td><td>4</td><td>8</td><td>4</td><td>8</td><td>2</td><td>4</td><td>1</td></tr>
<tr><td>4</td><td>16</td><td>4</td><td>8</td><td>4</td><td>8</td><td>1</td><td>4</td><td>1</td></tr>
</table>

The table consists of four tables (one for each quantization of HMMD). For each quantization, the table tells how to partition the subspaces to yield the overall partition.

## 2.4 Edge Histogram Descriptor

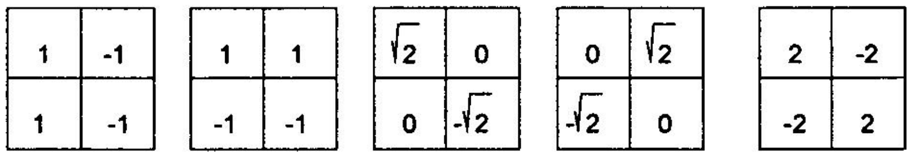
Histogram is the most commonly used characteristic to represent the global feature composition of an image. It is invariant to translation and rotation of the images and normalizing the histogram leads to scale invariance. By using the above properties, the histogram is considered to be very useful for indexing and retrieving images.

Edge in the image is considered an important feature to represent the content of the image. In MPEG-7, there is a descriptor for edge distribution in the image known as Edge Histogram Descriptor. The Histogram consists 80 bins contains the information about the local edges. There is semi global and global edge histograms which represents the semi global and global edge information of the image and these can be derived from the above local histogram. Semi global and global edge histograms contains 65 and 5 bins respectively.

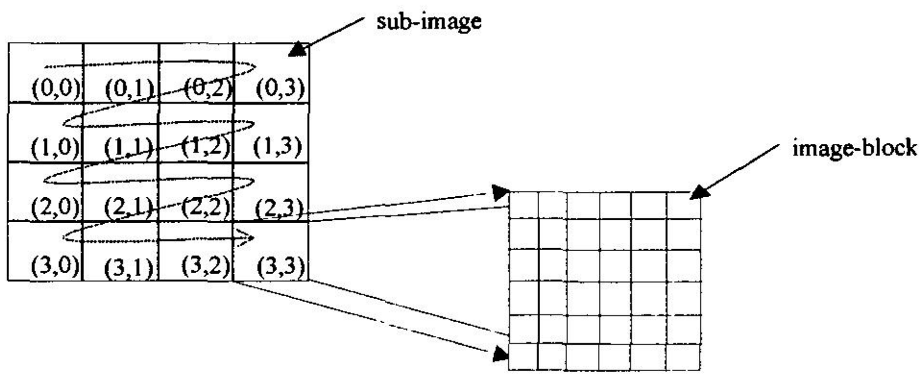
Firstly we divide the image space into 4x4 sub-images. Then, for each sub-image, we generate an edge histogram to represent edge distribution in the sub-image. To extract the different edges in the sub image, the sub-image is further divided into non-overlapping square blocks called image-blocks. These image blocks are used to extract both directional non-directional edges in the sub image. Regardless of the image size, we divide the sub image in to fixed number of image blocks. The size and the number of image-blocks in each sub-image will be calculated as follows

$$x = \sqrt{\frac{Image\_wigth \times Image\_height}{desired\_num\_block}}$$
$$block_{s}ize = \lfloor \frac{x}{2} \rfloor \times 2$$

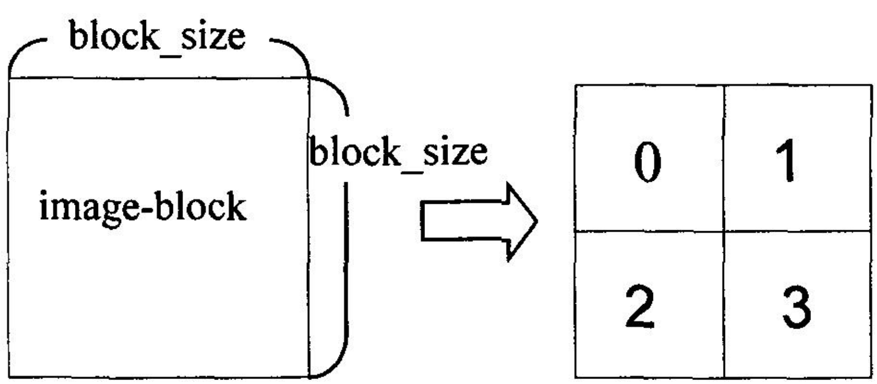
Here, image_ width and image_ height represent the number of columns and rows respectively. The desired_ num_ block is given and fixed. Edge feature is extracted from the image block by further dividing the image block in to four sub blocks. Mean gray value of each block is considered as the gray value of that block. By applying the following 5 kinds of edge templates on the sub block, we will determine the type of edge present in the image block.




a) Vertical edge b) Horizontal Edge c) Diagonal 45 degree d) Diagonal 135 e) Non Directional Edge

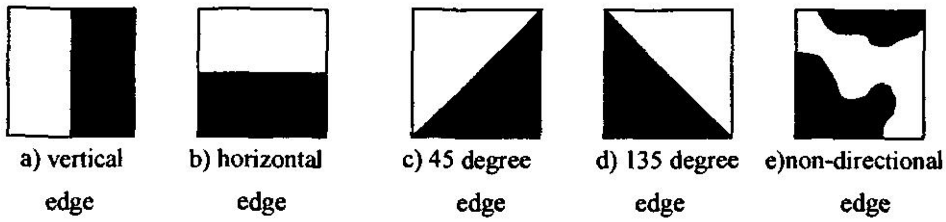



Sub images and Image blocks



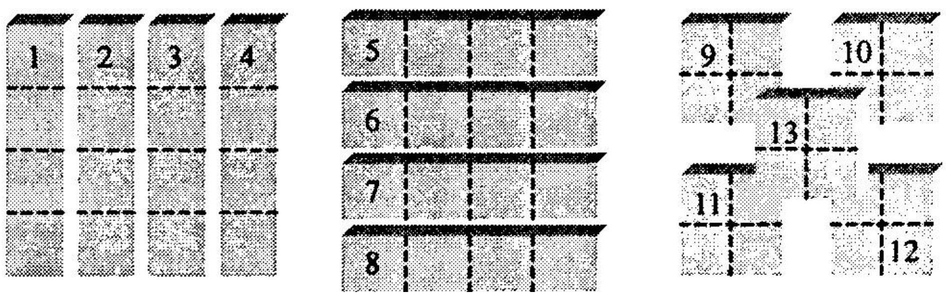

Dividing an image block in to 4 sub blocks

In this way for each of 16 sub images we will get a local edge histogram of 5 bins representing the following 5 kinds of edges.

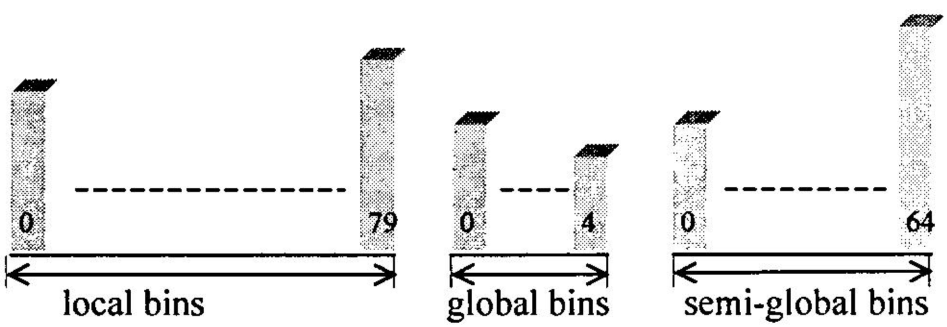



To achieve high retrieval performance, the local edge histogram may not be sufficient. Rather, we may need an edge distribution information for the whole image and some horizontal and vertical semi global edge distributions as well.

That is we need global and semi global edge histograms, which can be calculated from the local edge histogram. The global edge histogram represents the edge distribution of whole image. It contains 5 bins representing all 5 kinds of edges. For the semi global edge histograms, we cluster all sub images in to 13 different clusters as follows and calculate the histogram for each cluster.




So there are 80 bins (local) + 5 bins (global) + 65 bins (semi global) = 150 bins




Local, Global, Semi global Edge Histograms

# Chapter 3

# Neural Networks

Massively parallel interconnections of simple processing elements that interact with objects of the real world in a manner similar to biological systems. It has the natural property for storing experience based knowledge and making it available for use.

Knowledge is acquired from the environment through a learning process, the synaptic weights are used to store the acquired knowledge. Neural networks are very useful for classifing the patterns. It will update its weights through a lerning mechanism on training data, byusing these weights it will classify the test patterens.

## 3.1 Multi Layer Perceptron Network

A multilayer perceptron is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. It is widely used in pattern classification to classify any kind of patterns. It contains one input layer, an output layer and one or more hidden layers. Each layer contains some neurons and each neuron uses a nonlinear activation function. The two main activation functions used in current applications are tan hyperbolic and sigmoid functions, and are described by
Let $v_j h + 1$ be the total input received by neuron j in layer h+1 then

$$v_j h + 1 = \sum y_i h w_{ji} h \text{ - } b_j h + 1$$

Where
$y_i h$ is the output of the ith neuron in the preceding layer h,
$w_{ji} h$ is the weight of the connection from the ith neuron in layer h to the jth neuron in layer h+1, and
$b_j h + 1$ is the bias of the jth neuron in layer h+1.

The output of any neuron, except those in the input layer, is given as:

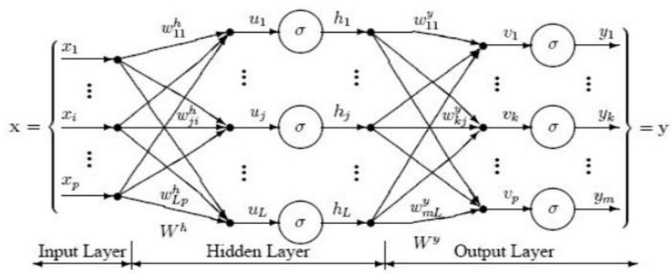
$$y_j h = \frac{1}{1+exp(-v_j)}$$

For nodes in the input layer:

$$y_j0 = x_j0$$

where $x_j0$ is the jth component of the input vector at the input layer.

Nodes in the input layer transmits the input to all the nodes in the next layer (first hidden layer). Nodes in the other layers performs weighted sum of the inputs to that node and computes the activation function of that weighted sum and it also passes this function value to each node in the next layer. In the output layer these function values are the outputs of the network.




Using the Back Propagation Algorithm the network will adjust its weights.

**Back Propagation Algorithm:**

Initially we will give random values to the weights from hidden layer to output layer once a input from the training data is presanted to the network it will calculates the ouputs in the output layer using these weights, then using these ouput values and desired output values of the training pattern it will update its weights as follows
Learning rule

$$w_{ji}(m+1) = w_{ji}(m) + \Delta w_{ji}(m)$$

Where $w_{ji}(m+1)$ is the weight in $m^{th}$ iteration and $\Delta w_{ji}(m)$ is the Error in the $m^{th}$ itearation from $i^{th}$ node to $j^{th}$ node
$\Delta w_{kj} = \eta \delta_k y_j$
$\eta$ is learning rate and $y_j$ is the input from the $j^{th}$node of previous layer
For the Hidden layer to the output layer

$$\delta_k = (d_j - y_j)y_j(1 - y_j)$$

For the Input layer to the Hidden layer

$$\delta_k = (\sum \delta_i w_{ik})y_k(1 - y_k)$$

$\delta_i$ is obtained from the next layer. $d_j$ is the desired output in the output layer.

## 3.2 Radial Basis Function Network

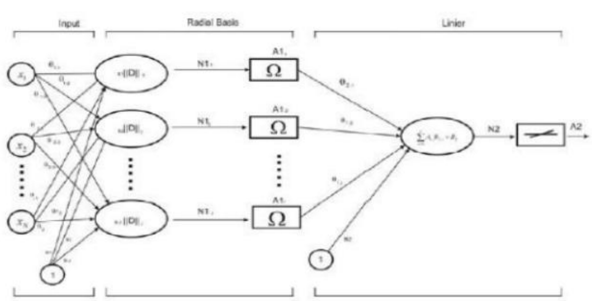
Radial basis functions are embedded into a two-layer feed-forward neural network. It is characterized by a set of inputs and a set of outputs. In between the inputs and outputs there is a Hidden layer. This layer has a variable number of neurons (the optimal number is determined by the training process). Each neuron consists of a radial basis function with a center. When an input vector from the input layer is presented to the network, a hidden neuron computes the Euclidean distance of the input vector from the neurons center and then applies the RBF function to this distance . The resulting value is passed to the output layer. The output units implement a weighted sum of hidden unit outputs.

$$y(x) = \sum_{i=1} w_i \phi(||x - c_i||)$$

Where $\theta$ is the activation function and $c_i$ is the center for the ith node and n is the number of nodes in the hidden layer

In order to use a Radial Basis Function Network we need to specify the hidden unit activation function, the number of processing units, center for each hidden-layer RBF function and a training algorithm for finding the parameters of the network. Finding the RBF weights is called network training. If we have at hand a set of input-output pairs, called training set, we optimize the network parameters in order to fit the network outputs to the given inputs. The fit is evaluated by means of a cost function, usually assumed to be the mean square error. After training, the RBF network can be used with data(Test data) whose underlying properties are similar to that of the training set.

Centers for RBF function is calculated as follows, by using K means clustering we have to find the center of each cluster for the training data, which are then used as the centers for the RBF functions.




RBFN Model

Some commonly used Radial Basis Functions are
Gaussian Function: $\phi(r) = exp(-\beta r^2)$ For some $\beta > 0$
Multiquadric: $\theta(r) = \sqrt{r^2 + \beta^2}$ For some $\beta > 0$

**Training:**

Initially we will give random values to the weights from hidden layer to output layer once a input from the training data is presanted to the network it will calculates the ouputs in the output layer using these weights, then using these ouput values and desired output values of the training pattern it will update its weights as follows

$$w_{ji}(m+1) = w_{ji}(m) + \Delta w_{ji}(m)$$

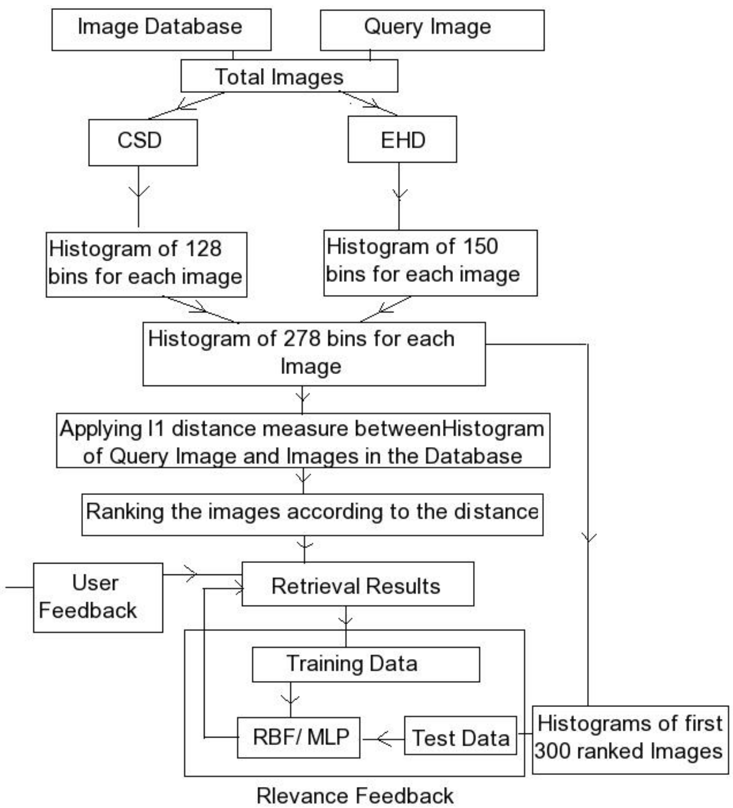
Where $w_{ji}(m+1)$ is the weight in $m^{th}$ iteration and $\Delta w_{ji}(m)$ is the Error in the $m^{th}$ itearation from $i^{th}$ node to $j^{th}$ node

$$\Delta w_{ji}(m) = \eta(d_{j-y_j})x_i$$

Where $d_j$ is the desired output of the input pattern from the $j^{th}$node of output layer, $y_j$ is the output that we got from the $j^{th}$ node of output layer and $x_i$ is the input from the $i^{th}$ node of hidden layer and $\eta$ is learning rate
This is called online learning

# Chapter 4

# PROPOSED WORK




The performance of the image retrieval system is tested upon the SIMPLICITY image database, which consists of 1000 images from 10 different categories: Africa(0), Beach(1), Buildings(2), Buses(3), Dinosaurs(4), Elephant(5), Flowers(6), Horses(7), Mountains(8) and Food(9). Each category has around 100 images, along with some images undergoing changes due to rotation, translation, scaling, noise injection, illumination, etc.
The experiments are performed in the following manner

Our objective is to retrieve similar images for a given query image. By applying CSD(Color Structure Descriptor) to all the images in the database will give a histogram of 256, 128, 64, 32 bins based upon the HMMD Space Quantization. Here we have quantized the Space in to 128 cells. So, Histogram for each image contains 128 bins. On applying CSD we will get a feature vector of size 128 for each image.

By appling EHD(Edge Histogram Descriptor) to all the images in the database will give three histograms, local edge histogram, semi global edge histogram and global edge histogram of 80, 65 and 5 bins repectively for each image. so, bytogether each image contains a histogram of (80+65+5) 150 bins. which forms a feature vector of size 150. So, bytogether CSD and EHD will produce a feature vector of size (128+150) 278. Here we have taken the desired number of blocks as 1100. Now we got the histograms for each image. For a given query image we will retrieve the relevant images as follows

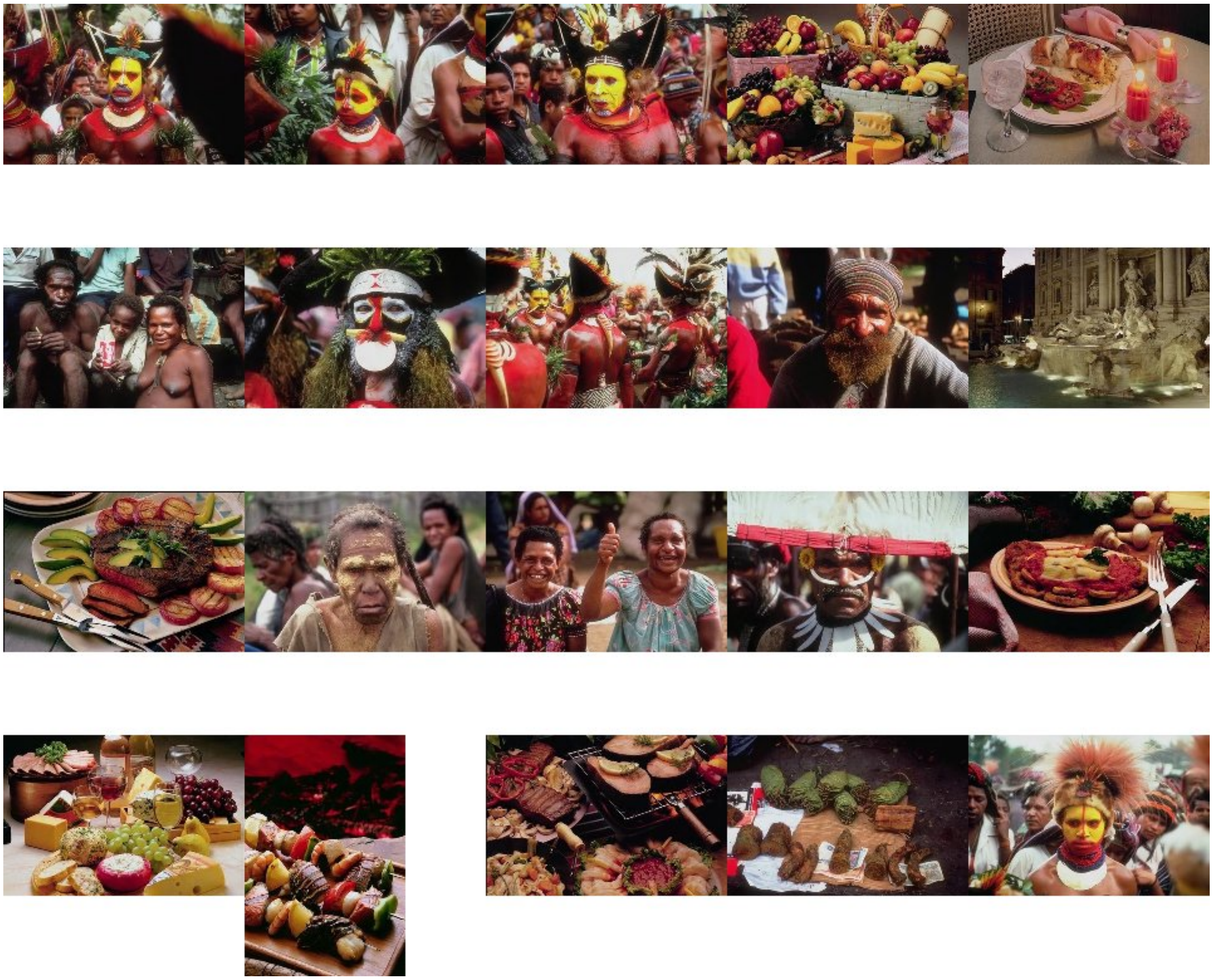
By using L1 distance measure we have computed the dissimilarity between the query image and the each image in the database and we ranked the images according to the this distance in the increasing order. We have displayed the first 20 similar images. It may happen that in these 20 images all are may not similar to the query image and some irrelevant images may have good rank compared to the most relevant images. So, to enhance the accuracy of the system using user feedback we have used Relevance Feedback Meachanism using Neural Networks. The main idea is to give less weight to the irrelevant images and more weight to the relevant images using user feedback.

For the relevance feedback we have used the Multilayer perceptron. The role of MLP here is to get good retrieval rate by updating its weights to the classes of relevant and irrelevant images which are marked by the user from the displayed similar images. So, by using these weights we have classified the similar images which we got initially in to the class of relevant and irrelevant images and we have displayed these relevant images. Still all these relevant images may not actually relevant to the query but the number of actual relevant images will increase compared to the relevant images in the similar images that we got initially. We can applay this method iteratively, in each iterarion user will mark the relevant and irrelevant images and MLP will adjust its weights according to the marked relevant and irrelevant images. In each iteration the number of relevant images will increase. The parameters to the MLP are as follows

In MLP the number of nodes in the input layer are 278 which is the length of the feature vector for each image and the input is the feature vector of the images. We have taken single hidden layer with 14 neurons and the number of nodes in the output layer is 1. Sigmoid function is taken as the activation function.

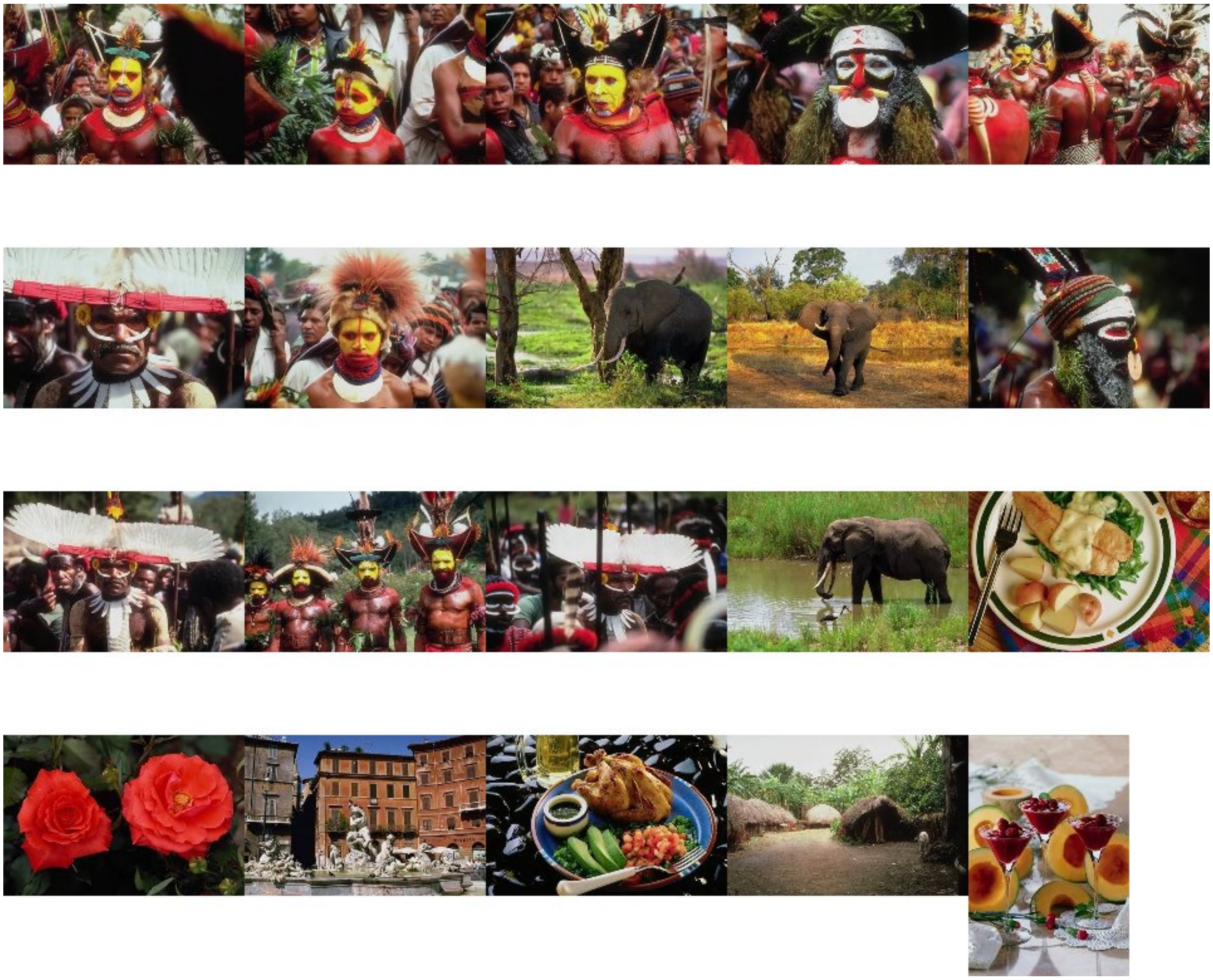
## 4.1 Results

First 20 Similar Images which we got by applying l1 distance measure to the histograms of query image to the histograms of images in the database. First image is the query image

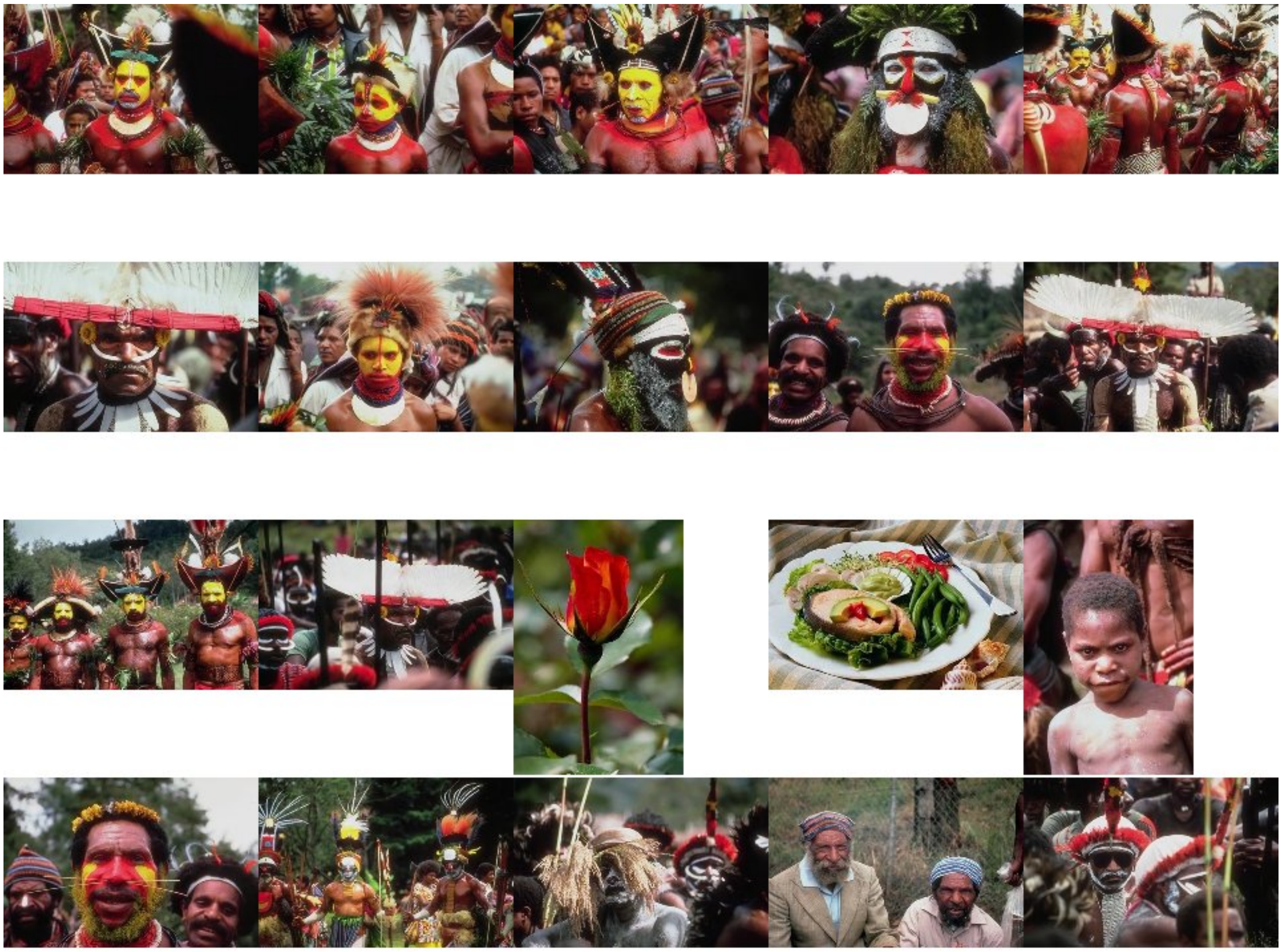


We have marked the images 0, 1, 2, 6, 7, 13, 19 as relevant and remaining images as irrelevant.

After applying the MLP the first 20 similar images are



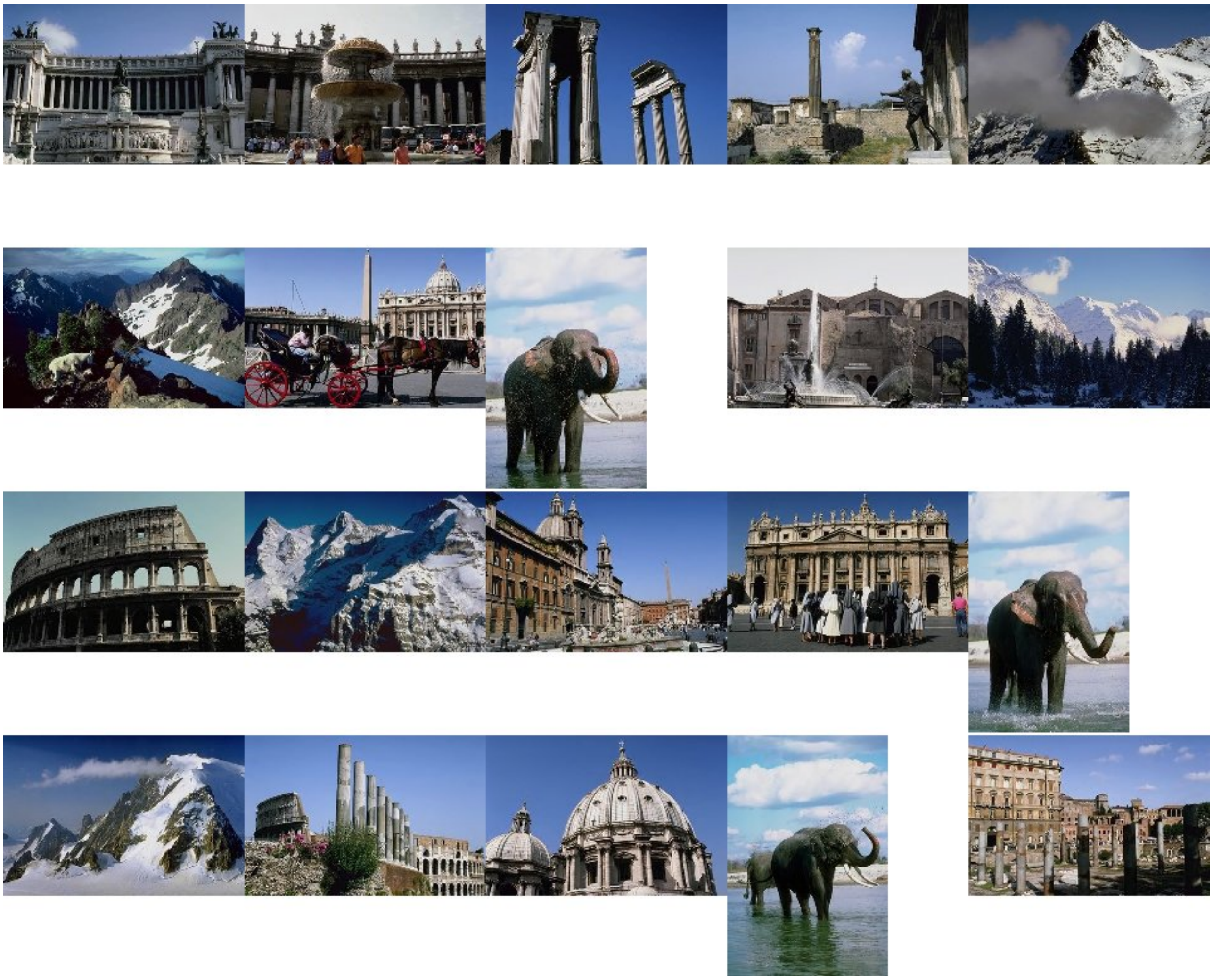
We have marked the images 0, 1, 2, 3, 4, 5, 6, 9, 10, 11, 12 as relevant and remaining images as irrelevant.
In the second iteration the first 20 similar images



We have marked the images 12, 13, 14, 18 as irrelevant and remaining images as relevant. In the third iteration the first 20 similar images

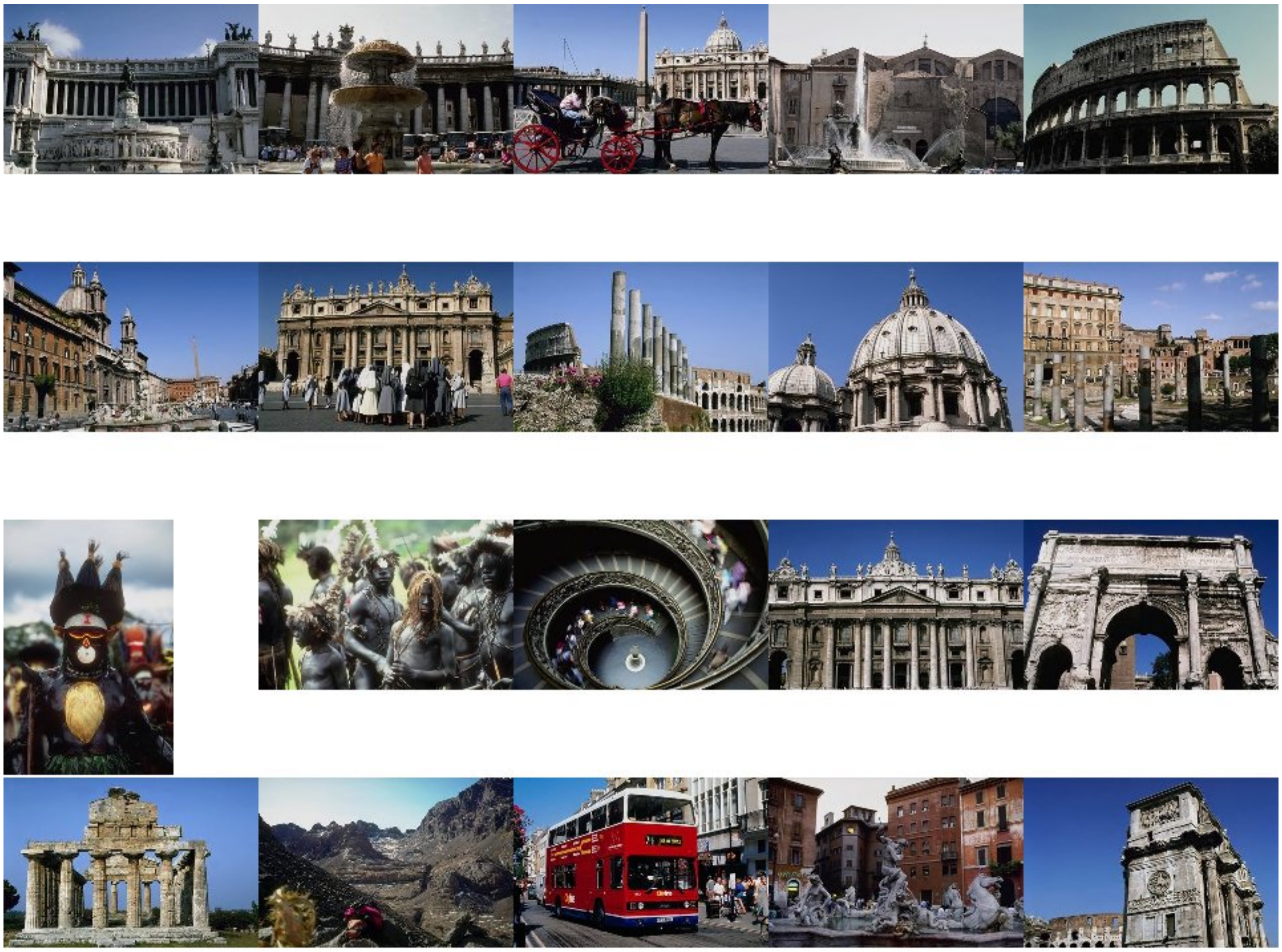
### Another Query

First 20 Similar Images which we got by applying l1 distance measure to the histograms of query image to the histograms of images in the database. First image is the query image

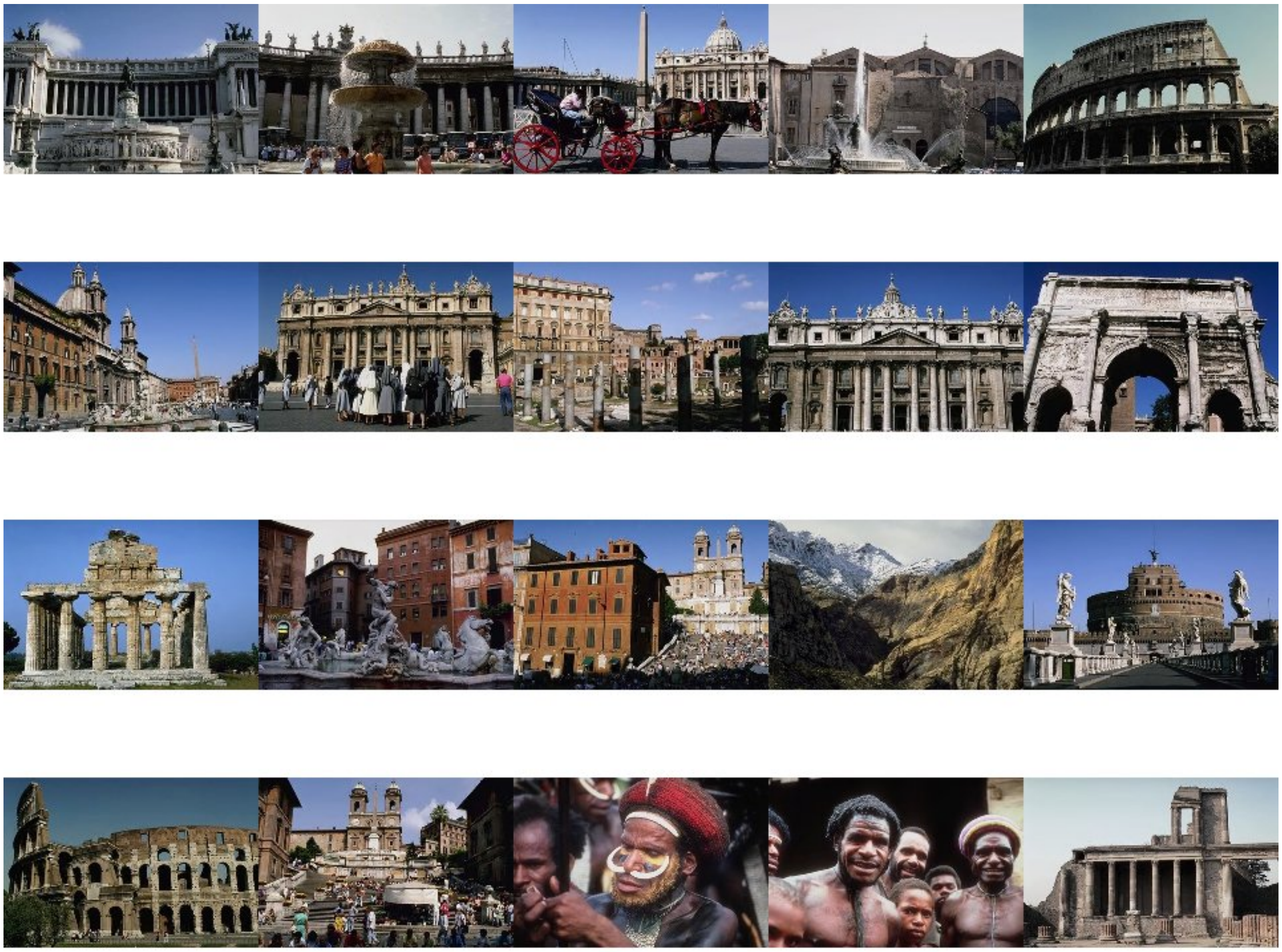


We have marked the images 0, 1, 6, 8, 10, 12, 13, 16, 17, 19 as relevant and remaining images as irrelevant.

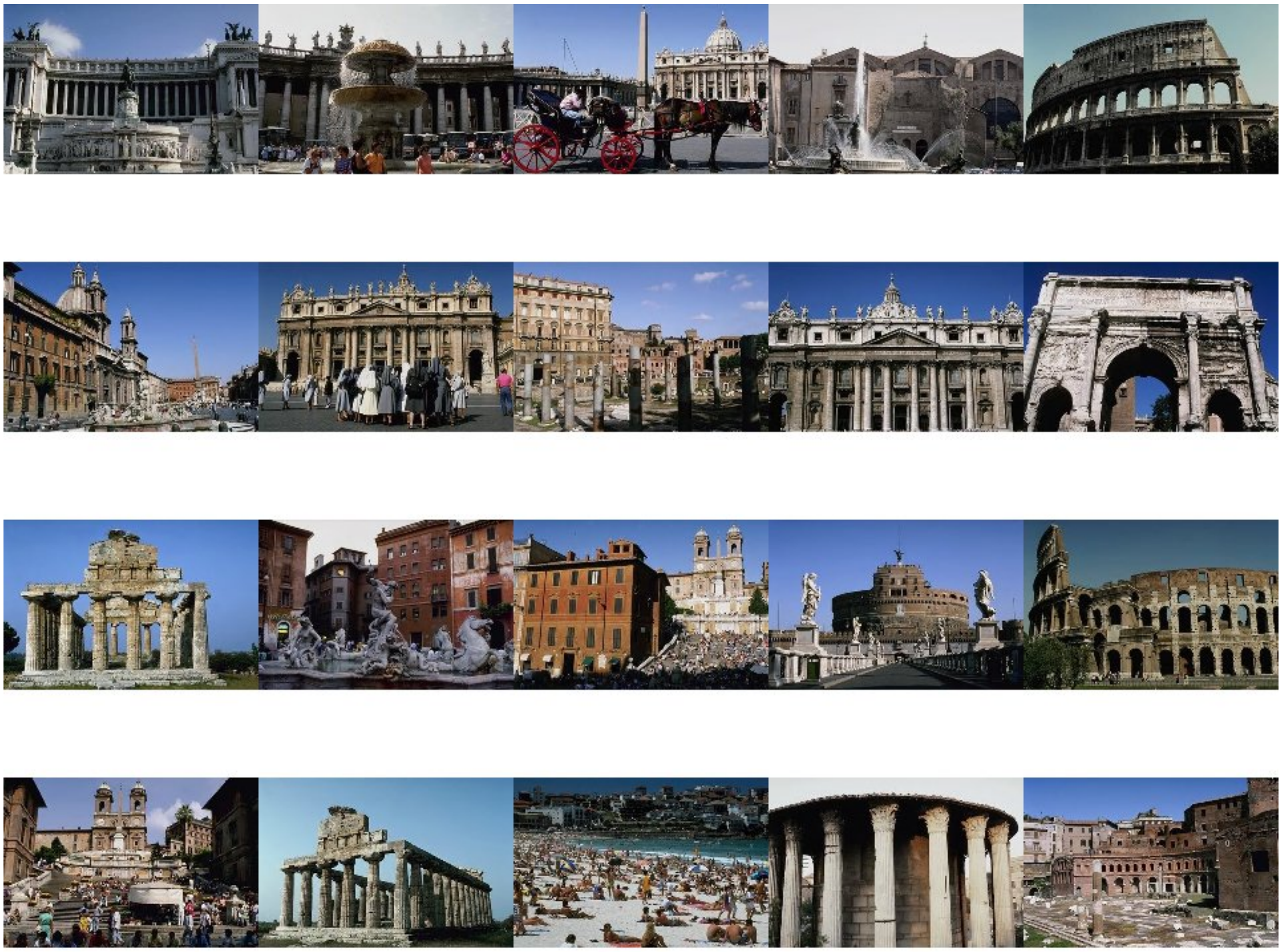
After applying the MLP the first 20 similar images are



We have marked the images 0, 1, 2, 3, 4, 5, 6, 9, 13, 14, 15, 18 as relevant and remaining images as irrelevant.
In the second iteration the first 20 similar images



We have marked the images 13, 17, 18, 19 as irrelevant and remaining images as relevant. In the third iteration the first 20 similar images



## Another Query

First 20 Similar Images which we got by applying l1 distance measure to the histograms of query image to the histograms of images in the database. First image is the query image


We have marked the images 0, 1, 2, 3, 4, 6, 8, 9, 13, 15, 16, 17 as relevant and remaining images as irrelevant.

After applying the MLP the first 20 similar images are


We have marked the images 15, 18, 19 as irrelevant and remaining images as relevant. In the second iteration the first 20 similar images

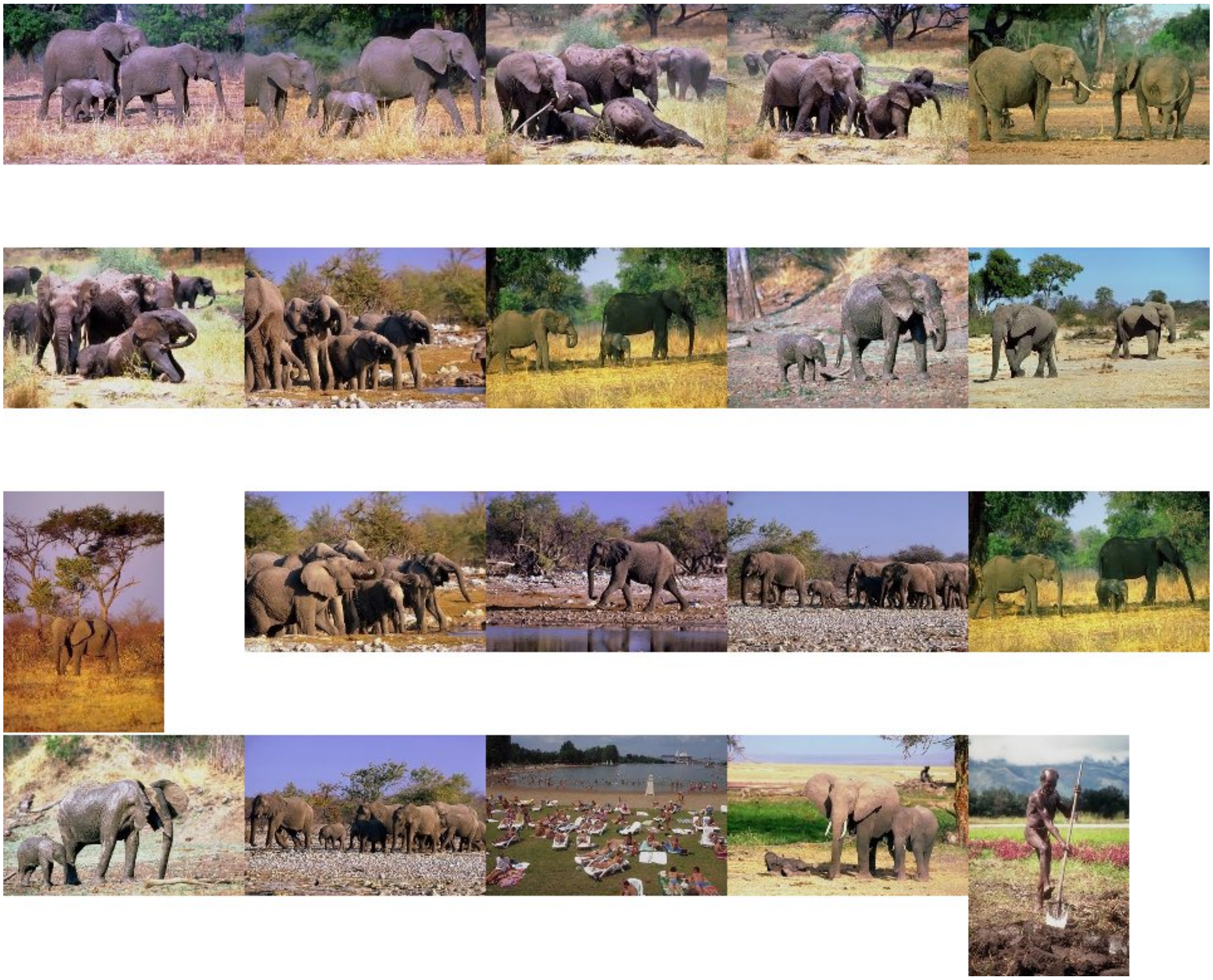
We have marked the images 17, 18, 19 as irrelevant and remaining images as relevant.
In the third iteration the first 20 similar images


### Another Query

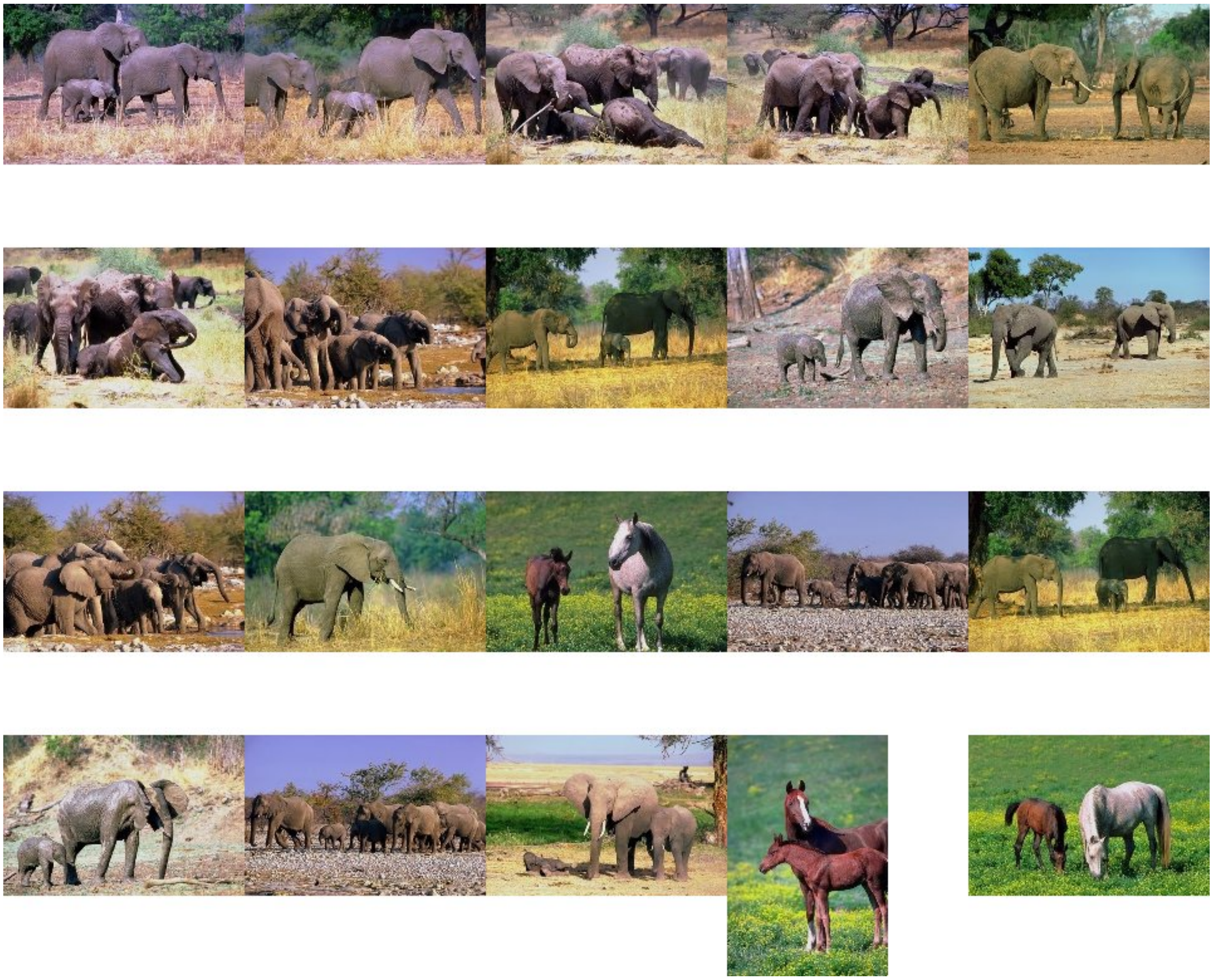
First 20 Similar Images which we got by applying l1 distance measure to the histograms of query image to the histograms of images in the database. First image is the query image

We have marked the images 0, 1, 3, 5, 8, 9, 12, 15, 17 as relevant and remaining images as irrelevant.

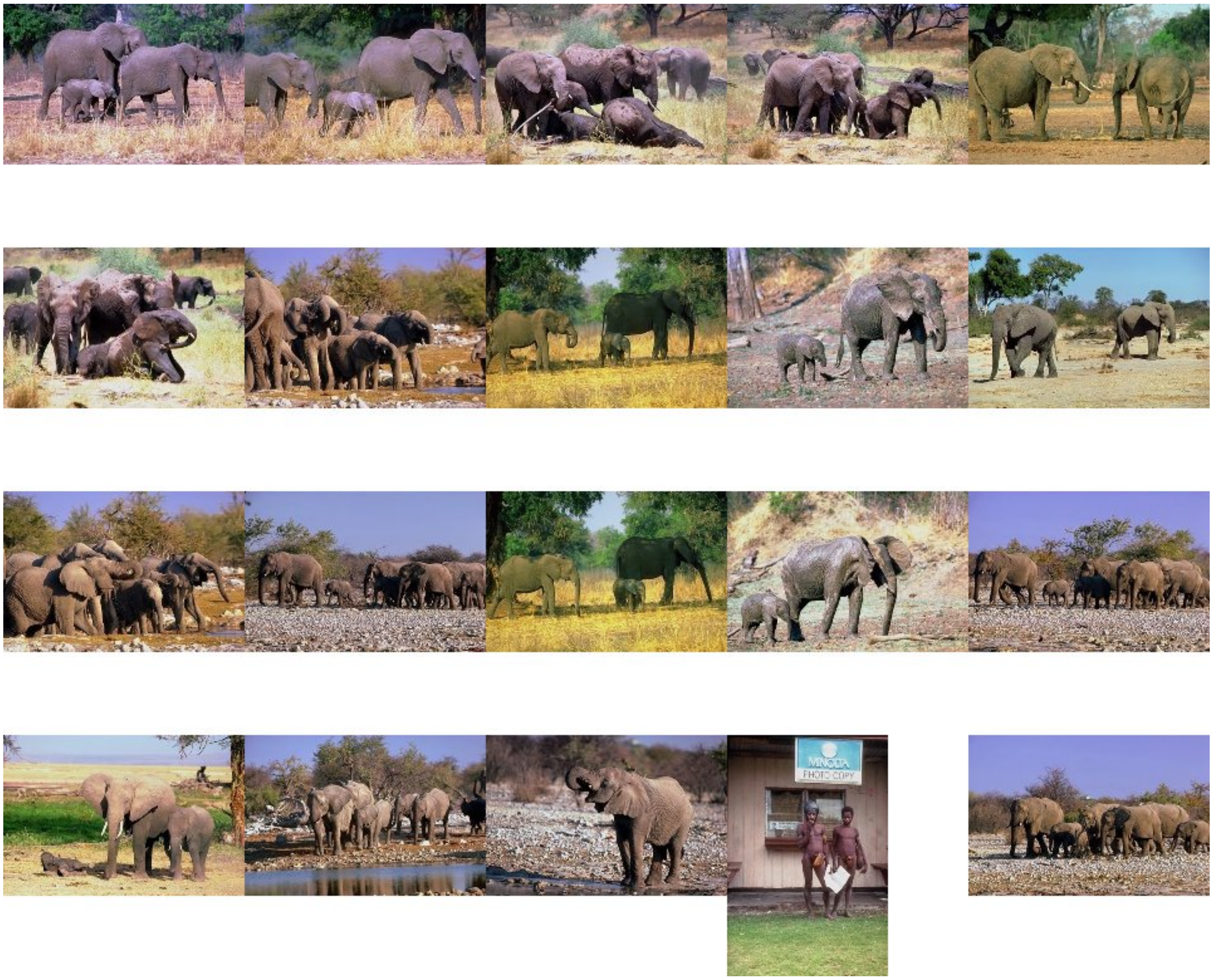
After applying the MLP the first 20 similar images are



We have marked the images 10, 12, 17, 19 as irrelevant and remaining images as relevant. In the second iteration the first 20 similar images

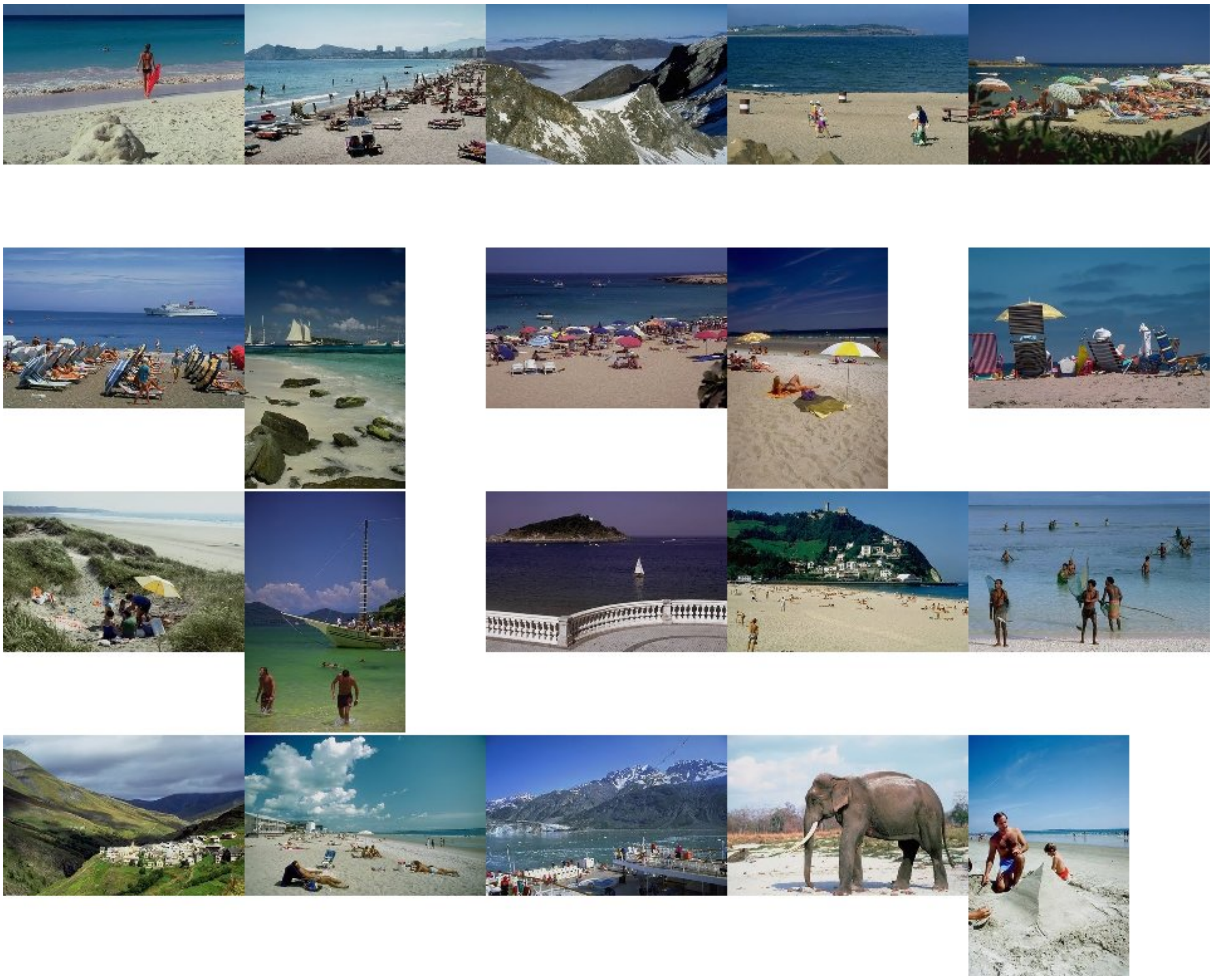


We have marked the images 11, 12, 18, 19 as irrelevant and remaining images as relevant. In the third iteration the first 20 similar images



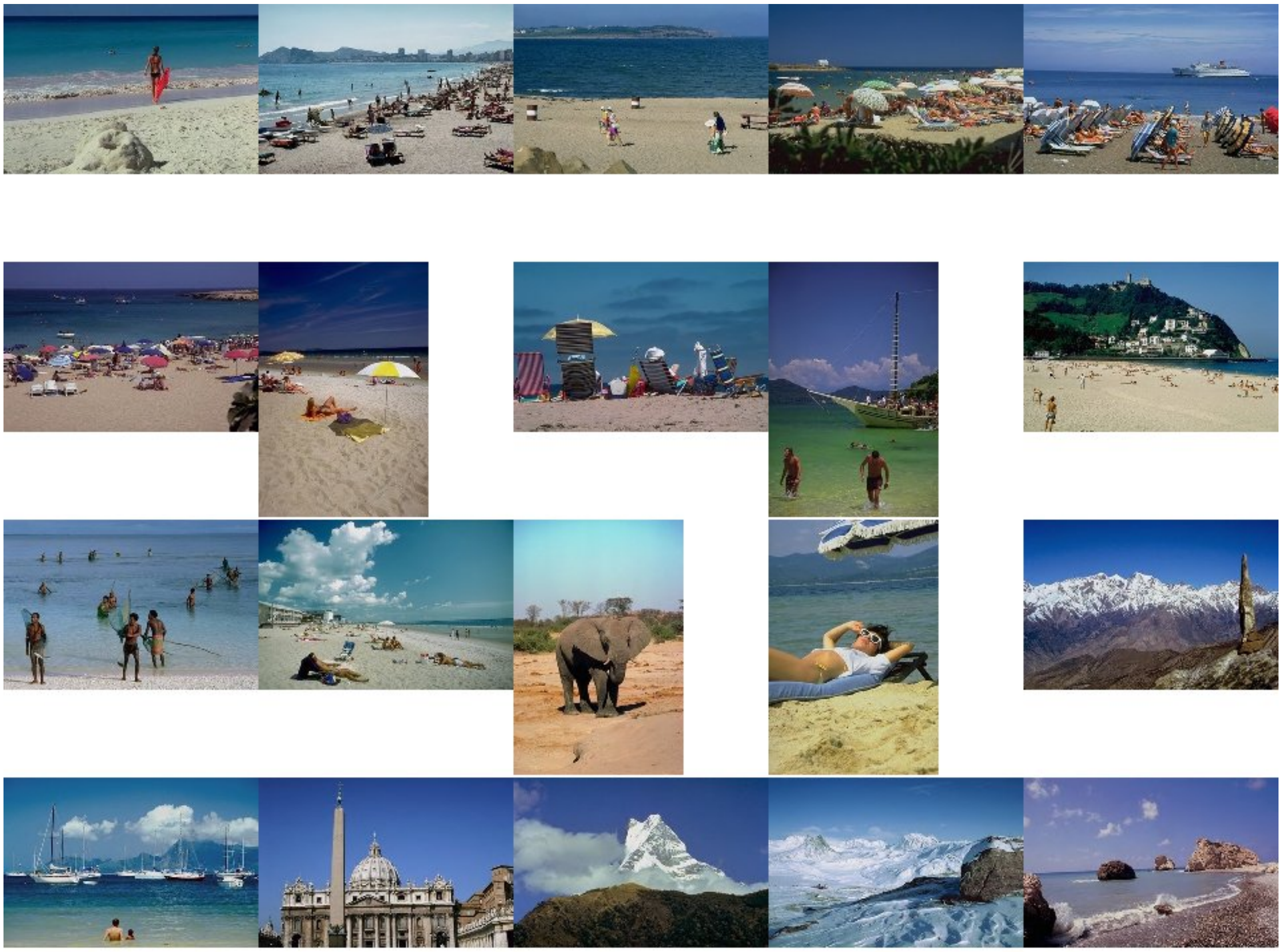
## Another Query

First 20 Similar Images which we got by applying l1 distance measure to the histograms of query image to the histograms of images in the database. First image is the query image



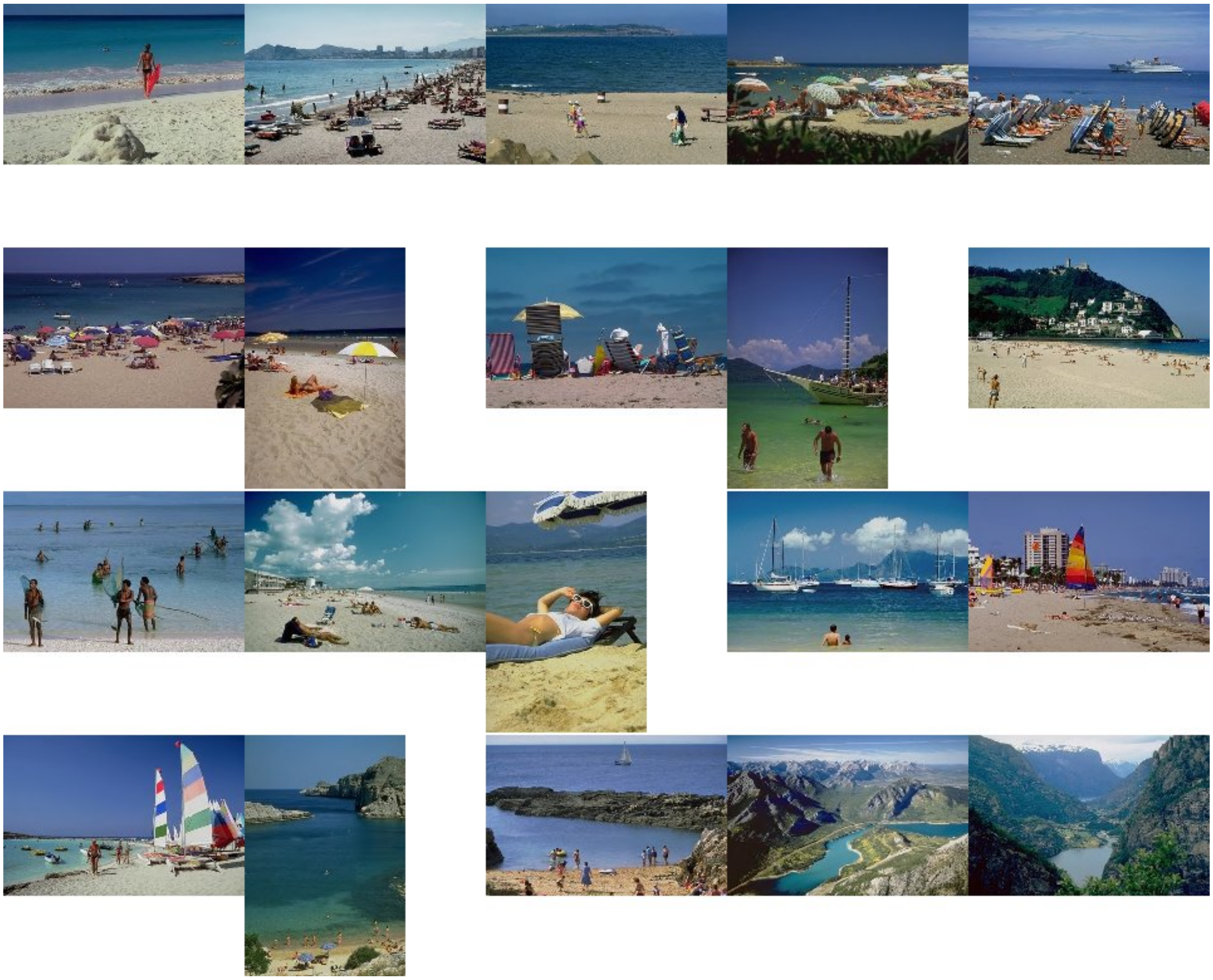
We have marked the images 0, 1, 3, 4, 5, 7, 8, 9, 11, 13, 14, 16 as relevant and remaining images as irrelevant.

After applying the MLP the first 20 similar images are



We have marked the images 12, 14, 16, 17, 18, 19 as irrelevant and remaining images as relevant.
In the second iteration the first 20 similar images

We have marked the images 12, 15, 17, 19 as irrelevant and remaining images as relevant. In the third iteration the first 20 similar images



## 4.2 Performance Measures

The following performance measures can be used to measure the performance of CBIR system

$$Precision = \frac{Numberofrelevantimagesretrieved}{Numberofimagesretrieved}$$

$$Recall = \frac{Numberofrelevantimagesretrieved}{Numberofrelevantimagesinthedatabase}$$

Recall-Precision graph is another important measure for performance.

# Chapter 5

# Conclusion and Future work

In this thesis we have presented an approch, which can efficiently retrieve the similar images to a query image. We have used color structure descriptor(CSD) and edge histogram descriptor(EHD) to extract the color, edge and texture features from the images and by using l1 distance measure we have calculated the distance between the query and other images. By using this distance value we have displayed the similar images. In the relevance feedback mechanism we have used the Multilayer perceptron and Radial basis function network.

# Chapter 6

# References

[1] M. K. Kundu, M. Banerjee and P. Bagrecha " Interactive Image Retrieval in a Fuzzy Framework", Proc. 8th International Workshop on Fuzzy Logic and Applications (WILF 2009) , Lecture Notes in Artificial Intelligence (LNAI 5571), pp.246 -253 , Palermo, Italy, June, 2009.

[2] M. K. Kundu and P. Bagrecha Color Image Retrieval Using MBand Wavelet Transform Based ColorTexture Feature ", Proc. 7th International Conf. on Advances in Pattern Recognition (ICAPR 2009) , pp. 117120, ISI, Kolkata, India, February, 2009.

[3] M. K. Kundu and M. Banerjee " Image Retrieval Using Fuzzy Relevance Feedback and Validation with MPEG7 Content Descriptors ", Proc. 2nd International Conference on Pattern recognition and Machine Intelligence(PREMI07), Lecture Notes in Computer Science series (LNCS 4815), pp. 144152, Kolkata, India, December, 2007.

[4] M. K. Kundu, M. Banerjee andP. Maji " Content based image retrieval using visually significant point features", Fuzzy Sets and Systems, 2009.

[5] B. S. Manjunath, Philippe Salembier and Thomas Sikora "Introduction to MPEG-7 Multimedia Content Description Interface"

[6]Yossi Rubner and Carlo Tomasi "Perceptual Metrics For Image Database Navigation"

[7] Alberto Del Bimbo "Visual Information Retrieval"

[8] Mark S. Nixon and Alberto S. Aguado "Feature Extraction and Image Processing"

[9]Ka-Man Wong, Lai-Man Po, and Kwok-Wai Cheung "DOMINANT COLOR STRUCTURE DESCRIPTOR FOR IMAGE RETRIEVAL"

[10]B.S Manjunath , J.R Ohm, V.V Vasudevan, A.Yamada "Color and Texture Descriptors" EEE Trans. on Circuits and Systems for Video Technology, vol. 11, no. 6, 703 -715,

June 2001"

[11]ISO/IEC/JTCI/SC29/WG11 : "Core Experiment Results for Edge Histogram Descriptor (CT4)," MPEG document M6174, Beijing, July 2000.

[12]A.K. Jain and A. Vailaya, "Image retrieval using color andshape," Pattern Recognition, Vol. 29, No. 8, pp. 1233-1244, 1996

[13]S. J. Park, D. K. Park, C. S. Won, " Core experiments on MPEG-7 edge histogram descriptor," MPEG document M5984, Geneva, May, 2000.

[14]Simon Haykin "Neural Networks - A Comprehensive Foundation"

[15]Ye Lu, Hongjiang Zhang, Liu Wenyin and Chunhui Hu "Joint Semantics and Feature Based Image Retrieval Using Relevance Feedback" IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 5, NO. 3, SEPTEMBER 2003.

[16]C. Buckley and G. Salton, Optimization of relevance feedback weights, presented at the SIGIR95.