TEXTURE CLASSIFICATION BASED ON FUZZY LOGIC

***Rukaya A. Abd Al-Jabar , **Sawsan K. Thamer and ***Loay E. George**

****Al-Anbar University / College of Science/ / Department of Computer Science***

****AL-Nahrain University/ College of Science/ Department of Computer Science**

*****Baghdad University / College of Science/ IT-Unit**

Abstract

The classification process is an important task in many application of computer image analysis for classifying images based on color or texture low-level features.

In this work, a texture classification system is presented which supports querying with respect to texture low-level feature. The fundamental idea is to generate automatically image description by analyzing the image content. The underlying techniques are based on the Gray-Level Co-occurrence Matrix (GLCM) and Gray-Level Run Length Matrix (GLRLM) as statistical approaches to texture analysis. These two techniques are applied in separated manner.

Each class is represented by features vector(s) in the features space and stored in a file. Then, a selection to the best set of features is done using fuzzy concepts (triangular membership functions or trapezoidal membership functions). Given the query image, the system first extracts its features vector, and then compares the selected features with those stored in the file to find the nearest class using fuzzy concept.

During the evaluation process, it was found that; the best results are obtained from combination among features which in turn achieve higher selection rate for features as well as for whole system (gets selection rate nearly 90% for combination among four features and 60% without combination)..

Index Terms—*Image Processing, Feature Extraction, Texture Classification and Fuzzy Logic.*

المستخلص :

تصنيف التركيب النسيجي بالاعتماد على القاعدة المضببة

تعتبر عملية التصنيف واحدة من أهم العمليات في العديد من تطبيقات الحاسوب لغرض تصنيف الصور بالنسبة الى الصفات المنخفضة المستوى كاللون أوالتركيب النسيجي للصورة. في هذا العمل يتم عرض نظام التصنيف الذي يدعم الاستفسار بالنسبة إلى خصائص التركيب النسيجي .تتلخص الفكرة الأساسية للعمل في توليد ميزات الصورة أوتوماتيكيا عن طريق تحليل محتوياتها. تعتمد التقنيات المضمنة على اتخاذ مصفوفة الظهور المتلازم للألوان الرمادية و مصفوفة طول السلسلة للألوان الرمادية كوسائل إحصائية لتحليل التركيب النسيجي. و قد طبقت هذه الأساليب بحالات منفردة. يتمثل كل صنف بواسطة متجه أو(مجموعة متجهات)في فضاء الصفات و تخزن بملف .لاحقا يتم اختيار أفضل مجموعة من المتجهات للصورة باستعمال مفاهيم مضببة (وظائف عضوية مثلثية او وظائف عضوية رباعية).عند الاستعلام عن الصورة ,يتم استخلاص متجه الصفات لهذه الصورة و من ثم مقارنة المتجهات المختارة المخزونة في الملف لايجاد اقرب صنف باستعمال القواعد المضببة.

خلال عملية التقييم وجد أن افضل النتائج نحصل عليها من الجمع بين الميزاتِ و الذي يحقق نسب اعلى لإختيار الميزاتِ بالإضافة إلى النظام ككل(تُصبح نسب الاختيار تقريبا 90%عند تجميع بين أربع ميزاتِ و % 60 بدون تجميع).

I. INTRODUCTION

Computer vision refers to the field of computer science that is concerned with the design and implementation of algorithms that allow machines to simulate human vision. Various fields related to computer processing of images are categorized according to the type of input which they take and type of results they produce; these categories are [27]:

- Image processing.
- Computer graphics.
- Pattern recognition and Computer vision.

In a typical pattern recognition or object classification process, the first step is the extraction of features or key properties of objects (i.e. mapping from the real world to the feature space). The next step is classification of objects according to their features (i.e. mapping from the feature space to the classification space). The human brain is an excellent classifier which can successfully classify objects in noisy environments even without significant features. However, it still cannot be expected the same performance from our artificial classifiers. Therefore, to work towards a successful classification, extracted features of different objects must show adequate separation in the feature space.

Texture analysis is one of the most important techniques used in analysis and classification of images presenting repetition of fundamental image elements. Texture can be recognized when it is seen, but it is a very difficult concept to define [9,30].

In classical two-state logic, an element either belongs or does not belong to a given class. In real life, however, the classes are often ill defined, or overlapping, or fuzzy and a pattern may belong to more than one class. In such a situation, the fuzzy set theoretic techniques have proved to be useful.

There has an increasing use of fuzzy set theory and fuzzy algorithms for image processing implementations. This is motivated by desire to model the ambiguity and noise contained in digitally defined image [4].

II. TEXTURE CLASSIFICATION

A. Texture

Quantitative study of images is often concerned with four types of parameters, which are of fundamental importance. These are Contrast (a very important measure in image processing which often determines the quality of an image), Color (adds more useful discriminatory information to the image), Shape (a measure which is used in recognizing the various object contained in an image), and Texture (Describe the spatial distribution of tonal value within band and provide useful information for performing automatic interpretation and recognition) [2].

B. Texture Analysis and its Applications

Major goals of texture research in computer vision are to understand, model and process texture, and ultimately to simulate human visual learning process using computer technologies. Four major application domains related to texture analysis are texture classification, texture segmentation, shape from texture, and texture synthesis
[38].

C. Texture Classification

Texture classification assigns a given texture to some texture classes. A supervised classifier is trained using set of images to learn the characterization for each texture class. [35,38].

Before the classification process is worked, the training process is done, where some known texture images are used to train the classifier. Training typically involves four major steps. These are:

- Image pre-processing,
- Sampling,
- Feature extraction,
- Classifier training.

The pre-processing step is used typically for image enhancement and noise removal, although some techniques perform scaling and rotation in this step as well, in order to compensate for variations in the training data.

Image data are often limited in terms of the number of original source images available, so sampling process is done in order to increase the amount of data which divided the images into sub images, either overlapped or disjoint, of a particular window size.

The most important stage of the classification process is the feature extraction stage, at which time the sample image is transformed into a much lower dimensionality feature vector. Many different techniques have been used to handle this stage.

The vectors from all of the training images are then input to the classification system for training. . Again, many different classifiers have been used, although some of them perform slightly better than others, generally the choice of classifier has the least effect on the overall performance of the system. Therefore, speed of training, ease of implementation, and suitability to a given task are more important factors in the choice of a classifier than is its raw performance [31].

III. FEATURE SELECTION AND EXTRACTION

Any pattern which can be classified in some category must possess a number of features. The first step in the process of classification is to consider the problem, what features to select and how to extract (measure) them [7].

A judicious selection of features for building classifiers is a very crucial aspect of classifier design, and deserves careful consideration. On one hand, there is certainly nothing to lose in using all available measurements in classifier design. On the other hand, too many features make the classifier increasingly complex (sometimes confusing too), in fact, unnecessarily so, in case some of the measurements are redundant.

Feature selection, is essentially the selection of the subset of measurements that optimizes some criterion of separability of classes, since, intuitively, the best set of features should discriminate most efficiently among the classes, that is, enhance the separability among them, while increasing homogeneity within classes at the same time.

Feature extraction, aims to reduce the number of measurements available in a different way by looking for a transformation of the original vector of measurements that optimizes some appropriately defined criterion of separability among classes, possibly leading to fewer features at the same time.

There are various methods of extracting texture features from images, some of these methods are: statistical, structural (or syntactic), and spectral [7, 28]. In this work, Co-occurrence and Run length matrices where used to extract image features.

IV. CLASSIFICATION USING FUZZY LOGIC

Image classification is an area where fuzzy representation and fuzzy reasoning can be successfully applied, mainly for two reasons: (1) ambiguity in the images to be recognized; and (2) the need for fast processing, that is, complicated formulas may not be applicable for a real-time recognition; in this case a fuzzy system may be more convenient.

Different approaches are possible depending on the image recognition tasks, two of them being (1) objects recognition, that is, recognizing shape, distance, and location of objects; and (2) texture analysis, for example, an image X of size $m \times n$ pixels can be represented as a set of fuzzy sets and membership degrees to which pixels belong to the fuzzy concepts, such as "brightness," "darkness," "edginess," "smoothness" [16].

V. FUZZY LOGIC CONCEPTS

The mathematical logic is called classical logic. The classical logic considers the binary logic which consists of truth and false. The fuzzy logic is a generalization of the classical logic and deals with the ambiguity in the logic [19].

Fuzzy logic is relatively young theory. Major advantage of this theory is that it allows the natural description, in linguistic terms, of problems that should be solved rather than in terms of relationships between precise numerical values. This advantage, dealing with the complicated systems in simple way, is the main reason why fuzzy logic theory is widely applied in technique. It is also possible to classify the remotely sensed image (as well as any other digital imagery); in such a way that certain land cover classes are clearly represented in the resulting image [26].

Fuzzy image processing is a kind of nonlinear image processing. The difference to other well-known methodologies is that fuzzy techniques operate on membership values. The image fuzzification (generation of suitable membership values) is, therefore, the first processing step. Generally, three various types of image fuzzification can be distinguished: histogram-based gray-level fuzzification, local neighborhood fuzzification, and feature fuzzification [13].

Some of the main characteristics of the fuzzy systems are [16]:

- Fuzzy concepts have to have linguistic meaning; they need to be articulated.
- Membership functions are numerical representations of the linguistic concepts; they can be built either through learning from data, or through experts' opinion, or through both.
- Fuzzy rules can represent vague, ambiguous or contradictory knowledge.
- Fuzzy systems are robust; even if some rules are removed from the rule map, the system could still work properly; fuzzy systems are also robust toward changing conditions in the environment.
- Fuzzy systems are simple to build, easy to realize, easy to explain.

VI. FUZZY-BASED TEXTURE CLASSIFICATION SYSTEM

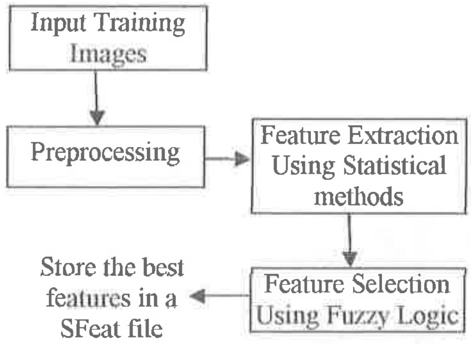
This section is devoted to describe Fuzzy-Based Texture Classification System (FTCS) implementation to offer the facilities, which may be required to perform the classification process by using fuzzy logic.

The fundamental idea of this work is to implement a program which takes texture image as an input and produces image class as output. This task is accomplished by using supervised classification, and statistical approaches have been used to extract texture features of images using co-occurrence and run length matrices. Fuzzy logic is used for computing membership values for all extracted features using triangular and trapezoidal membership functions. Then a combination between features is used to select the best features which satisfy the best selection rate and also using fuzzy logic with combination for comparing and producing the nearest classes to test texture images from training texture images.

A. FTCS Structure

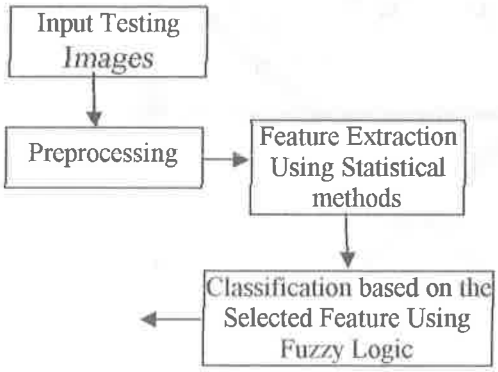
Texture classification assigns a given texture to some texture classes. Supervised classification is providing an example for each texture class as a training set. There are two phases to do classification process:

- Learning phase (offline phase), the target is to build a model for the texture content of each texture class presented in the training data, which generally comprises of images with known class labels. The texture content of the training images is captured with the chosen texture analysis method, which yields a set of textural features for each image. These features, could be scalar numbers, discrete histograms or empirical distributions, characterize a given textural properties of the images, such as spatial structure, contrast, roughness, orientation, etc. Figure (1) shows the basic modules for learning phase diagram.



Figure(1): Learning phase

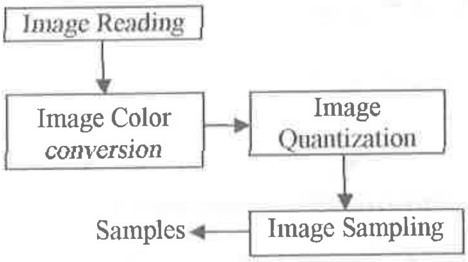
- Classification phase (online phase), the texture content of the unknown image is first described with the same texture analysis method applied in the learning phase. Then the textural features of the image are compared with those of the training images using classification algorithm and the image is assigned to the category with the best match. Figure (2) shows the basic module for classification phase diagram.



Figure(2): Classification phase

B. Image Preprocessing

The input to this step is an image of bitmap (BMP) type (24 bit/pixel) and the output are quantized gray samples. This stage contains a set of modules. Figure(3) shows the basic modules for preprocessing step.



Figure(3): Preprocessing Stage

- Image Reading module: Extracting required information from image file (image-width, image-height, and the image-data
- Image Color Conversion module: This module concerned with converting the color image to gray image.
- Image Quantization Module: The main drawback in using the Co-occurrence and Run Length matrices is the large memory requirement for storing these matrices. Quantization process overcomes this problem by removing some of the information details by mapping groups of data points to a single point. For this reason image quantization process is adopted in this work.
- Image Sampling Modules: The quantized image will be divided randomly into small sub-images (of size SLen×SLen) to increase the amount of data for each image which increase the discrimination between images, where SLen must be greater than 0 and less than width and height of the image.

C. Features Extraction

A feature vector is one method to represent an image, or part of an image object, by finding measurements on a set of features. The statistical features is one of the most important features that is used to evaluate the performance of co-occurrence matrices and run length matrices for solving texture classification problem, thus, statistical features are adopted in this work.

The texture feature extraction is applied for each sample get from gray image. The computation of the statistical features can be summarized by the following two modules:

- Module-1: For each sample the co-occurrence matrix C is extracted in a four direction and the total for these directions, the co-occurrence matrix is a two dimensional matrix of joint probability C(I,J) between pairs of pixels separated by a distance d in a given direction. Twenty one statistical texture features are calculated depending on the extracted co-occurrence matrix C. These 21 statistical texture features which are computed in this work are: Energy, Contrast, Correlation, Variance, Inverse difference moment, Entropy, Sum Average, Sum Entropy, Sum Variance, Difference Variance, Difference Entropy, Information measures of correlation1, Information measures of correlation2, Maximum Probability, Difference Moment of order 1, 2, 3, and 4, Homogeneity, Cluster Shade, and Cluster prominence.
- Module2: For each sample the run length matrix RL is extracted in a four direction and the total for these directions; 11 statistical texture features are calculated depending on the extracted run length matrix RL. These 11 statistical texture features which are computed in this work are: Short Run Emphasis, Long Run Emphasis, Gray Level Distribution, Run Length Distribution, Run Percentage, Low Gray-

Level Run Emphasis, High Gray-Level Run Emphasis, Short Run Low Gray-level Emphasis, Short Run High Gray-level Emphasis, Long Run Low Gray-level Emphasis, and Long Run High Gray-level Emphasis.

D. Fuzzification Process

Before the fuzzification process is done, extracted features are both normalized and quantized. This step is required to unify the dynamic ranges of the extracted features.

The *applied normalization process maps the extracted feature*'s values to the range [0, 1]. This was performed by finding the actual dynamic range (i.e., the highest and lowest values) [fmin, fmax] of each feature over all classes, taken into consideration that there are many samples in each class. The normalization of feature (f) is performed *using the following equation:*

$$f_{norm} = \frac{f - f_{min}}{f_{max} - f_{min}} \quad ......(1)$$

The uniform quantization process is used to map the normalized real values of the features to discrete integer indices, whose values lay within the range [0, Nbin-1], where Nbin is the number of quantization bins. Fuzzification is the operation of transforming a crisp set to fuzzy set, so each extracted feature is represented by a membership function. Two types of membership functions have been used, they are: triangular and trapezoidal. The fuzzification module is concerned with finding the parameters of the best triangular or trapezoidal membership function that fits the Probability Density Function (pdf) of each feature in each class.

Before the near optimal membership functions are computed, the histogram of each feature must be computed for each class. The corresponding probability density function (pdf) is computed from the histogram, and then it is used to find the optimum values of the membership function parameters (i.e. A, B and C values for the triangular membership function and the A, B, C and D values for the trapezoidal membership function) by using distance measure"χ^2".

The distance measure "χ^2" was used to find the minimum distance between pdf bin values and the corresponding membership values. χ^2 Distance measure is computed as follows:

$$X^2 = \sum_{i=0}^{Nbin-1} |pdf(i) - memebership(i)| \;......(2)$$

The parameters of membership functions that led the minimum sum of absolute differences (i.e.,χ^2) have been considered as the optimal values.

E. Feature Selection

Finding a specific features vector that has the best discrimination power has been one of the most important problems in the field of texture analysis and image classification. In practice a larger than necessary number of feature candidates is generated and then the best of them is adopted.

In this work, fuzzy logic is used to select the best features by converting the texture features to fuzzy numbers, which compute the membership values for each feature in all classes, and find the best features vector by calculating the success rate for each feature. The computation of success rate is done according to the following criteria: if the extracted feature from a class has highest membership value in that class relative to other classes, then the value of success rate of that feature is incremented by 1,. The steps taken to select the good features depend on the values for the success rates.

F. Classification

In this process, the classifier is trained to determine the class for each input image based on the obtained measures of the selected features. In this case, a classifier is a function which takes the selected features as input and texture classes as output by using fuzzy logic. To find the match class, first the features is extracted for tested image, compute membership values for each feature in each training class (i.e. using the same parameters for training classes (A, B, C for triangular and A, B, C, D for trapezoidal)), then search for the class which achieve higher membership value for each selected feature and increase the success rates for that class by 1, finally search for the match class which has high success rate.

G. Features combinations

To enhance the performance of the system for selection and classification process, the system will include the combination between two or more features by adding membership values for these features, and then re-determines the success rates for these combined features, and finding the best combined set of features based on success rates to be used in selection and classification process. Various combinations of two, three, and four features have been investigated. Other (bigger) combinations have been applied in similar way. The same way is applied in classification by comparing the combined features to find the class type.

VII. TESTS AND RESULTS

The FTCS were established using Visual Basic (version 6.0) programming language. The tests have been applied using a personal computer (Pentium 4, processor 1.60 GHz, RAM 1 G-byte).

A. Test Material

In FTCS, 10 textured images are selected to represent 10 different classes. These classes represent the training images for the system. These images have size 256×256 pixels with color resolution 24 *bit/pixel, as shown in figure* (4).

To perform the texture classification testing, 20 textured images are chosen for soft testing (which means that the test images are taken from the training images either part of it or rotate it), as shown in figure (5) and 30 textured images are chosen for hard testing, as shown in figure (6). *These images have different size with color resolution 24 bit/pixel.*

B. FTCS Models Analysis

The analysis includes testing various parameter values, number of gray levels was varied to be 16, 32 or 64, the distance between pixels for calculation of co-occurrence matrix was varied to be 1,2, or 3, number of selected sample was varied to be 100, 200, 500 or 1000, the length of each samples was varied to be 10, 25, 50, 75 or 100, which mapping type chose to do quantization also varied to be 1, 2, 3,or 4, and finally determine the membership types which take either trapezoidal or triangular. So the variation in the parameters will cause variation in the results of success rates for the features.

Features computed using co-occurrence and run length matrices are shown in table (1) and table (2) respectively, where column 1 represent the name of the features, column 2 to column 5 (Th0, Th45, Th90, Th135) represent the theta of the direction, and column 6 (Avg) represent the average for the four direction. Some of the experiment results are applied according to the parameters mentioned in table (3), which declares the cases taken in the results and the parameters which affect the success rate for features where column 1 (CaseNo) represents case number, column 2 (Mtype) represents mapping type, and column 3 (Glevel) represents gray level for quantization. Experiments with different parameters values are applied in table (4) to table (9), which contain maximum success rate value for different combinations of features, where column 1 (CaseNo) represent case number, column 2 (single) represent success rates for features with out combination, column 2 to column 4 represent success rates for features with combination (Comb2, Comb3, Comb4), and column 5(Trap/Traing) represent membership function that used in selection features. All of these tables applied results according to distance 1, but different distance is given in table (10).

Finally, the best features for the images used in figure (4) are obtained from the results for different experiments are shown in table (11) with column 1 (Best CocFeat) for best co-occurrence features and column 2 (Best RLFeat) for best run length features.

VIII. CONCLUSIONS

This paper is devoted to present the derived conclusions concerned with the performance of the classification methods based on using statistical features as discriminating attributes. Authors should consider the following points:

1. The co-occurrence matrices are calculated for the distances 1, 2, and 3. Best results for these selected images are achieved when distance between pixels for co-occurrence matrix is 1.
2. Best results are achieved when using general equation techniques to generate lookup table.
3. The length and number of the samples play an important role for increasing accuracy for the classification process.
4. Trapezoidal membership yield success rates better than triangular membership but the difference does not give a high improvement (93%, 91% respectively).
5. Co-occurrence method yields selection rates better than run length method (90%, 85% respectively).
6. The performance of the system increases when using combination between features (60% without combination, 75% with combination two features, 85% with combination three features, and 90% with combination four features).
7. Performance results nearly 95% for soft testing and 85% for hard testing.

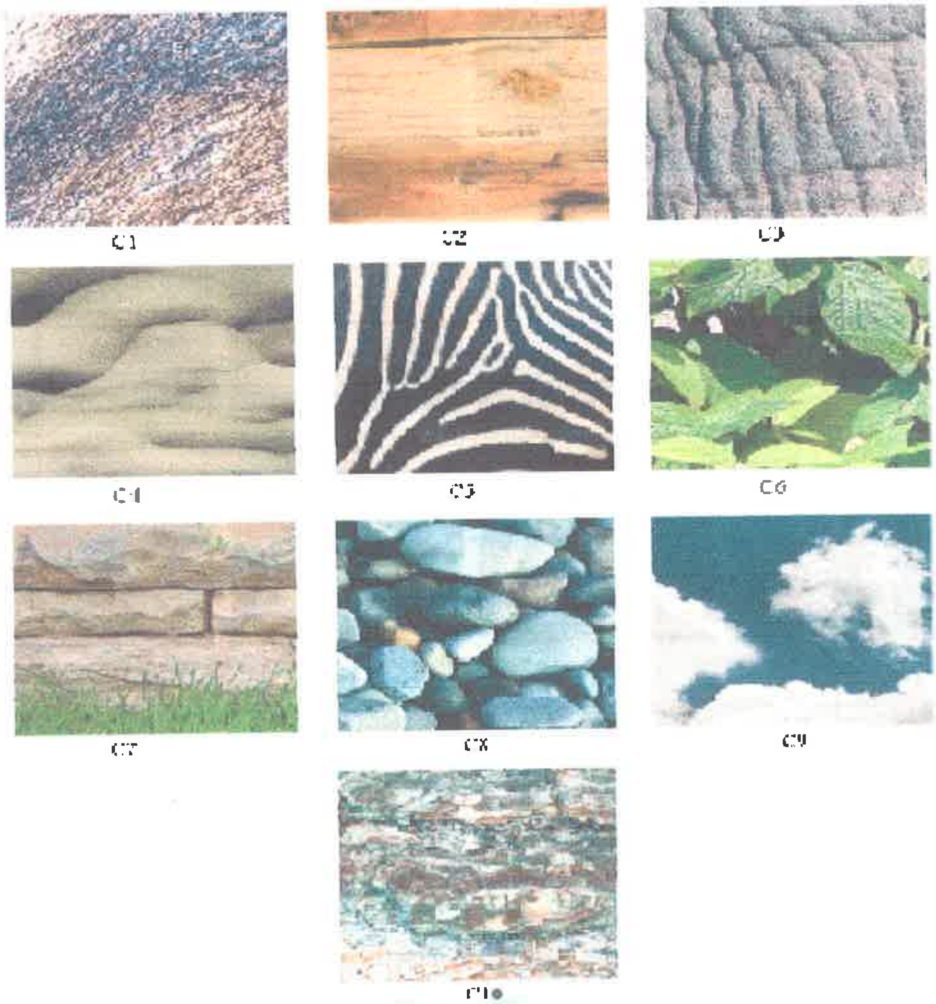


Figure (4): Training Images.

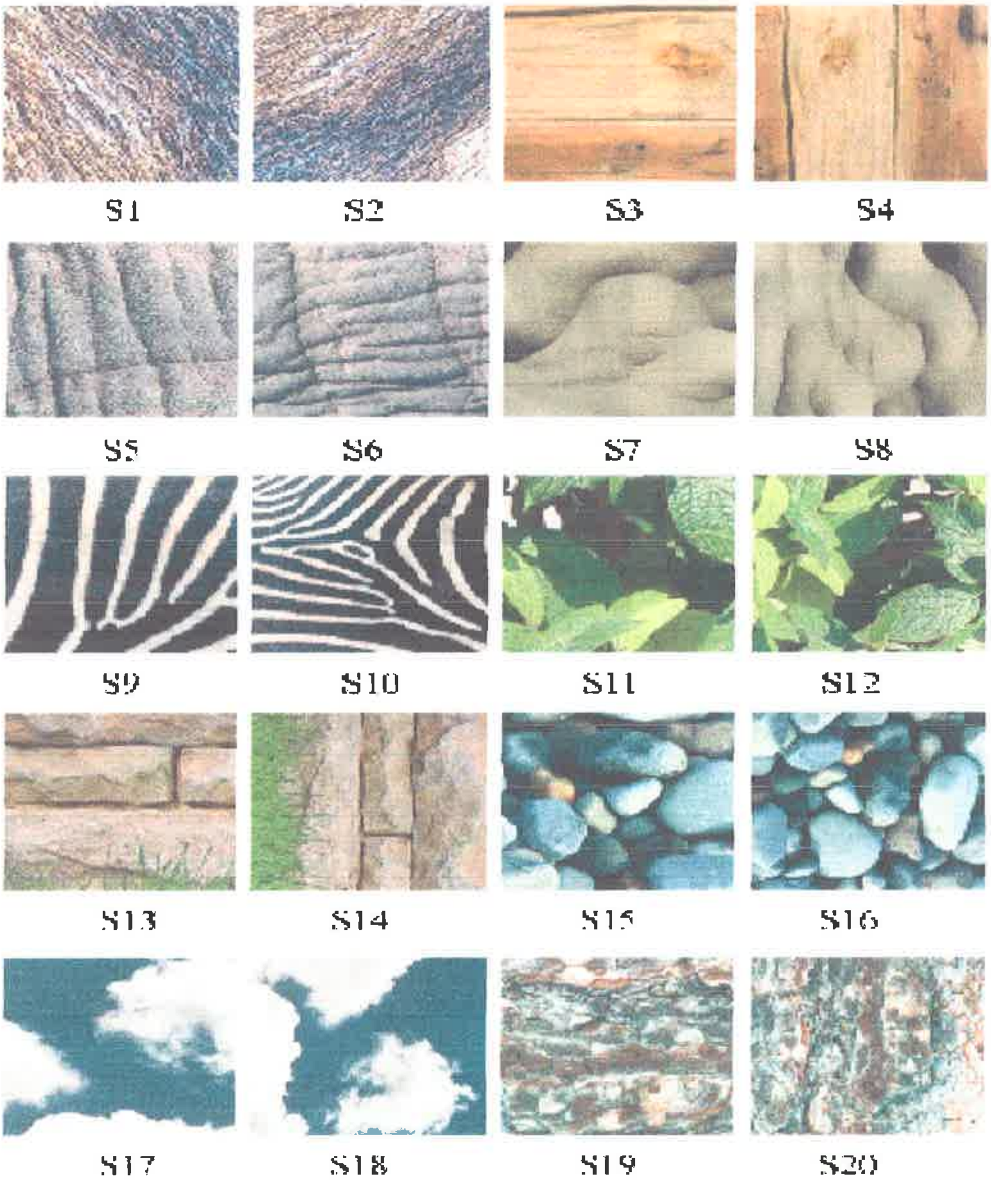


Figure (5): Test Images using in Soft Testing Process.

Figure (6): Test Images using in Hard Testing

Table (1) The index number of each co-occurrence features

FeatureName	Th0	Th45	Th90	Th135	Avg
Energy	F0	F21	F42	F63	F84
Contrast	F1	F22	F43	F64	F85
Correlation	F2	F23	F44	F65	F86
Variance	F3	F24	F45	F66	F87
InvDifMom	F4	F25	F46	F67	F88
Entropy	F5	F26	F47	F68	F89
SumAvg	F6	F27	F48	F69	F90
SumEnt	F7	F28	F49	F70	F91
SumVar	F8	F29	F50	F71	F92
DifVar	F9	F30	F51	F72	F93
DifEnt	F10	F31	F52	F73	F94
InfMeasCor1	F11	F32	F53	F74	F95
InfMeasCor2	F12	F33	F54	F75	F96
MaxProb	F13	F34	F55	F76	F97
DifMom1	F14	F35	F56	F77	F98
DifMom2	F15	F36	F57	F78	F99
DifMom3	F16	F37	F58	F79	F100
DifMom4	F17	F38	F59	F80	F101
Homog	F18	F39	F60	F81	F102
ClusterShade	F19	F40	F61	F82	F103
ClusterProm	F20	F41	F62	F83	F104

Table (2) The index number of each run length features

FeatureName	Th0	Th45	Th90	Th135	Avg
RP	F0	F11	F22	F33	F44
SRE	F1	F12	F23	F34	F45
LRE	F2	F13	F24	F35	F46
RLD	F3	F14	F25	F36	F47
LGRE	F4	F15	F26	F37	F48
HGRE	F5	F16	F27	F38	F49
GLN	F6	F17	F28	F39	F50
SRLGE	F7	F18	F29	F40	F51
SRHGE	F8	F19	F30	F41	F52
LRLGE	F9	F20	F31	F42	F53
LRHGE	F10	F21	F32	F43	F54

Table (3) Parameters for specific case

CaseNo	Mtype	Glevel
1	1	16
2	2	16
3	3	16
4	4	16
5	1	32
6	2	32
7	3	32
8	4	32
9	1	64
10	2	64
11	3	64
12	4	64

Table (4) Success Rates % for different combinations for Co-occurrence features for 500 samples with length 50

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	51	69	74	76	Traing
1	51	70	74	78	Trap
2	48	64	69	73	Traing
2	46	66	71	75	Trap
3	53	67	72	75	Traing
3	48	68	73	76	Trap
4	44	60	64	66	Traing
4	43	63	67	69	Trap

Table (5) Success Rates % for different combinations for Run length features for 500 samples with length 50

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	49	60	67	72	Traing
1	48	62	67	71	Trap
2	48	57	62	67	Traing
2	43	57	65	66	Trap
3	47	61	67	67	Traing
3	46	62	68	69	Trap
4	44	57	57	58	Traing
4	40	54	62	64	Trap

Table (6) Success Rates % for different combinations for Co-occurrence features for 500 samples with length 100

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	60	86	89	92	Traing
1	60	86	90	93	Trap
2	60	83	87	89	Traing
2	55	81	89	90	Trap
3	64	83	89	90	Traing
3	59	83	90	92	Trap
4	62	78	82	85	Traing
4	58	77	85	87	Trap
5	60	85	91	94	Traing
5	60	86	92	95	Trap
6	60	85	87	90	Traing
6	56	81	89	92	Trap
7	65	84	90	91	Traing
7	59	79	89	92	Trap
8	64	77	83	85	Traing
8	58	77	83	86	Trap

Table (7) Success Rates % for different combinations for Run length features for 500 samples with length 100

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	60	76	85	90	Traing
1	60	78	86	91	Trap
2	58	75	82	87	Traing
2	56	79	82	85	Trap
3	59	76	81	86	Traing
3	56	75	83	89	Trap
4	58	73	78	82	Traing
4	59	74	82	83	Trap
5	55	74	80	83	Traing
5	55	74	80	84	Trap
6	55	69	78	81	Traing
6	52	74	78	81	Trap
7	59	76	80	80	Traing
7	57	71	80	85	Trap
8	53	71	75	76	Traing
8	51	68	76	79	Trap

Table (8) Success Rates % for different combinations for Co-occurrence features for 1000 samples with length 100

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	61	85	89	92	Traing
1	60	86	90	93	Trap
2	61	83	88	90	Traing
2	53	79	88	91	Trap
3	62	84	90	93	Traing
3	60	85	89	92	Trap
4	62	75	82	85	Traing
4	57	75	84	86	Trap
5	62	86	89	91	Traing
5	55	76	80	84	Trap
6	61	83	88	90	Traing
6	54	74	81	83	Trap
7	63	85	88	91	Traing
7	55	74	8	84	Trap
8	64	77	83	85	Traing
8	64	77	83	85	Trap
9	64	86	89	93	Traing
9	58	83	89	92	Trap
10	60	82	86	89	Traing
10	52	80	88	90	Trap
11	62	82	89	93	Traing
11	57	80	90	92	Trap
12	61	78	84	88	Traing
12	57	77	86	81	Trap

Table (9) Success Rates % for different combinations for Run length features for 1000 samples with length 100

CaseNo	Single	Comb2	Comb3	Comb4	Trap/Traing
1	59	76	86	90	Traing
1	61	79	87	92	Trap
2	59	78	83	87	Traing
2	55	79	84	87	Trap
3	60	77	82	86	Traing
3	57	76	83	88	Trap
4	46	68	80	82	Traing
4	57	74	82	83	Trap
5	61	75	82	86	Traing
5	56	8	91	92	Trap
6	54	77	80	83	Traing
6	59	79	89	92	Trap
7	60	71	80	83	Traing
7	58	77	84	86	Trap
8	54	7	76	78	Traing
8	50	68	76	80	Trap
9	61	74	79	83	Traing
9	48	70	79	8	Trap
10	55	76	79	83	Traing
10	50	72	78	82	Trap
11	60	74	79	80	Traing
11	53	69	78	78	Trap
12	49	68	75	76	Traing
12	47	66	74	77	Trap

Table (10) Success Rates % for different combinations for Co-occurrence features for 500 samples with length 100

CaseNo	distance	Single	Comb2	Comb3	Comb4	Trap/Traing
1	1	60	86	89	92	Traing
1	2	59	81	87	90	Traing
1	3	55	78	86	88	Traing

Table (11) Best co-occurrence and run length features

Best CocFeat	Best RLFeat
F5	F0
F9	F1
F10	F3
F23	F6
F26	F17
F31	F22
F40	F23
F41	F25
F44	F26
F46	F28
F50	F29
F51	F34
F52	F36
F53	F39
F54	F40
F60	F45
F65	F47
F83	F50
F86	F51
F94	F52

IX. FUTURE WORKS

1. FTCS use image with single texture, so to make the system more flexible for multi texture the pre step for this system can be added, which segment an image into regions with the same texture, i.e. as a complement to grey level or color before classifying it.
2. Using structural attribute in addition to statistical attribute to increase the recognition accuracy.
3. Combine between fuzzy and neural to increase the performance of the system.
4. Using different color models to make the system more flexible, such as HSV (Hue, Saturation and Value) and HSI (Hue, Saturation and Intensity) which derived from RGB color model.

REFERENCES

[1] T. Acharya and A. K. Ray, "Image Processing: Principles and Applications", Wiley, 2005.

[2] L. Abdul Aziz Al-Ani, "Classification of Digital Satellite Images",(Ph.D.) thesis, College of Science, Al-Nahrain University, Iraq, 1996.

[3] Albregtsen, F., "Statistical Texture Measures Computed from Gray Level Run Length Matrices", Image Processing Laboratory, Department of Informatics, Oslo University, 1995.

[4] R. S. Ali, "Image Segmentation Using Fuzzy and Neural Networks", MSC thesis, College of science, Al-Nahrain University, Iraq, 1999.

[5] M. G. Duaimi, "Development of a Content-Based Image Retrieval System", (Ph.D) thesis, College of science, Al-Nahrain University, 2006.

[6] R. O. Duda, P. E. Hart and D. G. Stork, "Pattern Classification", 2nd edition, Wiley, 2000.
[7] M. Friedman and A. Kandel, "Introduction to Pattern Recognition: Statistical, Structural, Neural and Fuzzy Logic Approaches", World Scientific, 1999.
[8] R. C. Gonzalez and R. E. Gonzalez, "Digital Image Processing", Addison Wesley Publishing Company, 1987.
[9] J. Guibert, D. C. He and L. Wang, "Texture Discrimination Based on Optimal Utilization of Texture Feature", Pattern Recognition, Vol.21, No.2, PP.141-149, 1988.
[10] R. M. Haralick, "Statistical and Structural Approaches to Texture", Proceedings of the IEEE, Vol.67, PP.786-804, 1979.
[11] K. Hirota and W. Pedrycz, "Implicitly - supervised fuzzy Pattern recognition", IEEE, 1994.
[12]: A. M. Ibrahim, "Fuzzy Logic: for Embedded Systems Applications", Elsevier Science, 2004.
[13] B. Jähne, H. Haussecker and P. Geissler, "Handbook of Computer Vision and Applications: Volume2 Signal Processing and Pattern Recognition", Academic Press, 1999.
[14] L. C. Jain and N. M. Martin, "Fusion of Neural Networks, Fuzzy Systems, and Genetic Algorithms: Industrial Applications", CRC Press LLC, 1998.
[15] K. Karu and A. K. Jain, "Learning Texture Discrimination Masks", Proc. IEEE Int'l Conf Neural Networks, June, PP.4374-4397, Orlando, 1994.
[16] N. K. Kasabov, "Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering", Massachusetts Institute, 1996.

[17] E. S. Konak, "A Content-Based Image Retrieval System for Texture and Color Queries", MSC thesis, Computer Engineering, Engineering and Science Bilkent University, 2002.

[18] L. I. Kuncheva, "Combining Pattern Classifier: Methods and Algorithms", Wiley, 2004.

[19] K. H. Lee, "First Course on Fuzzy Theory And Applications", Springer, 2005.

[20] C. Limpsangsri, "Image Retrieval using texture", DePaul University, 2004.

[21] J. P. Marques de Sa, "Pattern Recognition: Concepts, Methods and Applications", Springer, 2001.

[22] A. Materka and M. Strzelecki, "Texture Analysis Methods", Technical University of Lodz, Institute of Electronics, Brussels, 1998.

[23] F. M. McNeill and E. Thro, "Fuzzy Logic: A Practical Approach", Academic Press, 1994.

[24] M. H. Mohammed and S. M. Faizur Rahman, "Fuzzy Features Extraction from BANGLA Handwritten Character", International Conference on Information and Communication Technology ICICT, March 2007.

[25] A. Monadjemi, "Towards Efficient Texture Classification and Abnormality Detection", (Ph.D.) thesis, Department of Computer Science, Faculty of Engineering, University of Bristol, October 2004.

[26] I. Nedeljkovic, "Image Classification Based on Fuzzy Logic", Fuzzy Inference System FIS, 2004.

[27] W. Niblack, "An Introduction to Digital Image Processing", Prentice_Hall International, 1986.

[28] S. K. Pal and A. Pal, "Pattern Recognition from Classical to Modern Approaches", World Scientific, 2001.
29. [Pra01]: W. K. Pratt, "Digital Image Processing", 3rd edition, Wiley, 2001.
[30] J. J. Roan, J. K. Aggarwal and W. N. Martin, "Multiple Resolution Imagery and Texture Analysis", Pattern Recognition, Vol.20, No.1, PP.17-31, 1987.
[31] K. H. Sager, "Fractal Based Classification for Color
Textural Images", (Ph.D.) thesis, College of Science, Baghdad University, Iraq, 2006.
[32] V. W. Samawi, "An Investigation in to the use of Neural Networks in Texture Classification", (Ph.D.) thesis, College of science, Al-Nahrain University, Iraq, 1999.
[33] M. Sonk, V. Hlavac and R. Boyle, "Image Processing Analysis and Machine Vision", Thomson, 3rd edition, 2008.
[34] S. Theodoridis and K. Koutroumbas, "Pattern Recognition", 2nd edition, Elsevier, 2003.
[35] M. Tuceryan and A. K. Jain, "Texture Analysis", Chapter 2.1 in Handbook of Pattern Recognition and Computer Vision, 2nd Edition, World Scientific, 1998.
[36] A. Webb, "Statistical Pattern Recognition", 2nd edition, Wiley, 2002.
[37] L. Y. Wei, "Texture Synthesis by Fixed Neighborhood Searching", (Ph.D.) thesis, Electrical Engineering, Stanford University, Li-Yi-Wei, 2001.
[38] D. Zhou, "Texture Analysis and Synthesis using a Generic Markov-Gibbs Image Model", (Ph.D.) thesis, Computer Science, Auckland University, 2006.