Medical images classification for skin cancer diagnosis based on combined texture and fractal analysis

RADU DOBRESCU, MATEI DOBRESCU, STEFAN MOCANU, DAN POPESCU "Politehnica" University of Bucharest, Faculty of Control and Computers, 313 Splaiul Independentei, Bucharest ROMANIA radud@isis.pub.ro

Abstract. Fractal and texture analysis are computer techniques which can discriminate between the shapes of benign and malignant tumors. The goal of present paper is to describe a method and an algorithm for automatic detection of malignancy of skin lesions which is based on both local fractal features (local fractal dimension) and texture features which derives from the medium co-occurrence matrices (contrast, energy, entropy, homogeneity). The global application was tested on a set of medical images obtained with a dermoscope and a digital camera, all from cases with known diagnostic. The experimental results confirm the efficiency of the proposed method.

Keywords: image processing, texture analysis, fractal analysis, box-counting algorithm, co-occurrence matrix, computer aided diagnosis, image classification.

1 Introduction

Medical imaging techniques produce images containing a lot of information about the anatomical structures being investigated, valuable for making correct diagnoses, choosing the most adequate therapy, surveying the evolution of the treatment and so on. The analysis of medical images has always been performed visually by physicians, but in the last two decades, a strong impulse has been given to develop automated systems capable of assisting physicians in this task, mainly because they have the desire of quantizing a number of anatomical and functional parameters, useful for diagnosis and therapy, which can be evaluated only qualitatively by human beings. However, the presence of noise and of masking structures, the variability of biological shapes and tissues, imaging system anisotropy etc. make the automated analysis of medical images a very hard task. The approach to overcome these problems has always been i) to simplify the objective of the analysis and ii) to exploit some kind of a priori information about the imaged structures. This information about the structures to be analyzed can be anatomical knowledge about their typical appearance (shape, gray levels) and position, or knowledge about statistical properties of the gray levels of the tissues included in those structures. Depending upon the particular diagnostic task, the extracted features for classification process have different properties: morphological properties, colour properties, fractal properties or textural properties.

The textural properties computed are closely related to the application domain to be used. Thus, Sutton and Hall [1] discussed the classification of pulmonary diseases by texture features. They used three types of texture features to distinguish normal lungs from diseased lungs: a directional contrast measure, an isotropic contrast measure, and a Fourier domain energy spectrum. Harms, Gunzer and Aus [2] used a combination between texture features (micro-edges and size of textons) and colour features to diagnose malignancy in blood cells. In [3], the authors used textural features to estimate tissue scattering parameters in ultrasound images. Recently, fractal geometry was used to investigate the relationship between the complexity of the epithelial/connective texture interface (as determined by fractal dimension) and the malignancy of the gastric tumor [4]. For breast cancer, the mammography is the most effective method for early diagnosis, which is crucial for the treatment methods. A considerable amount of research work was focused on the automated detection of mammary micro-calcifications [5], [6]. In this paper a new approach to the analysis of dermatological medical images based on both texture and fractal features is described. It differs from the other ones for two reasons: first, the analysis of medical images is here considered as a cognitive process by which the information is

transformed into explicit descriptions of the shape, the dimensions, etc.; second, the adopted processing procedure based on the determination of an aggregate feature vector seems to be the right strategy to overcome the reliability and speed problems encountered in the analysis of medical images.

2 A basic system for automated imagistic cancer diagnosis

The general configuration of a system for automated diagnosis in cancer, based on fractal and texture features is presented in Fig. 1.

The system has a graphic user interface that allows to be easily used in clinical environments.

The significances of notation in Fig.1 are the following:

I – interface; AL – auto-level, stretching of the minimum and maximum grey levels of the image to the maximum interval (0-255); NR – noise rejection; B – segmentation block; ED – edge detection block; M – memory for different preprocessed images (grey level image after noise rejection operation, segmented images with manifold thresholds, contour images); TFC – texture feature computation; FDC – computation of fractal dimensions and some derived parameters; DB – decision block.



Fig.1. Configuration of a system based on digital image processing for automated diagnosis of tumors.

After image acquisition, a primary image processing (noise rejection, segmentation - in order to obtain binary image - and contour extraction) is necessary. The contour is reduced to a single pixel width in order to avoid the effects of the thickness of the contour [15]. The algorithms for image primary processing are the following (nonlinear type):

- noise rejection (median filter); this filter reduces white noise without affecting the region contours:

 $b_2' = Me \{a_2, b_1, b_2, b_3, c_2\};$

- segmentation in order to detach the shapes of the interesting masses; it is based on binary images with variable threshold :

 $b_2' = 0$ if $b_2 < P$, and $b_2' = 1$ if $b_2 \ge P$, where P is a threshold chose by the program;

- contour extraction; this is a simple data compression without apparent loss of information in classification process:

$$b'_2 = b_2 \wedge \left(\overline{a}_2 \vee \overline{b}_1 \vee \overline{b}_3 \vee \overline{c}_2\right)$$

All these operators are local type and the elements are in a 3x3 neighbourhood of the central pixel b_2 (Fig.2). The result of each operator is noted by b_2 '.



Fig.2. Local filtering cross type neighbourhood based.

The selected features must be invariant to translation, rotation and scale transformations and must provide pathology information.

3 Features extraction for image analysis

3.1 Statistical methods to texture analysis

The statistical approach to texture analysis is more useful than the structural one. The simplest statistical features, like the mean μ and standard deviation σ^2 , can be computed indirectly in terms of the image histogram *h*. Thus,

$$\mu = \frac{1}{N} \sum_{i=1}^{K} x_i h(x_i)$$

$$N = \sum_{i=1}^{K} h(x_i)$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{K} (x_i - \mu)^2 h(x_i)$$

where $N = n_1 \times n_2$ is the image dimension, and *K* is the number of grey levels.

The shape of an image histogram provides many clues to characterize the image, but sometimes it is inadequate to discriminate textures (it is not possible to indicate local intensity differences) [16]. Another simple statistic features is the edge density per unit of area, $Den_e[17]$, The density of edges, detected by a local binary edge detector, can be used to distinguish between fine and coarse texture, Den_e can be evaluated by the ratio between the pixel number of extracted edges N_e (which must be tinned – one pixel thickness) and image area A (pixel number of image region):

$$Den_e = N_e / A$$

In order to characterize textured images, connected pixels must be analyzed. For this reason, correlation function R, difference image I_d in certain direction $d = (\Delta x, \Delta y)$, and co-occurrence matrices C_d , must be considered:

$$R(x, y) = \frac{\sum_{u=0}^{n_1-1} \sum_{v=0}^{n_2-1} I(u, v) I(u + x, v + y)}{\sum_{u=0}^{n_1-1} \sum_{v=0}^{n_2-1} I^2(u, v)}$$
$$I_d(x, y) = I(x, y) - I(x + \Delta x, y + \Delta y)$$

From the histogram of the difference image h_d , one can extract the mean μ_d and standard deviation σ_d^2 :

$$\mu_{d} = \frac{1}{N} \sum_{i=1}^{K} x_{i} h_{d}(x_{i})$$

$$\sigma_{d}^{2} = \frac{1}{N} \sum_{i=1}^{K} (x_{i} - \mu_{d})^{2} h_{d}(x_{i})$$

The most powerful statistical method for textured image analysis is based on features extracted from the Grey-Level Co-occurrence Matrix (GLCM), proposed by Haralick [7]. GLCM is a second order statistical measure of image variation and it gives the joint probability of occurrence of grey levels of two pixels, separated spatially by a fixed vector distance $d = (\Delta x, \Delta y)$. The co-occurrence matrix is able to capture the spatial dependence. Smooth texture gives co-occurrence matrix with high values along diagonals for small d. The range of grey level values within a given image determines the dimensions of a co-occurrence matrix. Thus, 4 bits grey level images give 16x16 co-occurrence matrices. The elements of a co-occurrence matrix C_d depend upon displacement $d=(\Delta x, \Delta y)$:

$$C_{d}(i,j) = Card\{((x,y),(t,v))/I(x,y) = i, (t,v) = j, (x,y), (t,v) \in N \times N, (t,v) = (x + \Delta x, y + \Delta y)\}$$

From a co-occurrence matrix C_d one can draw out some important statistical features for texture classification. From these features, which have a good discriminating power, from the most utilized in this paper we refer to: contrast Con_d , energy Ene_d , entropy Ent_d , homogeneity Omo_d , and correlation Cor_d .

$$Con_{d} = \sum_{i=1}^{L} \sum_{j=1}^{L} (i - j)^{2} C_{d}(i, j)$$

$$Ene_{d} = \sum_{i=1}^{L} \sum_{j=1}^{L} C_{d}(i, j)^{2}$$

$$Ent_{d} = -\sum_{i=1}^{L} \sum_{j=1}^{L} C_{d}(i, j) \log(C_{d}(i, j))$$

$$Omo_{d} = \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{C_{d}(i, j)}{1 + |i - j|}$$

$$Cor_{d} = \sum_{i} \sum_{j=1}^{L} \frac{(i - \mu i)(j - \mu j)P_{d}(i, j)}{\sigma_{i}\sigma_{j}}$$

One can observe some relations between texture aspect and statistical characteristics:

a. The contrast measures the coarseness of texture. The contrast is expected to be high in coarse texture, namely, if the grey levels of each pixel pair are dissimilar. Large values of contrast correspond to large local variation of the grey level.

b. The energy is a measure of the textural uniformity of the image. The energy has the highest value when the distribution of the grey levels is constant or periodic.

c. The entropy measures the degree of disorder or non-homogeneity, being in this sense opposite to the energy. Large values of entropy correspond to uniform GLCM.

d. Homogeneity measures the local homogeneity of a pixel pair. The homogeneity is expected to be large if the grey levels of each pixel pair are similar [18].

e. Correlation refers to pixel's neighbourhood influence on the whole image surface.

3.2. Fractal dimension as textural feature

Fractals are very useful and became popular in modeling these properties in image processing [8], [9]. Self-similarity across scales in fractal geometry is a crucial concept. A deterministic fractal is defined using this concept of self-similarity as follows. Given a bounded set A in a Euclidean n-space, the set A is said to be self-similar when A is the union of N distinct (non-overlapping) copies of itself, each of which has been scaled down by a ratio of r. The fractal dimension D is related to the number N and the ratio r as follows:

$$D = \frac{\log N(r)}{\log(1/r)}$$

There are a number of methods proposed for estimating the fractal dimension D. One method is box-counting algorithm that assumes determination of fractal dimension in function of the evolution of the object size in connection with the scale factor. There are many programs for counting the fractal dimension in different forms, but the most familiar algorithm is the box counting. A lot of specialized papers describe this algorithm. For the box counting basic algorithm, the image must be binary type. The method consists in splitting the image, successively, in equivalent squares with normalized size r = 1/2, 1/4, 1/8, ..., and computing every time the number N(r) of squares covered by the object image. The dividing process is limited by the image resolution. The fractal dimension can be obtained plotting lnN(r) for different values of ln(1/r), where r is the side length of covering boxes and calculating the slope of the resulting curve, which is approximated by a line. A linear regression is performed using the logarithmic coordinates x = log(1/r), y = logN(r). The regression slope a is used to determine the box counting fractal dimension *FD*, y = a x + b where:

$$a = FD = \frac{n \sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right) \left(\sum_{i=1}^{n} y_i\right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

The notation significances in the above equation are the following: $x_i = \log_2 (1/r_i)$, $y_i = \log_2 (N(r_i))$, n– number of partitions, i = 1,2,3,...,n – the function points in the graphical representation.

Towards evaluating the fractal dimension of a grey level image, we applied the box-counting algorithm to contours extracted from the binary images which are obtained by different thresholds [10]. Because the binary image (and also the fractal dimension) depends on the threshold, we used in our algorithm all the significant grey levels contained in the image. The fractal dimensions computed for every grey level will be represented into a graphic named fractal dimension spectrum. Thus, the algorithm calculates a mean fractal dimension *MFD* from individual values FD_j of some image contours detected from initial grey-level image:

$$MFD = \frac{1}{k} \sum_{j=1}^{k} FD_j$$

The algorithm, which was implemented in MATLAB, consists of the following steps:

a. Reading and converting the color image into a 256 grey levels image of same size;

b. Converting the 256 grey levels image to a binary level image using a fixed threshold T_i ;

c. Extraction of the image contour using 3x3 neighborhoods;

d. Computing of the fractal dimension FD_j , from the contour image, applying the box-counting algorithm;

e. Iteration of the steps 1-4, for j = 1, ..., k;

f. Determination of MFD.

In the color image case, we considered the natural decomposition by R, G, and B components [11]. For each component we calculated the mean fractal dimension by means of the grey level algorithm: *MFDR*, *MFDG*, and *MFDB*. These characteristics can be utilized as features in the texture classification process but, so far, we have not made enough tests in this direction. The same procedure can be applied to HSV decomposing.

4. Computer assisted medical image diagnosis

4.1 Combined texture and fractal image analysis

Fractal analysis is a mathematical and computer technique that quantifies complex shapes. Fractal analysis can discriminate between the shapes of benign and malignant tumors. Thus, a method to relieve the irregularity of the contour is to calculate and combine different forms of fractal dimension.

Experimentally, it was established that most of the information about the malignity of a tumor is contained in the contour of the tumor shape. Doubtful tumors are characterized by blurred contours which are changing by different threshold used to separate the tumor from background (image segmentation). The outline of each image was analyzed by estimating the global fractal dimension, the local fractal dimension and local connected fractal dimension [12]. For this purpose we used an original software package which was described in detail in [6].

In order to describe the heterogeneous nature of a tumor image, we may compute, for every single point in the image, a *local fractal dimension* (box-counting dimension, for example), limited to a neighbourhood of the central pixel. Thus, instead of a single value meant to characterize the whole image, we have a set of values, one for each point in the analyzed object. The values will be represented into a histogram in order to give emphasis to the distribution of the local irregularities of the image. We may consider that the global dimension (of the whole image) is the local dimension with the highest frequency.

4.2 Algorithms for fractal features computation used in computer diagnosis.

In brief, the fractal dimension of each image outline was measured by the box-counting algorithm and the local fractal dimension and local connected fractal dimension were estimated according to the algorithms published in [8] and [6].

a) Algorithm for box counting fractal dimension (A_1)

This algorithm evaluates a global fractal dimension for the object of interest in selected region and contains 5 steps:

S1. A selected region is considered (selection achieved by radiologist);

S2. With a threshold T, ranging from T_{min} to T_{max} , the grey level image is converted in a binary image and the contour is extracted.

S3. The region is scanned from left to right and from top to bottom with a side increasing square (from 5 pixels to 1/3 minimum side region). For each side dimension s, the number of squares N(s) which cover the contour is memorized.

S4. For each pair s_i and s_j of adjacent points in loglog curve, the local slope is determined. Then the most frequently slope is considered to be box counting dimension corresponding to threshold T.

S5. For tumor classification the mean of all box counting dimensions is considered.

b) Algorithm for local connected fractal dimension (A₂)

The images were analyzed by estimating their local mass scaling properties. The computer program measured the total number of pixels locally connected in a window of increasing size, centered at a point [4]. Locally connected relates to all pixels within the largest box used which belong to the cluster connected to the pixel where the box is centered. After getting over the steps S_1 and S_2 form A_I , the algorithm A_2 continues with the following steps: S2. Consider the current point P_1 :

S3. Consider the current point P;

S4. Mark all the points connected with P within a growing s-size window centered at P (s is under a fixed s_{max} value which may not be modified during the analysis of the whole image: 32, for example). Notice that every point has eight neighbour pixels (N, NW, V, SW, S, SE, E, and NE).

S5. Count every time how many points N(s) of the analyzed object are within the window;

S6. Using the least square method, compute the slope of the log-log curve composed by the $(\log(N(s),\log(s)))$ points.

c) Algorithm for local fractal dimension(A_3). This algorithm is similar with A_2 . After getting over the steps S₁ and S2 form A_1 , the algorithm A₂ continues with the following steps:

S3. Consider the current point P;

S4. Count every time how many points N(s) of the analyzed object are within the window;

S5. Using the least square method, compute the slope of the log-log curve composed by the $(\log(N(s),\log(s)))$ points.

4.3 FMedDiagnose application description

Although the advantages of using the local and local connected dimensions are obvious, they present an important disadvantage: the distribution of the local fractal dimensions and local connected fractal dimensions depends on the choice of the maximum window size. For better efficiency, one can combine fractal and texture analysis. In this aim was utilized a complex image processing software application, named FMedDiagnose. The facilities of this application are the following:

▶ Digital image processing (screening of a region for processing and saving this region in a file; graphically visualization of colours, brightness levels, and grey levels; noise rejection with mean based filter and median filter; image segmentation in binary form with a selected threshold and binary image visualization; edge extraction with local binary filter and contour visualization; selection of the image for high level processing: contour or shape)

Fractal analysis (fractal dimension global box determination: counting fractal dimension for each binary threshold and mean fractal dimension, local fractal dimension, and local connected fractal dimension; fractal dimension spectrum visualization; histogram of slope frequency visualization; selection of slope with maximum frequency)

► Texture analysis (co-occurrence matrices calculus; evaluation of texture features: energy, entropy, contrast, and correlation).

4.3.1 Software architecture

The application is a distributed client-server architecture, where both the server and the client run on the same machine at this early stage of the evolution. In the future, if the application will be applied on a largest area, the server will be moved from the local machine on the different one. As software technologies, in front-end (client) was utilized AIR (Adobe Integrated Runtime) and in back-end (server) PHP and Matlab. The knowledge base is a MySQL database.

The information flow client-server-client is represented in fig.3. By an user-friendly interface, the user sent requests to the PHP files on the server. Usually the server runs executable files compiled in Matlab environment. The server response is delivered to the AIR client in XML format, to be easy processed by dedicated classes.



Fig.3. Client-server information flow.

In Table 1 one can observe the parameters of the sendToServer function . The significance of the parameters is:

- *action_* - string of characters for the identification of the request type

- adData_ - the object that transmits input data

- *callbackFunction* - o call function which is activate when the answer was send back to the client.

Table 1 – sendToServer function

<pre>private function sendToServer(action_: String,</pre>					
<pre>adData_: Object, callbackFunction_: Function): void {</pre>					
var data : Object = new Object();					
$data = adData_;$					
data.action = action_;					
initMyService()					
var call : AsyncToken = httpService.send(data);					
call.param =					
httpService.request;					
call.handler =					
callbackFunction_;					
}					

4.3.2 Folders and files

In the client structure one find *FMedDiagnose* which is the parent folder of the application. In fig.4. is presented the structure of client folders and files. It contains the folders *bin-debug* (the results of the compilation) and *src* (the project sources) and the files *FMedDiagnose.mxml* - the client's backbone- and *FMedDiagnose-app.xml*.

The folder where the result of the compilation is obtained is *bin-debug*. It contains two files – *FMedDiagnose.swf*, the main result of the application which is loaded by FlashPlayer when the application is launched, and *FMedDiagnoseapp.xml*, a file which is utilized in FlashPlayer configuration for running the *src* application. The *src* folder contains all the classes used in the application programs. It has the following structure:

-the folder *classes* contains Actionscript classes, without user graphic interface;

- the folder *components* – contains *.*mxml* – classes with interface;

- the folder *connect* – contains the *DatabaseConnect.as* class which assure the connection with the server.



Fig.4. Structure of client folders and files

In the server structure the important files are:

- *DatabaseConnect.php*, which assure the connection with the database BD and the launch in execution of Matlab scripts;

- *functions.php*, a utilitarian which performs two functions: the connection function which grants the access at the BD and the parsing function which allow separating the access to the BD and the effective operations in BD;

- *bcResults* is the folder which receives the results furnished by running the file *BoxCounting.exe*. The following two entities are created in this folder: i) the text file *BoxCountingResults.txt* which stores the necessary data sets to plot the log-log diagram and also the fractal dimension of the analyzed image computed with box-counting; ii) the file *contur.jpg* – the graphic aspect of the binarized contour of the analyzed image (the threshold for binarization is obtained from the user's graphical interface);

- *coResults*, a folder that receives the results furnished by running the file *coMatrix.exe*, which produces the following entities: the text file *cooccurenceMatrix.txt* which stores the elements of the co-occurrence matrix and the text file *cooccurenceResults.txt* which stores the computed statistical features

- *exes*, a folder which contains two executable files *BoxCounting.exe* (computes the box-counting dimension of an input image) and *coMatrix.exe* (computes the co-occurrence matrix and the statistical features) and a batch file, *close.bat* which is called by both *BoxCounting.exe* and *coMatrix.exe* to stop the launched process at the end of its execution.

In fig. 5 one can observe the structure of folders on the main page of the server application.



Index of /FMedDiagnose_Server

🔟 Most Visited 📋 Customize Links 💐 Free Hotmail 📋 Windows Marketplace 📶 Window

٠	Parent Directory
	100.0

- DatabaseConnect.php
- <u>bcResults/</u>
- <u>coResults/</u>
- <u>crossdomain.xml</u>
- <u>exes/</u>
- functions.php
- images/
- <u>utils/</u>

Fig.5. Structure of server folders and files

In Table 2 is shown in detail the structure of the *DatabaseConnect.php* file.

Table	2.	Main	components	of	the		
DatabaseConnect.php file.							

echo " <data>";</data>			
\$action = \$ POST['action']:			
switch(\$action){			
case 'addNewAnalysisResult' : {			
addNewAnalysisResult(\$ POST):			
echo			
" <answer>addNewAnalysisResult</answer> ".			
hreak.			
}			
case 'retriveMedicalReports' · {			
case reinvelvieurcarkeports . {			
retriveMedicalReports(\$ POST)			
echo			
cciio			
<aliswei>ietrieveivieulearkepoits/aliswei>,</aliswei>			
bicak,			
∫ case 'getClassDeferenceDeports' : [
getClassReferenceReports(),			
callswei-getelasskeiereneekepolis/allswei/,			
Dieak,			
}			
case updateReport : {			
updateReport(\$_POST);			
ecno			
" <answer>updateReport</answer> ";			
break;			
}			
case 'connectToMatlab' : {			
connectToMatlab(\$_POST);			
break;			
}			
}			
echo "";			

4.3.3 Saving and storing of computed results

The saving of results is performed in a separate screen, after the complete analysis by both fractal and texture methods. In fig. 6 is illustrated an example of results representation. On the left side, there are two windows, one containing the original dermatographic image, the other some medical remarks (if necessary). On the right side, there are three windows. On the top, identification data of the patient. The red arrow points the omission of the patient's family name. In the middle, there are the results of the fractal analysis (the fractal dimension for a certain binarization threshold). In the bottom left are shown the results of the textural analysis, in order : image resolution, contrast, correlation, energy, homogeneity and entropy. The successful end of the saving process is signaled by a modal alert (the red arrow in the upper right corner).

Rezultate



Fig.6. Screenshot of the saving module results

The user can save and store the analysis report and to use them either as reference for comparison with other classified medical images, or for the case history of a medical patient (anamnesis). The module which performs the comparison between the results of a current analysis with previous results stored in the image database was implemented with two objects from the Analysis class: the first object is the current analysis, the second is the comparison analysis. Fig. 7 represents a page of the comparison module results. The interface is split in three zones: on the left side are the results of the current analysis, in the central zone is loaded a specific analysis from the previous executed and on the right side is the list of all the previous results.

Comparare rezultate



Fig.7. Screenshot of the comparison module results

5 Image comparison and classification

The information is stored in a MySQL database, which is accessed with SQL requests as INSERT, UPDATE and SELECT, using PHP language.

The database has a simple structure, having an unique table titled *results*, with the following fields:

- *id* – the identification number of an analysis report al unui raport de analiză;

- *firstName* patient first name
- lastName patient family name
- type type of analysis: reference or result
- className affiliation class of the result

- *analysisDate* – date of the result insertion in the data base

- *image* – the unique path to the location on the server where the image is stored

- *binarThreshold* – the threshold used in the image processing stage

- fdValue - box-counting dimension

- *pixelOffset* – the range for computing the co-occurrence matrix

- energy - value of energy statistical feature

- *entropy* =– value of entropy statistical feature

- *homogeneity* – value of homogeneity statistical feature

- contrast – value of contrast statistical feature

- *correlation* – value of correlation statistical feature

- *comments*- brief comments on the analysis result.

The above presented fields and their type are presented in fig.8. One can observe the quasisimilarity between the structured information from the *Results* Table and those included on the *Analysis* class.

At this stage of research we dispose of a data base with about hundred medical images with skin lesions. For each record a medical diagnostic confirmed by biopsy is attached. Several tests were performed on this database. First, the images were classified in four classes, according to a metric which combine the distance between two images computed from the textural features and the fractal dimensions which must differ at most with 5%. The first two classes (1 and 2) are associated to benign lesions, having regular forms with well defined contour. Class 4 is associated with malignant lesions. Class 3 reports suspect anomalies, the images having a weak defined contour, the delimitation of the tumor from the healthy tissue being very difficult to obtain. The malignancy risk of the tumors in this class is 15-50% and the simple visual diagnostic is not recommended [13].

Field	Туре
<u>id</u>	int(10)
firstName	varchar(255)
lastName	varchar(255)
type	varchar(255)
className	varchar(255)
analysisDate	datetime
image	text
binarThreshold	int(3)
fdValue	float
pixelOffset	int(10)
energy	float
entropy	float
homogeneity	float
contrast	float
correlation	float
comments	text

Fig. 8. The fields of the Analysis class

Given a new image, we have to decide which images in the database are relevant to it, and we have to retrieve the most relevant ones as the results of the query. In our experiments we use a likelihood ratio approach which is a Gaussian classifier, which define two classes, namely the relevance class R and the irrelevance class I. Two traditional measures for retrieval performance were used: Precision Rate (PR) and Recall Rate (RR). Precision Rate PR=n/T is the percentage of retrieved images that are correct and Recall Rate RR=n/N is the percentage of correct images that are retrieved, where N represents the number of images to which the user think the query image Qis similar, T is the number of images which are returned by the system automatically and n is the number of returned images which are considered similar to Q by the users. Note that computation of these measures requires prior ground-truth of the database. Since our automatically generated ground-truths are not the ones required for precision and recall, these measures cannot be used directly. We use modified versions of them to evaluate the performance of our algorithm. To test the retrieval performance, we use the procedure proposed in [14]. Given an input query image of size *KxK*, images are retrieved in descending order of likelihood ratio or ascending order of distance for nearest neighbor rule. If the correct image is retrieved as one of the *k* best matches, it is considered a success. Average rank of the correct image is also computed. If *Q* represents a query image and $g_1, g_2, ..., g_n$ represent the images which are searched by the system, we define rank (g_i) (i=1,2, ...,n) as the sequence number of the images in the result set, and the average rank as: $AR = (1/n) \sum_{i=1}^{n} rank(g_i)$.

6 Conclusions

The paper describes a method to analyze and classify medical dermatographic images using for combined fractal and texture features. The system performs image processing operations (noise rejection, segmentation to binary images with variable threshold and contour extraction), statistical and fractal features extraction and image comparison and classification. The texture features (energy, entropy, contrast, homogeneity, and correlation) derive from the average grey level cooccurrence matrix and fractal features derive from fractal dimension (box counting dimension, local dimension, and local connected dimension). The most important feature is the local connected fractal dimension calculated for contours. In the box counting algorithm, the fractal dimension is considered for different binary thresholds, and finally a mean value of these dimensions is considered. On the other hand, the algorithms for local fractal dimension and local connected fractal dimension assume the histogram plotting of these local dimensions in all selected region pixels. .

To recover images from the medical database closest to a similar one a query-by-example schema is used. Given an image as query item, a function of the algorithm will extract its general and regionalized features and then compare them to the ones in the index files. If they are similar enough the corresponding image will be a result of the query. The decision to put one of the search result images before another is given by the amount of elements that make the image similar to the query item.

The experimental results for the test images database worked with excellent results, the errors in retrieving were very few or not at all. Also the time of processing was small and all of this pointed to the fact that all the algorithm's functions were implemented correctly. Our future plans include improvement of threshold level selection, development of a new or improved procedure for calculating R,G,B mean fractal dimensions. We also aim to improve the processing speed and efficiency by implementing all the algorithms and procedures on a parallel processing platform.

ACKNOWLEDGEMENTS

This work was partially supported by the Romanian Ministry of Education and Research under grants PN2 61-031/2007 and ID_1032/2007.

References:

[1] Sutton, R. and E. L. Hall, Texture Measures for Automatic Classification of Pulmonary Disease, *IEEE Transactions on Computers*, vol.C-21, 1992, pp. 667-676;

[2] Harms, H., U. Gunzer, and H. M. Aus (1996), Combined Local Color and Texture Analysis of Stained Cells, *Computer Vision, Graphics, and Image Processing*, vol.33, 1996, pp.364-376;

[3] Insana, M. F., R. F. Wagner, B. S. Garra, D. G. Brown, and T. H. Shawker, Analysis of Ultrasound Image Texture via Generalized Rician Statistics, *Optical Engineering*, vol.25, 1996, pp. 743-748;

[4] Vasilescu, C., A. Herlea, B. Ivanov, R. Dobrescu and F. Talos, A survey on differences between intestinal and diffuse type of gastric carcinoma, in : R. Dobrescu, C. Vasilescu (Eds) *Interdisciplinary Applications of Fractal and Chaos Theory*, Editura Academiei Romane, Bucuresti, 2004;

[5] Wang, T.C., Karayiannis, N.B., Detection of microcalcifications in digital mammograms using walvets, *IEEE Trans. On Medical Imaging*, vol.17, no.4, pp. 498-509, 1998.

[6] Crisan, A.D., *Image processing using fractal techniques*, Ph.D Thesis, Politehnica University Bucharest, 2005;

[7] Haralick, R.M., Shanmugam, K., & Dinstein, I. Textural Features for Image Classification. *IEEE Transactions on Systems, Man. and Cybernetics*, SMC3(6), 1073, pp. 610-621

[8] Mandelbrot, B.B. (1982). *Fractal Geometry of Nature*. New York: Freeman.

[9] Barnsley, M., *Fractals Everywhere*, Academic Press, 1988

[10] Popescu, D., and Dobrescu R. Plastic Surface Similarity Measurement Based on Textural and Fractal Features. *Revista de materiale plastice*, 45(2), 2008, pp. 158-163. [11] Popescu, D., Dobrescu, R., and Angelescu, N. Color Textures Discrimination of Land Images by Fractal Techniques. *Proceedings of the 4th WSEAS International Conference on REMOTE SENSING -REMOTE'08*, 2008, pp.51-56

[12] Landini, G., and Rippin, J.W., How important is tumor shape? Quantification of the epithelialconnective tissue interface in oral lesions using local connected fractal dimension analysis, *J. Pathol.*, 179, 1996, pp. 210-217

[13] R. Dobrescu, C. Vasilescu, L.Ichim – Fractal and scaling analysis in tumor growth evaluation, *WSEAS Transactions on Systems*, Issue 1, vol.6, January 2007, p. 102-108

[14] M. Dobrescu. *Distributed Image Processing Techniques for Multimedia Applications*, Ph.D. Thesis, Politehnica Univ. of Bucharest, 2005.

[15] Popescu, D., Dobrescu, R, Mocanu, S., Dedicated Primary Image Processors For Mobile Robots, *WSEAS Transactions on Systems*, Issue 8, Vol.5, August 2006, p. 1932-1939

[16] Wong, A., Bishop, W., Perceptually-adaptive Image Super-resolution using Statistical Methods, *WSEAS Transactions on Signal Processing*, Issue 1, Volume 3, 2007, pp.44-49

[17] Petrescu C. D., Stefanoiu D., Lupu C., Performance Evaluation of Motion Estimation in DCT Transform Domain, *WSEAS Transactions on Signal Processing*, Issue 6, Volume 4, 2008, pp.371-380

[18] Udroiu, I., Tache, I., Angelescu, N., Caciula, I., Methods of Measure and Analyse of Video Quality of the Image, *WSEAS Transactions on Signal Processing*, Issue 8, Volume 5, 2009, pp.283-291