Image Classification Based on Hybrid Compression System

Dr. Nidaa F. Hassan
Computer Science Department, University of Technology/Baghdad.
Email:nidaaalalousi_5@yahoo.com
Noor Emad A. lhamza
Computer Science Department, University of Technology/Baghdad
Email:nooremad8760 @yahoo.com

Received on: 30/6/2014 & Accepted on: 8/1/2015

ABSTRACT
Due to the fast development of internet technologies and multimedia archives are growing rapidly, especially digital image libraries which represent increasingly an important volume of information, it is judicious to develop powerful browsing computer systems to handle, index, classify and recognize images in database. In this paper a new algorithm image classification is proposed. This paper presents an efficient content-based image indexing technique for searching similar images using daubechies wavelet with discrete cosine transform. The aim of this work was to realize the image classification using hybrid compression system. The image was classified using 10 classes.

اقتراح خوارزمية لتصنيف الصور بناء على نظام المدمج لضغط الصورة

الخلاصة
بسبب التطور السريع لتكنولوجيا الانترنت وارشيف الوسائط المتعددة التي تنمو بسرعة ,خاصة مكتبات الصور الرقمية التي تمثل خزين هام للمعلومات،لذلك من الحكمة تطوير انظمة قوية للتصفح تقوم بمعالجة ،فهرسة ,تصنيف وتمييز الصورفي قاعدة البيانات.في هذا البحث تم اقتراح خوارزمية جديدة لتصنيف الصور. في هذه الورقة تقنية جديدة للفهرسة قدمت للبحث عن الصور المتشابهة معتمدة على موجات دوباجيز وتحويل المتقطع للجيب تمام. الهدف من العمل هو ادراك تصنيف الصور باستخدام نظام الضغط.الصورة تم تصنيفها باستخدام 10 اصناف.

Keywords: Image Classification, Compression, Image Features.

INTRODUCTION
The current improvement in the digital storage media, image capturing devices like scanners, web cameras, digital cameras and rapid development in internet provide a huge collection of images. This leads to the retrieval of these images for visual information efficiently and effectively in different fields of life like medical, medicine, art, architecture, education, crime preventions. To achieve this purpose many image retrieval systems have been developed [2]. A successful image classification will

significantly enhance the performance of the content-based image retrieval system by filtering out images from irrelevant classes during matching. The proposed algorithm improve classical technique by using hybrid compression (daubechies wavelet and discrete cosine), the result of hybrid compression shows a good deal of promise.

Image Classification

Image Classification refers to grouping of a digital image into different classes within a particular dataset, based on attribute values. It is done to replace visual analysis of the image data with quantitative techniques. The most important step in an image classification system is the image description. Indeed, features extraction gives a feature vector per image which is a reduced representation of the image visual content, because images are too big to be used directly for Indexing and retrieval [8].Intelligently classifying image by content is an important way to mine valuable information from large image collection. The classification procedures can be "supervised" or "unsupervised".

Supervised Classification

Supervised Classification can identify examples of the information classes of interest in the image. These are called "training sites”. Supervised classification categories the unknown pixels into different themes, based on the spectral (or statistical) characteristics of manual defined sampled pixels. After the classification process, all the unknown pixels with similar spectral characteristics of the defined categories are assigned and pixels which have their characteristics difference from the categories are categorized as unknown [5].

Unsupervised Classification

Unsupervised classification is a method which examines a large number of unknown pixels and divides into a number of classed based on natural groupings present in the image values [5].

Image Compression System

Image compression system is used to reduce the amount of data required to represent a given quantity of information, the size of image data files are reduced, while retaining necessary image information[2].In the image compression significance coding, unknown pixels is classified and coded based on the information provided by the known pixels [10].

Transform Coding Techniques

Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels consequently; these correlations can be exploited to predict the value of a pixel from its respective neighbors [2]. Compression schemes that operate in the transform domain, first transform the image using daubechies wavelet transform (DAWT), and then apply the discrete cosine transform (DCT) on low-low(LL) subband to obtain the optimal compressed image. For our work, compression system in the two most used transforms (the DCT and DAWT) is explained below.

Daubechies Wavelets Transform (DAWT)

Ingrid Daubechies, one of the brightest stars in the world of wavelet research, invented what are called compactly supported orthonormal wavelets — thus, making discrete wavelet analysis practicable. The names of the Daubechies family wavelets are written dbN, where N is the order, and db the "surname" of the wavelet [11].

Image Decomposition

The wavelet filters for sub-band decomposition derived from Daubechies wavelets are of non-linear phase, for this reason they are rarely used in image processing applications, such as denoising and compression. However, Daubechies wavelets can be derived from the mother wavelet high pass and low-pass filters in the dyadic sub-band image decomposition. The coefficients of the lowest frequency band are grouped in the upper left corner, meanwhile, the coefficients of the higher frequency bands are in the other three image corners, in order to obtain the information contained in the images, sub-level signal decompositions are performed to separate the signal characteristics and to analyze them independently [13].Figure (1) shows the transformation of the image into the sub-images. Approximation image is the sub-image compose of the low frequency parts in both row and column directions (LL), and details images are the remaining three images, containing high frequency components (LH, HL, and HH).

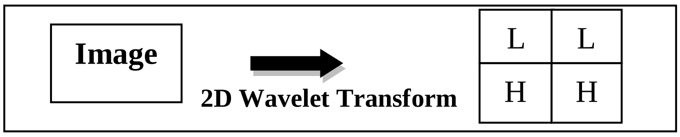


Figure (1): Transformation of The Image into Sub-Images [14].

- The scaling function coefficients are:

$$\boldsymbol{h0} = \mathbf{1} + \frac{\sqrt{3}}{4\sqrt{2}} \quad \text{... (1)}$$

$$\boldsymbol{h1} = \mathbf{3} + \frac{\sqrt{3}}{4\sqrt{2}} \quad \text{... (2)}$$

$$\boldsymbol{h2} = \mathbf{3} - \frac{\sqrt{3}}{4\sqrt{2}} \quad \text{... (3)}$$

$$\boldsymbol{h3} = \mathbf{1} - \frac{\sqrt{3}}{4\sqrt{2}} \quad \text{... (4)}$$

- The wavelet function coefficient values are [49]:

$$g0 = h3 \quad \text{...(5)}$$

$$g1 = -h2 \quad \text{....(6)}$$

$$g2 = h1 \quad \text{....(7)}$$

$$g3 = -h0 \quad \text{... (8)}$$

The sub band LL represents the approximation of the original signal, while the sub bands LH, HL, and HH represent the details [14].

Discrete Cosine Transform

Discrete cosine transform is a lossy compression algorithm that discards those frequencies which do not affect the image as the human eye perceives it [8]. Minimizes the amount of visible blocking artifacts compared to other transforms provides a good compromise or balance between information packing ability and computational complexity [1]. The two
dimensional DCT can be written in terms of pixel values f(i, j) for i,j = 0,1,…,N-1 and the frequency-domain transform coefficients F(u,v) [3]:

$$F(u,v) = I/\sqrt{2N}\quad C(u)\,C(v) \sum_{x}^{n-1}\sum_{y}^{n-1} Cos\left[\frac{\pi(2x+1)u}{2N}\right] Cos\left[\frac{\pi(2y+1)v}{2N}\right] \qquad \ldots(9)$$

For u = 0,1,2,…,N −1. Similarly, the inverse transformation is defined as

$$F(i,j) = \sum_{x}^{n-1}\sum_{y}^{n-1} C(u)\,C(v)F(u,v) = Cos\left[\frac{\pi(2x+1)u}{2N}\right] Cos\left[\frac{\pi(2y+1)v}{2N}\right] \qquad \ldots(10)$$

For x = 0,1,2,…,N −1. In both equations (1) and (2) C(x) is defined as

$$\mathbf{c}(x) = \begin{bmatrix} \sqrt{\frac{1}{N}} \text{ for } x = 0 \\ 1 \text{ for } x \neq 0 \end{bmatrix} \qquad \ldots.(11)$$

Image Feature Extraction

Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately [23].Every image is characterized by a set of features such as Texture, color, shape and others, extract these features at the time of injecting new image in image database. Then summarize these features in a reduced set of k indexes and store it in image feature database. The query image is processed in the same way as images in the database. Matching is carried out on the feature database [17]. Nine features are exploited for our proposed algorithm.

Color Feature

Color Feature is the most significant one in searching collections of color images of arbitrary subject matter. Color plays very important role in the human visual perception mechanism, besides that image color is easy-to analyze, and it is invariant with respect to the size of the image and orientation of objects on it. The simplest and most frequently used way to represent color is color histograms [17].

Mean

It tells something about the general brightness of the image, where a bright image has a high mean while a dark image has a low mean. It provides average color value in the image [19]. It is calculated using following statics:

$$\bar{g} = \sum_{g=0}^{L-1} g\,P(g) = \sum_r \sum_c \frac{I(r,c)}{M} \qquad \ldots.(12)$$

Where L is the gray level range such as [0, 1], [0 to 7] or [0 to 255], r is image row, c is image column and I(r, c) is the pixel at row r and column c.

Standard Deviation

Standard deviation is the square root of the variance of the distribution [19]. It is calculated using following statics:

$$\sigma = \sqrt{\sum_{g=0}^{L-1} (g-\bar{g})^2 \, P(g)} \qquad \dots (13)$$

Skewness

It gives measure of the degree of asymmetry in the distribution [19]. It is calculated using following statics:

$$Skew = \frac{1}{\sigma^3} \sum_{g=0}^{L-1} (g-\bar{g})^3 \, P(g) \qquad \dots (14)$$

Texture Feature

Textures are homogeneous patterns or spatial arrangements of pixels that cannot be sufficiently described by regional intensity or color features [18]. They are six textures.

Entropy

The entropy is a measure that tells how many bits are needed to code image data. Entropy is a measure of information content.
It measures the randomness of intensity distribution [16]. It can be calculated using the following equation:

$$Entropy = -\sum_{g=0}^{L-1} P(g) \log_2[P(g)] \qquad \dots (15)$$

Energy

The Energy measure tells something about how gray levels are distributed. Energy measures the uniformity of intensity in the histogram [32]. It is defined as follows:

$$Energy = \sum_{g=0}^{L-1} [P(g)]^2 \qquad \dots (16)$$

Homogeneity

Homogeneity returns a value that measures the closeness of the distribution of elements in image [20].Homogeneity takes high values for low-contrast images [24].

$$Homogeneity = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{p(i,j)}{|1+i-j|} \qquad \dots (17)$$

Variance

Variance in the gray level in a region in the neighborhood of a pixel is a measure of the texture [25].

$$\boldsymbol{Var} = \frac{1}{w^2 \sum_{i=0} \sum_{j=0} img(i,j) - mean(k))2} \qquad \dots.(18)$$

3rd Moment

3 **rd** moment measures the skewness of a histogram [33]. 3rd Moment measures distortion of gray level.

$$\boldsymbol{3^{rd}\ Moment} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left((i-j)\right)^3 p(i.j) \qquad \dots(19)$$

Step

Step measures distribution of gray level [33].

$$\boldsymbol{Step} = \sum_{i=0}^{m-1} \sum_{n=0}^{n-1} p(i,j) \qquad \dots(20)$$

Proposed Hybrid DAWT-DCT Based Algorithm

In this algorithm, hybrid compression is performed on query image and database images. First compression method is 2D DAWT followed by DCT. In DCT image, data are divided up into 8*8 number of block.

DCT converts the spatial image representation into a frequency map: the average value in the block is represented by the low-order term, strength and more rapid changes across the width or height of the block represented by high order terms. The DCT is applied to the DAWT low-frequency components that generally have zero mean and small variance, and accordingly results in much higher compression ratio (CR) with important diagnostic information. Figure (2) shows the flowchart of Hybrid DAWT-DCT algorithm, which is also illustrated in algorithms (1), (2) and (3).

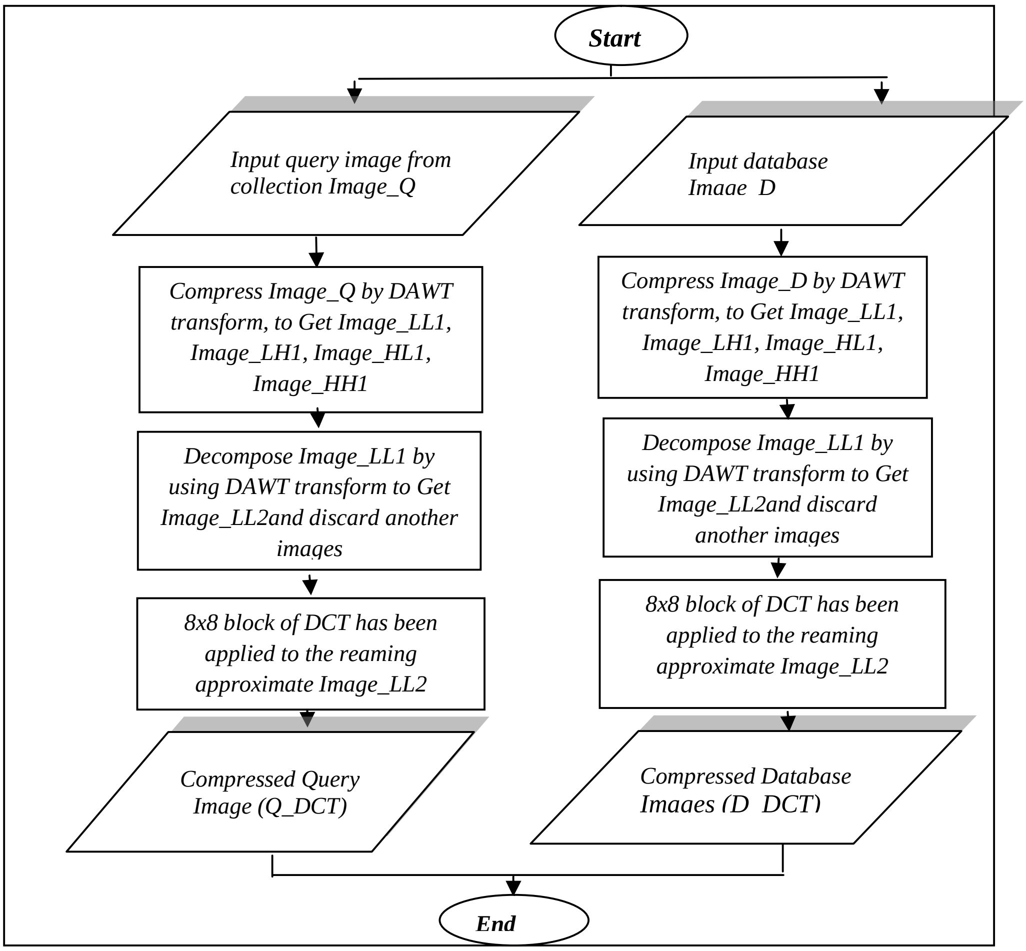


Figure (2): The Flowchart of Hybrid DAWT-DCT Compression Algorithm.

First Level of DAWT Decomposing

2-D DWT decomposed each 2x2 blocks of the Image_q a
nd Image_d to be classified, the low-frequency coefficients are LL and high-frequency coefficients are HL, LH and HH . Approximation image is the sub-image composes of the low frequency parts in both row and column directions (LL). By convolution of the Image_q and Image_d with a filter bank (-0.48296,-0.8365,-0.2241,0.1294). Perform the average of four pixels then divide the result on 4 .

New value is saved in an array called LL_array ( ), to be used in the next step. Like LL Decomposes, high-frequency coefficients HL is obtained by convolution of the Image_q, Image_d with same filter bank (-0.48296 ,-0.8365,-0.2241,0.1294). applying the average of four pixels then divide result on 4.a new value is saved in an array called HL_array ( ), to be discarded. High-frequency coefficients HL is obtained by convolution of the Image_q, Image_d with same filter bank (0.1294, -0.2241, -0.8365, 0.48296). performing the average of four pixels then divide result on 4 .a new value is saved in an array called

LH_array ( ), to be discarded. High-frequency coefficients HH is obtained by convolution of the Image_q, Image_d with same filter bank (0.1294, -0.2241, -0.8365 , 0.48296). performing the average of four pixels then divide result on 4 .a new value is saved in an array called HH_array ( ), to be discarded. The following algorithm (1) illustrates these steps.

Algorithm (1): First Level of DAWT Decomposing.
Input: The query image (Image_Q)
Output: The first decomposed images (Image _LL1), (Image_LH1), (Image_HL1) and (Image_HH1) level.
Step 1: Initialize j=0,i=0
Step 2: For x = 0 To Image_Q .Width Step 2 do
For y = 0 To Image_Q .Height Step 2 do
Get X1 'get pixel (x,y) from Image_Q and multiply X1 by Lowpass coefficient filter X5 = (-0.48296 * X1)
Get X2 'get pixel (x,y) from Image_Q and multiply X2 by Lowpass coefficient filter X6 = (-0.8365 * X2)
Get X3 ' get pixel (x,y) from Image_Q and multiply X3 by Lowpass coefficient filter X7 = (-0.2241 * X3)
Get X4 ' get pixel (x,y) from Image_Q and multiply X4 by Lowpass coefficient filter
X8 = (0.1294 * X4)
Find the average of four pixels
LL1(i, j) = ((X5 + X6 + X7 + X8) / 4) ' LL1-array ( ) represents lowlow frequency image to use it in second level decomposition then display LL1_array ( ) in Image_LL1.
End For
End For
Step 3: For x = 0 To Image_Q .Width Step 2 do
For y = 0 To Image_Q .Height Step 2 do
Get X2 ' get pixel (x,y) from Image_Q and multiply X2 by Lowpass coefficient filter
X10 = (-0.8365 * X2)
Get X3 ' get pixel (x,y) from Image_Q and multiply X3 by Lowpass coefficient filter

X11 = (-0.2241 * X3)
Get X4 ' get pixel (x,y) from Image_Q and multiply X4 by Lowpass coefficient filter
X8 = (0.1294 * X4)
Get X1 'get pixel (x,y) from Image_Q and multiply X1 by Lowpass coefficient filter
X9 = (-0.48296 * X1)
Find the average of four pixels
HL1(i, j) = ((X9 + X10) - (X11 + X12)) / 4
' HL1_array ( ) represents highlow subband which discard in second level of decomposing then display HL1_array ( ) in Image_HL1

Second Level of DAWT Decomposing

The passed **LL** components are further decomposed using another 2-D **DAWT** with same filter bank and the detail coefficients (**HL, LH and HH**) are discarded. The following algorithm (2) illustrates these steps.

Algorithm (2): Second Level Decomposing of DAWT **.**
Input: The low frequency subband image (Image _LL1) .
Output: The lowlow frequency image (Image _LL2).
Step 1: Initialize j=0,i=0.
Step 2: For x = 0 To Image_ LL1 .Width Step 2 do
For y = 0 To Image_ LL1 .Height Step 2 do
' Convolution window of 2*2 from Image_ LL1.
X5 = (-0.48296 * X1)
Get X2 ' get pixel (x,y) from Image _LL1 and multiply X2 by Lowpass coefficient filter
X6 = (-0.8365 * X2)
Get X3 ' get pixel (x,y) from Image _LL1 and multiply X3 by Lowpass coefficient filter
X7 = (-0.2241 * X3)
Get X4 ' get pixel (x,y) from Image _LL1 and multiply X4 by Lowpass coefficient filter
X8 = (0.1294 * X4)
Find the average of four pixels
LL2(i, j) = ((x5 + x6 + x7 + x8) / 4
' LL2_array ( ) represent lowlow frequency image then display LL2_array ( ) in Image_LL2.
End For
End For

Hybrid DAWT-DCT Compression

The presented hybrid **DWT-DCT** algorithm for image compression is to exploit the properties of both the **DAWT** and the **DCT**. After **LL** decomposing, 8x8 block of **DCT** has been applied to the reaming approximate **DAWT** coefficients (**LL**) and can be achieved high CR. **DCT** converts the spatial image representation into a frequency map. The average value in the block is represented by the low-order term, strength and more rapid changes across the width or height of the block represented by high order terms. The Discrete Cosine Transform (DCT) has been shown to be near optimal for a large class of images in energy concentration and decorrelation. The following algorithm (3) illustrates these steps.

Algorithm (3): Hybird DAWT_DCT Compression.
Input: lowlow frequency subband image (Image_LL2).
Output: The compressed image (CM_ Image).
Step 1: Initialize f=1,d=0, pi = 3.14159265358932.
Step 2: For X = 1 To Image_LL2.Width - 8 Step 8 do
For Y = 1 To Image_LL2.Height - 8 Step 8 do
Get 8*8 block of pixels from Image_LL2
P_array(f) = Image_LL2(x,y)
F = F+1 'Increment f value to get another pixels from Image_LL2.
Step 3: For r = 0 To f do
Step 4: Perform discrete cosine equations:
transf(0) = 1 / sqrt(f) * transf(0)
Step 5: For u = 1 To f do
For k = 0 To f do
transf(u) = transf(u) + p(k) * cos (2 * k + 1) * (u * pi)) / (2 * f)))
End For
transf(u) =transf(u) * sqrt(2) / sqrt(f)))
End For
Step 6: For l = 0 To f do
' Transfer values of discrete cosine for one block to bigcolor_array( ) and use transf_array( ) for another block of original image.
bigcolor(d) = transf(l): D = D + 1
transf(l) = 0: P(l) = 0
End For
F=0
End For
Step 7: Display bigcolor_array( ) values in CM_Image **.**

Features Based Algorithm

Feature Extraction is a process that begins with feature selection. In this Algorithm, the image classification process deals with the compressed image from previous section to generate feature vectors .Training the classification systems with these features incorporated could increase accuracy rate. The extracted features include: mean,standarddeviation,entropy,energy,homogeneity,step,3rdMoment,skewness,variance .After trying a number of features for query and database images , these nine features are chosen, feature extraction is divided into following steps:

1.Compute probability of each pixel in image and save it in prob_array .after that we are calculating **mean** feature. Each element of prob _array is saved in variable called pro then make prob_array =0 to be used for another iteration. pro is divided on size of image (width and height) are called **s** and **q**.
2. Extract standard deviation feature.
3. Extract the entropy feature.
4. Extract the Energy feature
5. Extract the homogeneity feature.
6. Extract the step feature.
7. Extract the variance feature.
8. Extract the 3rd moment feature.
9. Extract the skewness feature .The following algorithm (4) illustrates these steps.

Algorithm (4): Image Feature Extraction.
Input: The compressed image (CM_Image).
Output: feature vector (FV_query).
Step 1: Initialize i=0.
Step 2: Compute probability of each pixel in image using the following equations:

```
For f = 1 To CM_Image.Width – 1 do
    For g = 1 To CM_Image.Height – 1 do
        pro  = prob(i)
        pro  = pro / (s * q)
    End For
End For
```

Step 3: Extract mean feature

```
V = pro * i
Mean = Mean + V
Textbox1 = (10000 * Mean) / 10000
```

Step 4: Manipulating of standard deviation feature is accomplished by using The following equations:

```
StdDev = StdDev + ((i - Mean) ^ 2 * pro
Std = Sqrt(StdDev)
TxtStdDev = (10000 * Std) / 10000
```

Step 5: The entropy feature can be extracted by performing following equations:
Step 7: The homogeneity feature can be extracted by applying the following equations:

```
V = pro / (1 +Abs(i))
Homo = Homo + V
Textbox3 = (10000 * Homo) / 10000
Textbox4 = (10000 * Distr) / 10000
```

Step 8: The step feature can be extracted by performing the following equations:

```
V = pro
Distr = Distr + D
```

Step 9: The variance feature can be extracted by performing the following equations:

```
V = pro * (i * i)
Var = Var + V
Textbox5 = (10000 * Var) / 10000
```

Step 10: The 3rd moment feature can be extracted by applying following equations:

```
V = pro * (i * i * i)
Disto = Disto + V
TextBox6 = (10000 * Disto) / 10000
```

Step 11: The skewness feature is extracted by performing the following equations:

```
Skew = Skew + ((i - mean) ^ 3 * pro)
Skew1 = (1 / std) * Skew
TextBox7 = (10000 * Skew1) / 10000
```

Algorithm (4): Image Feature Extraction.
Input: The compressed image (CM_Image).
Output: feature vector (FV_query).
Step 1: Initialize i=0.
Step 2: Compute probability of each pixel in image using the following equations:

```
For f = 1 To CM_Image.Width – 1 do
For g = 1 To CM_Image.Height – 1 do
pro   =  prob(i)
pro    =  pro / (s * q)
End For
End For
```

Step 3: Extract mean feature

```
V   =   pro * i
Mean    =  Mean + V
Textbox1 =  (10000 * Mean) / 10000
```

Step 4: Manipulating of standard deviation feature is accomplished by using The following equations:

```
StdDev   = StdDev + ((i - Mean) ^ 2 * pro
Std   = Sqrt(StdDev)
TxtStdDev = (10000 * Std) / 10000
```

Step 5: The entropy feature can be extracted by performing following equations:

```
V   =   pro * Log(pro) * Log(2)
Entropy   =  Entropy + V
Textbox3 =  (10000 * Entropy) / 10000
```

Step 6: The energy feature is extracted by using the following equations:

```
V  =    pro * pro
Energy    =   Energy + V
Textbox2  =  (10000 * Energy) / 10000
```

Step 7: The homogeneity feature can be extracted by applying the following equations:

```
V  =   pro / (1 +Abs(i))
Homo   =   Homo + V
Textbox3  =   (10000 * Homo) / 10000
Textbox4 =  (10000 * Distr) / 10000
```

Step 8: The step feature can be extracted by performing the following equations:

```
V  =   pro
Distr    =   Distr + D
```

Step 9: The variance feature can be extracted by performing the following equations:

```
V  = pro * (i * i)
Var  = Var + V
Textbox5 = (10000 * Var) / 10000
```

Step 10: The 3rd moment feature can be extracted by applying following equations:

```
V  =  pro * (i * i * i)
Disto  =  Disto + V
TextBox6  = (10000 * Disto) / 10000
```

Step 11: The skewness feature is extracted by performing the following equations:

```
Skew  =  Skew + ((i - mean) ^ 3 * pro)
Skew1   =  (1 / std) * Skew
TextBox7    = (10000 * Skew1) / 10000
```

Sum-of-Absolute Differences (SAD)

The similarity measurement is done using Sum-of-Absolute Differences distance. Then the top closest images are retrieved. When Δ D is the distance between the feature vector Q_f and D_f and N represent the number of color and texture feature.

Δ D is calculated using the following equation [21]:

$$\Delta D\,(\mathrm{Qf}, \mathrm{Df}) = \sum_{i=0}^{n-1} \mathrm{Qf}\,(\mathrm{i}) - \mathrm{Df}\,(i) \qquad ..(21)$$

Subsequently, the Δ D is stored in the ascending order then the selected images are retrieved. Both query and database images are similar for Δ D =0 and the small value of Δ D shows the relevant image to the query image [21].

Results and Discussion

DCT and Daubechies Transform algorithms are implemented on 100 color images. Size of the images is 256*256pixels. These images are arranged in 10 semantic groups: People, Beaches, Building, Buses, Dinosaurs, Elephants, Roses, Horses, Mountains and Food. It includes 100 images from each semantic group. The images are in jpeg format. The algorithm is executed with 10 query images selected for each category in the image database. The results obtained are shown in the following tables. Table (1) shows precision and recall for performing daubechies wavelet transform. Table (2) shows precision and recall for performing Discrete Cosine Transform. Table (3) shows precision and recall for hybrid system of Daubechies Wavelet Transform and Discrete Cosine Transform.

Table (1)

Class number	Categories	precision	recall
1	People	0.20	0.02
2	Beaches	0.20	0.02
3	Buildings	0.40	0.04
4	Buses	0.30	0.03
5	Dinosaurs	0.90	0.09
6	Elephants	0.30	0.03
7	Roses	0.60	0.06
8	Horses	0.40	0.04
9	Mountains	0.20	0.02
10	Foods	0.10	0.01

Table(2)

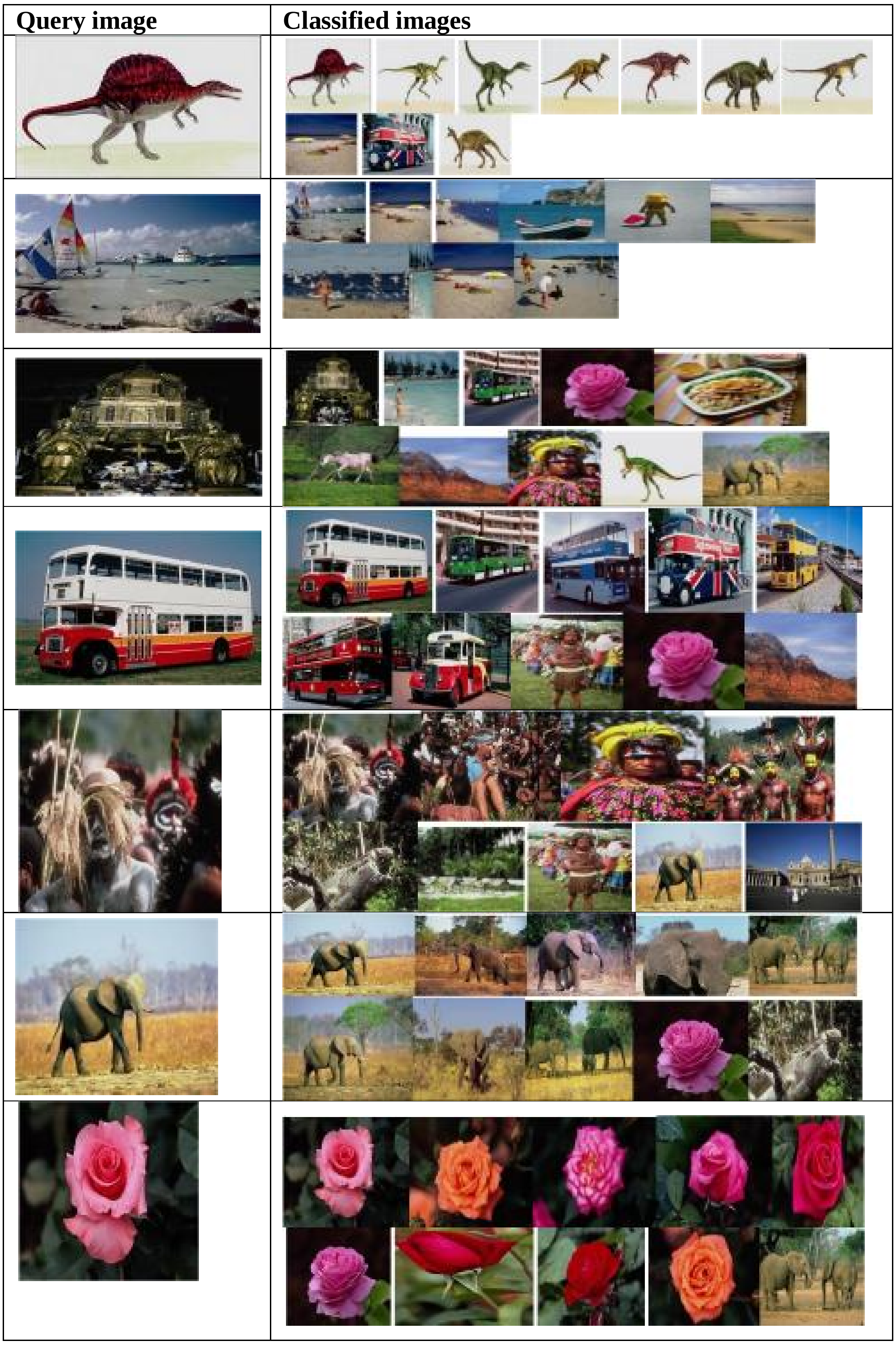
Class number	Categories	precision	recall
1	People	0.40	0.04
2	Beaches	0.20	0.02
3	Buildings	0.40	0.04
4	Buses	0.40	0.04
5	Dinosaurs	0.90	0.09
6	Elephants	0.20	0.02
7	Roses	0.30	0.03
8	Horses	0.50	0.05
9	Mountains	0.20	0.02
10	Foods	0.10	0.01

Table(3)

Class number	Categories	precision	recall
1	People	0.70	0.07
2	Beaches	1	0.10
3	Buildings	0.10	0.01
4	Buses	0.70	0.07
5	Dinosaurs	0.80	0.08
6	Elephants	0.80	0.08
7	Roses	0.90	0.09
8	Horses	0.70	0.07
9	Mountains	0.90	0.09
10	Foods	0.80	0.08

CONCLUSIONS

Classification is the process of finding a model or a function that describes and distinguishes data classes. In this paper, a new algorithm is proposed to classify images; the basic idea depends on using hybrid compression method (DAWT and DCT). To evaluate this algorithm, a heterogeneous image database is used which is downloaded from the website http://wang.ist.psu.edu/iwang/test1.tar. Hybrid compression produced better result (precision and recall values are improved 80%) in compare with using DAWT or DCT alone and are improved 90% in compared with proposed method which is presented in[22].the proposed compression produced better result(since precision and recall values are improved 80%)in compared with proposed method which is also presented in [23]. Figure (3) shows the classification of images using proposed compression algorithm. Table (4) shows the values of images features that are saved in database.


Query image
Classified images

Figure(3) : The classification of images using proposed compression algorithm.

Table (4): The values of Images Database Features for People, beaches, buildings, buses, dinosaurs, elephant, roses ,horses, mountains, foods Category.

Image number	*Mean*	*Standard Deviation*	*Entropy*	*Energy*
Image_1	*44.48*	*94.08*	*0.56*	*0.5965*
Image_2	*42.54*	*92.80*	*0.56*	*0.6065*
Image_3	*43.38*	*93.37*	*0.57*	*0.602*
Image_4	*43.35*	*93.24*	*0.56*	*0.6027*
Image_5	*43.3*	*93.39*	*0.57*	*0.602*
Image_6	*43.38*	*93.39*	*0.56*	*0.6019*

Image_7	*43.33*	*93.34*	*0.57*	*0.6024*
Image_8	*43.30*	*93.35*	*0.56*	*0.6028*
Image_9	*43.29*	*93.25*	*0.56*	*0.6028*
Image_10	*43.14*	*93.02*	*0.56*	*0.6043*

Image number	***mean***	***Standard Deviation***	***entropy***	***variance***	***3rd moment***
Image_1	*43.1996*	*93.1829*	*0.5627*	*10911*	*2756267*
Image_2	*40.3539*	*90.8815*	*0.5527*	*10181*	*2569194*
Image_3	*43.4536*	*93.456*	*0.5696*	*10975*	*2772379*
Image_4	*42.6145*	*92.6438*	*0.5625*	*10757*	*2715831*
Image_5	*42.6288*	*92.8084*	*0.5691*	*10764*	*2718604*
Image_6	*42.6724*	*92.7629*	*0.5698*	*10774*	*2720511*
Image_7	*42.7244*	*92.7146*	*0.5626*	*10787*	*2723858*
Image_8	*42.8336*	*92.7799*	*0.5532*	*10829*	*2738152*
Image_9	*42.6738*	*92.7334*	*0.5676*	*10774*	*27206987*
Image_10	*42.6928*	*92.6916*	*0.5593*	*10784*	*2724260*

Image number	***Mean***	***Standard Deviation***	***Entropy***
Image_1	*46.07*	*95.73*	*0.5636*
Image_2	*44.428*	*94.72*	*0.5617*
Image_3	*44.7588*	*94.8742*	*0.5542*
Image_4	*44.7858*	*94.9751*	*0.5652*
Image_5	*44.7804*	*94.9698*	*0.5629*
Image_6	*44.81*	*94.94*	*0.56*
Image_7	*44.8134*	*94.8542*	*0.5518*
Image_8	*44.8891*	*95.0928*	*0.5662*
Image_9	*44.9312*	*95.1089*	*0.5666*
Image_10	*44.6819*	*94.882*	*0.5622*

Variance	*Step*	*Homogeneity*
11645	*0.9439*	*0.7623*
11223	*0.9439*	*0.7687*
11313	*0.9439*	*0.7675*
11314	*0.9439*	*0.7673*
11311	*0.9439*	*0.7673*
11324	*0.9439*	*0.7673*
11327	*0.9439*	*0.7673*
11340	*0.9439*	*0.7669*
11348	*0.9439*	*0.7667*
11287	*0.9439*	*0.7677*

Homogeneity	*Step*	*Variance*	*3rd Moment*	*Skewness*
0.7685	*0.9439*	*11241*	*2840596*	*21494*
0.7759	*0.9439*	*10737*	*2709732*	*21310*
0.7727	*0.9439*	*10955*	*2766436*	*21407*
0.7729	*0.9439*	*10951*	*2766418*	*21362*
0.7727	*0.9439*	*10952*	*2765166*	*21399*
0.7727	*0.9439*	*10955*	*2766554*	*21414*
0.7729	*0.9439*	*10940*	*2762217*	*21385*
0.7731	*0.9439*	*10938*	*2763357*	*21384*
0.7731	*0.9439*	*10932*	*2761024*	*21377*
0.7737	*0.9439*	*10895*	*2751924*	*21269*

homogeneity	*step*	*variance*	*3^{rd} moment*	*skew value*
0.7803	*0.9439*	*10438*	*2632582*	*21141*
0.7814	*0.9439*	*10371*	*2615096*	*21086*
0.7834	*0.9439*	*10232*	*2578910*	*21033*
0.7834	*0.9439*	*10242*	*2582601*	*21028*
0.7832	*0.9439*	*10251*	*2584431*	*21041*
0.7832	*0.9439*	*10254*	*2585311*	*21067*
0.7834	*0.9439*	*10237*	*2580676*	*21014*
0.7832	*0.9439*	*10252*	*2584525*	*21025*
0.7832	*0.9439*	*10251*	*2584230*	*21043*
0.7832	*0.9439*	*10255*	*2585922*	*21048*

Image number	*mean*	*standard deviation*	*entropy*	*energy*
Image_1	*41.3967*	*91.7218*	*0.5706*	*0.6128*
Image_2	*41.1353*	*91.4941*	*0.5665*	*0.6144*
Image_3	*40.6066*	*91.1871*	*0.5707*	*0.6171*
Image_4	*40.6258*	*91.043*	*0.5668*	*0.6173*
Image_5	*40.6694*	*91.096*	*0.5674*	*0.6169*
Image_6	*40.674*	*91.201*	*0.5694*	*0.6167*
Image_7	*40.6159*	*91.0585*	*0.5649*	*0.6173*
Image_8	*40.6702*	*91.1047*	*0.5636*	*0.617*
Image_9	*40.6686*	*91.0572*	*0.5658*	*0.6169*
Image_10	*40.6774*	*91.1151*	*0.5657*	*0.6169*

Image number	*mean*	*standard*	*entropy*	*energy*	*homogeneity*
Image_1	*43.1996*	*93.1829*	*0.5627*	*0.6037*	*0.7735*
Image_2	*40.3539*	*90.8815*	*0.5527*	*0.6193*	*0.7846*
Image_3	*43.4536*	*93.456*	*0.5696*	*0.6019*	*0.7725*
Image_4	*42.6145*	*92.6438*	*0.5625*	*0.6067*	*0.7757*
Image_5	*42.6288*	*92.8084*	*0.5691*	*0.6064*	*0.7757*
Image_6	*42.6724*	*92.7629*	*0.5698*	*0.606*	*0.7755*
Image_7	*42.7244*	*92.7146*	*0.5626*	*0.6062*	*0.7753*
Image_8	*42.8336*	*92.7799*	*0.5532*	*0.6062*	*0.7751*
Image_9	*42.6928*	*92.6916*	*0.5593*	*0.6066*	*0.7755*
Image_10	*42.6928*	*92.6916*	*0.5593*	*0.6066*	*0.7755*

step	*variance*	*3^{rd} moment*	*skew value*
0.9439	10911	2756267	21332
0.9439	10181	2569194	20994
0.9439	10975	2772379	21414
0.9439	10757	2715831	21268
0.9439	10764	2718604	21319
0.9439	10774	2720511	21329
0.9439	10787	2723858	21278
0.9439	10829	2738152	21294
0.9439	10784	2724260	21274
0.9439	10784	2724260	21274

Image name	*mean*	*standard*	*entropy*	*energy*	*homogeneity*
e1.jpg	43.4035	93.3789	0.566	0.6024	0.7727
e2.jpg	42.8221	92.867	0.5657	0.6055	0.7749
e3.jpg	45.3545	94.6669	0.5697	0.592	0.7651
e4.jpg	45.3037	94.7517	0.5745	0.592	0.7653
e5.jpg	45.3567	94.5969	0.567	0.5922	0.7651
e6.jpg	45.5556	94.8877	0.5719	0.5907	0.7643
e7.jpg	45.4715	94.7912	0.5686	0.5915	0.7647
e8.jpg	45.6181	95.0514	0.5755	0.5901	0.7641
e9.jpg	45.4181	94.7058	0.5687	0.5918	0.7649
e10.jpg	45.2131	94.6221	0.5672	0.593	0.7657
fl1.jpg	47.439	96.052	0.5739	0.5817	0.757
fl2.jpg	47.9479	96.4617	0.5732	0.5787	0.755
fl3.jpg	48.2037	96.2517	0.5656	0.5788	0.754
fl4.jpg	48.3713	96.5506	0.5635	0.5774	0.7534
fl5.jpg	48.5183	96.5111	0.5634	0.5768	0.7528
fl6.jpg	48.6835	96.6032	0.5696	0.5764	0.7522
fl7.jpg	48.4786	96.7154	0.569	0.5765	0.753
fl8.jpg	48.3221	96.5584	0.5668	0.5774	0.7536
fl9.jpg	48.5177	96.8713	0.5736	0.5756	0.7528
fl10.jpg	48.4667	96.191	0.5561	0.579	0.7532
h1.jpg	47.5627	96.3046	0.5844	0.58	0.7564
h2.jpg	47.5835	95.5368	0.5538	0.5828	0.7564

h3.jpg	*47.3237*	*95.9934*	*0.5706*	*0.5822*	*0.7574*
h4.jpg	*47.3289*	*96.0089*	*0.5701*	*0.5819*	*0.7574*
h5.jpg	*47.3261*	*96.0264*	*0.5724*	*0.5819*	*0.7574*
h6.jpg	*47.6375*	*96.084*	*0.5665*	*0.5808*	*0.7562*
h7.jpg	*47.3801*	*95.8081*	*0.5641*	*0.5825*	*0.7572*
h8.jpg	*47.5328*	*96.2051*	*0.5731*	*0.5809*	*0.7566*
h9.jpg	*47.3795*	*96.0024*	*0.5695*	*0.5819*	*0.7572*
h10.jpg	*47.4048*	*95.9576*	*0.5656*	*0.5825*	*0.7572*
m1.jpg	*46.6563*	*95.615*	*0.5723*	*0.5852*	*0.7601*
m2.jpg	*46.5542*	*95.5911*	*0.5697*	*0.5859*	*0.7605*
m3.jpg	*46.4558*	*95.2079*	*0.5581*	*0.5872*	*0.7609*
m4.jpg	*46.2837*	*95.3915*	*0.5753*	*0.5869*	*0.7615*
m5.jpg	*46.5534*	*95.5127*	*0.5688*	*0.586*	*0.7605*
m6.jpg	*46.3874*	*95.4782*	*0.5762*	*0.5862*	*0.7611*
m7.jpg	*46.8007*	*95.3393*	*0.5597*	*0.5856*	*0.7595*
m8.jpg	*46.3943*	*95.4835*	*0.5755*	*0.5863*	*0.7611*
m9.jpg	*46.6571*	*95.7205*	*0.5737*	*0.5851*	*0.7601*
m10.jpg	*46.4158*	*95.4765*	*0.5681*	*0.5866*	*0.7611*
fo1.jpg	*43.3053*	*93.2759*	*0.5635*	*0.6029*	*0.7731*
fo2.jpg	*41.2208*	*91.6048*	*0.5614*	*0.6142*	*0.7812*
fo3.jpg	*41.5647*	*91.9569*	*0.5688*	*0.6118*	*0.7797*
fo4.jpg	*41.2896*	*91.795*	*0.5562*	*0.614*	*0.781*
fo5.jpg	*41.5584*	*91.932*	*0.5724*	*0.6118*	*0.7797*
fo6.jpg	*41.4296*	*91.7403*	*0.5617*	*0.613*	*0.7803*
fo7.jpg	*41.2152*	*91.6817*	*0.5628*	*0.6142*	*0.7812*
fo8.jpg	*41.8168*	*92.1205*	*0.5569*	*0.6112*	*0.7789*
fo9.jpg	*41.5947*	*91.8403*	*0.5606*	*0.6122*	*0.7797*
fo10.jpg	*41.0956*	*91.4835*	*0.5649*	*0.6147*	*0.7816*

step	***variance***	***moment***	***skew value***
0.9439	*10963*	*2769318*	*21372*
0.9439	*10811*	*2729580*	*21298*
0.9439	*11459*	*2895645*	*21538*
0.9439	*11446*	*2892440*	*21573*
0.9439	*11460*	*2896114*	*21510*
0.9439	*11509*	*2908172*	*21579*
0.9439	*11493*	*2905060*	*21556*

0.9439	*11528*	*2913589*	*21643*
0.9439	*11478*	*2901288*	*21540*
0.9439	*11426*	*2888056*	*21519*
0.9439	*12126*	*3066921*	*2176*
0.9439	*12191*	*3083340*	*21582*
0.9439	*12237*	*3095884*	*21683*
0.9439	*12272*	*3104576*	*21657*
0.9439	*12317*	*3116790*	*21632*
0.9439	*12265*	*3103463*	*21759*
0.9439	*12225*	*3092977*	*21715*
0.9439	*12272*	*3104447*	*21813*
0.9439	*12272*	*3107831*	*21538*
0.9439	*12022*	*3038824*	*21771*
0.9439	*12032*	*3042611*	*21421*
0.9439	*11965*	*3025446*	*21669*
0.9439	*11968*	*3026437*	*21698*
0.9439	*11966*	*3025903*	*21703*
0.9439	*12046*	*3046492*	*21661*
0.9439	*11981*	*3029738*	*21595*
0.9439	*12019.628*	*3039485*	*21700*
0.9439	*11980*	*3029614*	*21673*
0.9439	*11993*	*3034477*	*21628*
0.9439	*11795*	*2982237*	*21670*
0.9439	*11769*	*2975696*	*21646*
0.9439	*11745*	*2969810*	*21523*
0.9439	*11697*	*2956476*	*21654*
0.9439	*11769*	*2975572*	*21635*
0.9439	*11724*	*2963318*	*21674*
0.9439	*11829*	*2990327*	*21511*
0.9439	*11727*	*2964615*	*21675*
0.9439	*11796*	*2982398*	*21678*
0.9439	*11738*	*2968713*	*21671*
0.9439	*10939*	*2763465*	*21373*
0.9439	*10401*	*2624641*	*21118*
0.9439	*10484*	*2644971*	*21192*
0.9439	*10422*	*2631188*	*21134*
0.9439	*10481*	*2643891*	*21198*
0.9439	*10455*	*2638621*	*21148*
0.9439	*10398*	*2623600*	*21113*

0.9439	*10560*	*2667091*	*21195*
0.9439	*10499*	*2650668*	*21177*
0.9439	*10363*	*26138268*	*21093*

REFERENCES

[1] Liu, Y, D Zhang, G Lu, and W Ma. "A survey of content-based image retrieval with high-level semantics" January 2007.
[2] H.B. Kekre, Tanuja K. Sarode, Meena S. Ugale"Performance Comparison of Image Classifier using Discrete Cosine Transform and Walsh Transform" International Journal of Computer Applications® (IJCA) 2nd International Conference and workshop on Emerging Trends in Technology (ICWET) 2011.
[3] Joaquim Jose Furtado1 , Zhihua Cai1 & Liu Xiaobo11" Digital Image Processing :Supervised Classification using Genetic Algorithm in Matlab Toolbox" China University of Geosciences, 2010.
[4] Syed Ali Khayam" The Discrete Cosine Transform (DCT): Theory and Application" Department of Electrical & Computer Engineering Michigan State University March 10th 2003.
[5] Sunill Malviya M-tech Student, Neelesh Gupta HOD(ECE) "An Improved Image Compression Algorithm Based on Daubechies- Wavelets with Arithmetic Coding" Journal of Information Engineering and Applications, Vol.3, No.6, 2013.
[6] A. C. Gonzalez-Garcia1, 2, J. H. Sossa-Azuela 2, E. M. Felipe-Riveron2 and O. Pogrebnyak2" Image Retrieval Based on Wavelet Transform and Neural Network Classification" , April 20, 2007.
[7] H. Hashim, S. Ramli1, N. Wahid, M. S. Sulaiman, N. Hassan," Recognition of Psoriasis Features via Daubechies D8 Wavelet Technique ",International Journal on Smart Sensing and Intelligent Systems vol.6,no.2,april 2013.
[8] http://www.bearcave.com/misl/misl_tech/wavelets/daubechies/
[9] Jasmeet kaur1" A Combined DWT-DCT approach to perform Video compression base of Frame Redundancy" International Journal of Advanced
Research in Computer Science and Software Engineering, Volume 2, Issue 9, September 2012.
[10] John See " TDI2131 Digital Image Processing" Faculty of Information Technology Multimedia University Chapter 8: Image Compression 2003.
[11] P. Mohanaiah, P. Sathyanarayana, L. GuruKumar"Image Teature feature Extraction using GLCM Aproach" International Journal of Scientific and Research Publications, Volume 3, Issue 5, May 2013 1 ISSN 2250-3153.
[12] Mr. MilindV.Lande , Prof.PraveenBhanodiya , Mr.Pritesh Jain"Analysis and Comparison of Color Features for Content Based Image Retrieval" International Journal of Computers & Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.
[13] Jayamala K. Patil1 Raj Kumar2" Color Feature extraction of Tomato Leaf Diseases" International Journal of Engineering Trends and Technology- Volume2Issue2- 2011.
[14] Alberto Del Bimbo, " Visual Information Retrieval", Morgan Kaufmann, Inc.,San Francisco, California, 1999.

[15] S.Selvarajah 1 and S.R. Kodituwakku"Analysis and Comparison of Texture Features for Content Based Image Retrieval " International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) 108 Volume 2, Issue 1, March 2011.
[16] Holalu Seenappa Sheshadri and Arumugam Kandaswamy,"Breast Tissue Classification Using Stastical Feature Extraction of Mammograms", Department of Electronics & Communication Engineering, In final form June 2, 2006.
[17] Seema H.Jadhav1, Dr.Shah Aqueel Ahmed,"A Content Based Image Retrieval System using homogeneity Feature extraction from Recency-based Retrieved Image Library" IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 7, Issue 6 (Nov-Dec. 2012), PP 13-24.
[18] G. N. Srinivasan, and Shobha G, “Statistical Texture Analysis", processing of world academy of science, engineering and technology volume 36 december 2008 ISSN 2070-3740.
[19] Rakesh Kumar, Senior Member, IACSIT and Jyotishree, Member, IACSIT " Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms" International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.
[20] Fazal Malik and Baharum Bin Baharudin, "Feature Analysis of Quantized Histogram Color Features for Content-Based Image Retrieval Based on Laplacian Filter", International Conference on System Engineering and Modeling (ICSEM 2012).
[21] P.S Hiremath,Jagadeesh Pujari "Content Based Image Retrival based on Color, Texture and Shape features using Image and its complement", International Journal of Computer Science and Security.
[22] Manish Maheshwari, DR.Mahesh Motwani,DR.Sanjay Silakari, "New Feature Exraction Technique for Color Image Clustering”, International Journal of Comuter Science and Electronics Engineering(IJCSEE) Volume 1,Issue 1(2013) ISSN 2320-4028.