Advances and Applications in Statistics
Volume 29, Number 1, 2012, Pages 1-31
Published Online: October 2012
Available online at http://pphmj.com/journals/adas.htm
Published by Pushpa Publishing House, Allahabad, INDIA

APPLICATION OF GENETIC ALGORITHM FOR CLASSIFICATION OF MEDICAL IMAGES

Asanao Shimokawa and Etsuo Miyaoka

Faculty of Science
Tokyo University of Science
1-3, Kagurazaka, Shinjuku-ku
Tokyo 162-8601, Japan

Abstract

Medical image processing and classification involves multiple processes that can be classified into the following general steps: preprocessing step, feature extraction step, feature selection step, and classification step. In this study, we examine the availability of medical image classification techniques that involve the processing and pattern recognition for gray images by CT scan of lung cancer patients with different grades of the disease. In the feature extraction step, features of interest were extracted from whole images as well as regions of interest by using statistical texture patterns and wavelet transformation. In the feature selection step, we used a genetic algorithm to search for a feature set that could be used for the effective classification of the images. However, in order to optimize the classification, weighting each feature helped decide the exclusion/ inclusion of features. Several genetic-algorithm-based methods for determining the optimal weight of the feature space have previously

2010 Mathematics Subject Classification: 62H30, 68T10.

Keywords and phrases: pattern classification, feature extraction, feature selection, genetic algorithm, weighting.

Received July 20, 2012

been proposed. In this article, we propose a new method for weighting each feature; the method involves the use of iterations of a genetic algorithm. The method was compared with another method.

1. Introduction

A digital monochrome image is obtained by sampling a continuous image and storing the discretized values in the form of a two-dimensional image matrix $f(x, y)$ with $x = 0, 1, ..., N_x - 1$ and $y = 0, 1, ..., N_y - 1$. Every element (x, y) is called a *pixel*, and its gray value can be determined by using $f(x, y)$. There are b quantization levels leading to $t = 2^b$ distinct gray levels. In this study, we used a resolution (b) of 8 bits/pixel.

Medical image processing and classification involves multiple processes. It consists of a preprocessing step, feature extraction step, feature selection step, and classification step. In the preprocessing step, in order to simplify the next step, unnecessary information in the image is removed and important information is emphasized through denoising, deblurring, edge detection, etc. In the subsequent feature extraction step, features are extracted from the preprocessed image. A 512×512 image has 262,144 pixels, and it is evident that we cannot use this raw information for classification. Therefore, we have to generate new features from the available image matrix $f(x, y)$, and the generated features should contain all the relevant information contained in the original image. The features acquired here are divided into four categories: nontransformed structural characteristics, transformed structural characteristics, structural descriptions, and graph descriptors (Ciaccio et al. [3]). In the feature selection step, less discriminatory features are removed. Algorithms such as exhaustive search, branch-and-bound, and sequential forward selection algorithms are used. Further, the number of features used in classification is reduced, and as a result, faster and more accurate classification is possible. Finally, in the classification step, the features obtained in the feature selection step are used for the classification of unclassified images. Techniques used this step are divided into supervised classifications and unsupervised classifications. Supervised classification

uses the data which can be used in advance as the knowledge of the classification, which are referred to as training data. On the other hand, unsupervised classification is performed without using training data. This method classifies the data given without any external criterion. Examples of supervised classification are the maximum-likelihood method, which estimates unknown parameters using a set of known feature vectors in each class, the Bayesian methods, which uses known a priori information, and the minimum distance classifiers, which classifies an input vector on the basis of its distance to the learned prototypes. On the other hand, examples of unsupervised classification techniques are *k*-means clustering and hierarchical clustering.

In this study, we focus on the extraction of texture features. A gray-level co-occurrence matrix (GLCM) was used for extracting texture features. GLCM-based statistics have been widely used to represent the global aspects of texture in images, but they are not sufficient to completely describe the texture, particularly the local aspects of the texture. So we used the wavelet-transformation-based method that Dhawan et al. [4] used. Wavelet theory has been widely used in a number of signal processing applications such as multiresolution signal processing, subband coding, and image and speech compression (Mallat [7]). Laine and Fan [6] stated that by using wavelet decomposition, the texture can be represented with all details. They successfully classified 25 natural textures without any error by using texture features acquired with Daubechies wavelets (D_6, D_{20}).

CT images used in this study contain parts neighboring the lung, such as fat and heart. To consider only the lung region, we attempted to remove the region of interest (ROI) from the whole image and to extract the texture features from it. First, binary image processing was used as a means to execute it. An image is composed of pixels, and each pixel has a discrete gray level $(y_1, y_2, ..., y_t)$. In binary image processing, pixels with a gray level higher than a threshold value y_k are set to 1 and the others are set to 0. The method used in our study for determining y_k is the Otsu method. This

method uses the between-class variance as the criterion for determining y_k (Otsu [11]). However, in this study, there were several images from which the ROI could not be removed by using only binary image processing. In such images, we attempted to remove the ROI by adding the edge-detected image to the binary image. An edge is a point such that the gray level of adjacent pixels changes rapidly. We used a Laplacian-of-Gaussian (LoG) filter and detected the zero-crossing point for edge detection. By convolving with the LoG filter, the noise is removed from the image and the sign of the edge is changed from plus to minus or minus to plus (Marr and Hildreth [8]).

A genetic algorithm (GA) was used for feature selection. The GA refers to a model introduced and investigated by Holland [5]. It is a computational model inspired by evolution. This algorithm encodes a potential solution to a specific problem on a simple chromosome-like data structure and applies recombination operators to these structures to preserve critical information. Although the GA is often viewed as a function optimizer, the range of problems to which it has been applied is quite broad (Whitley [16]). Siedlecki and Sklansky [14] showed that the GA is a powerful tool for feature selection when the number of initial feature sets is large. If the initial number of features is *L*, then the size of the entire search space is 2^L. This search space increases with *L*, and exploring the entire search space requires a lot of time, and occasionally, it is impossible. In the study of Siedlecki and Sklansky [14], when the initial number of features exceeds 20, the feature selection problem is treated as a large-scale problem, and comparative studies of the exhaustive search method, branch-and-bound method, and GA method were performed. The study showed that the GA method is a powerful tool for large-scale feature selection.

In this study, for the purpose of image classification, we used *k*-nearest neighbor classification or classification by nearest neighbor of the center of each class. There are several reasons for using nearest neighbor methods. One is their simplicity, which makes them easy to implement, and another is the goodness of classification performance for a wide range of real world data sets (Raymer et al. [13]). In the study of Punch et al. [12], a GA-based

approach that combines feature selection and data classification was proposed. They used a GA combined with a *k*-nearest neighbor algorithm to optimize classification; the optimization was achieved by determining an optimal feature weighting. The weights were chosen to have a value from 256 values between 0 and 10, and each weight was represented by using 8 bits/feature. Further research was carried out by Raymer et al. [13]. By using a GA, they proposed the addition of a step to the feature selection step to select the number of optimal nearest neighbors, *k*.

Here, we propose an approach for searching for the weights of features and examine its effectiveness. In this approach, the images given for learning are divided into two groups at random. The images in one group are used for training nearest neighbor classifier, while those in the other group are used for testing. The nearest neighbor method is used to perform supervising with the training images, and testing with the test images. By using the nearest neighbor method and a GA, an optimal feature set for classifying the test images is determined. Next, the images given for learning are again divided into two groups at random, and an optimal feature set is selected. By repeating this operation, the probability that each feature is chosen at the end of the feature selection is decided, and we use this probability directly as the weight in the feature space. The motivation to propose this iteration approach is that it is expected that the available images can be utilized more effectively. By performing cross-validation, we verified the classification accuracy of the proposed approach.

This paper is organized as follows. The methods used for feature extraction and selection — the GLCM, the Otsu method, the method involving an LoG filter, wavelet transformation, and the method involving a GA — are described in the next section. Further, the feature weighting method proposed in this study and experimental results obtained by cross-validation are presented in the same section. Conclusions are given in Section 3.

2. Methods and Setup

2.1. Data

In our analysis, we used a data set containing 39 CT images taken from different lung cancer patients. These included 19 images taken from patents with low-grade malignancy and 20 images taken from patients with high-grade malignancy. By performing medical images processing, we determined the percentage of images in which the malignant grade could be correctly identified. Each image had 512×512 pixels and the gray-level range was transformed into 8 bits (0-255).

2.2. Feature extraction

A digital monochrome image is obtained by sampling the corresponding continuous image function $I(x, y)$ and storing the discretized values in form of a two-dimensional image matrix $f(x, y)$, with $x = 0, 1, ..., N_x - 1$ and $y = 0, 1, ..., N_y - 1$; where N_x and N_y are the size of the image. Every element (y, x) is called a *pixel*, and each pixel have a gray level $(y_1, y_2, ..., y_t)$ that can be determined from the image matrix $f(x, y)$. y_t is given by quantization level b and set to y_{2^b}. Generally, the gray level takes an integer value in the range from zero to the number of gray levels.

Three types of texture features were extracted in this research: statistical texture features extracted by using the gray-level co-occurrence matrix (GLCM) of the entire image matrix, statistical texture features extracted by using the GLCM of the region of interest (ROI) in the image matrix, and texture features extracted by wavelet transform of the entire image matrix. What follows is an explanation of the methods used for extracting these features.

2.2.1. Features extracted by gray-level co-occurrence matrix

A GLCM is a $y_t \times y_t$ matrix whose elements represent the relationship between gray levels of the pixels of two points in the image matrix for a

fixed distance vector $\mathbf{d} = (\Delta_x, \Delta_y)$, where Δ_x and Δ_y are fixed values in the ranges $0, 1, ..., N_x - 1$ and $0, 1, ..., N_y - 1$, respectively. In other words, the GLCM represents the frequency of an arbitrary gray level combination in a fixed distance in the image matrix. Each element of the GLCM is represented as $G(y_q, y_r, \mathbf{d})$, where y_q and y_r indicate the arbitrary gray label combination and assume values $y_1, y_2, ..., y_t$.

Features obtained from the GLCM are widely used for the analysis of textures. Let $H(y_q, y_r, \mathbf{d})$ be an element of the matrix obtained by dividing the $G(y_q, y_r, \mathbf{d})$ by the sum of the all element of the GLCM. In this study, we extracted 10 features by using $H(y_q, y_r, \mathbf{d})$ as a reference (Dhawan et al. [4]). The definition of these features is as follows:

1. The contrast of $H(y_q, y_r, \mathbf{d})$ is defined as

$$\sum_{y_q=y_1}^{y_t} \sum_{y_r=y_1}^{y_t} \delta(y_q, y_r) H(y_q, y_r, \mathbf{d}),$$

where $\delta(y_q, y_r) = (y_q - y_r)^2$.

2. The correlation of $H(y_q, y_r, \mathbf{d})$ is defined as

$$\frac{\sum_{y_q=y_1}^{y_t} \sum_{y_r=y_1}^{y_t} y_q y_r H(y_q, y_r, \mathbf{d}) - \mu_{H_m(y_q, \mathbf{d})} \mu_{H_m(y_r, \mathbf{d})}}{\sigma_{H_m(y_q, \mathbf{d})} \sigma_{H_m(y_r, \mathbf{d})}},$$

where

$$H_m(y_q, \mathbf{d}) = \sum_{y_r=y_1}^{y_t} H(y_q, y_r, \mathbf{d})$$

and

$$H_m(y_r, \mathbf{d}) = \sum_{y_q=y_1}^{y_t} H(y_q, y_r, \mathbf{d})$$

are one-dimensional marginal distributions of $H(y_q, y_r, \mathbf{d})$. μ and σ represent the mean value and the standard deviation, respectively.

3. The energy of $H(y_q, y_r, \mathbf{d})$ is defined as

$$\sum_{y_q=y_1}^{y_t} \sum_{y_r=y_1}^{y_t} [H(y_q, y_r, \mathbf{d})]^2.$$

4. The homogeneity of $H(y_q, y_r, \mathbf{d})$ is defined as

$$\sum_{y_q=y_1}^{y_t} \sum_{y_r=y_1}^{y_t} \left[\frac{H(y_q, y_r, \mathbf{d})}{1+\delta(y_q, y_r)} \right] \text{ for } y_r \neq y_q.$$

5. The entropy of $H(y_q, y_r, \mathbf{d})$ is defined as

$$-\sum_{y_q=y_1}^{y_t} \sum_{y_r=y_1}^{y_t} [H(y_q, y_r, \mathbf{d})] \log[H(y_q, y_r, \mathbf{d})].$$

6. The mean of $H_m(y_q, \mathbf{d})$ is defined as

$$\sum_{y_q=y_1}^{y_t} y_q H_m(y_q, \mathbf{d}).$$

7. The deviation of $H_m(y_q, \mathbf{d})$ is defined as

$$\sqrt{\sum_{y_q=y_1}^{y_t} \left[y_q - \sum_{y_p=y_1}^{y_t} y_p H_m(y_q, \mathbf{d}) \right]^2 H_m(y_q, \mathbf{d})}.$$

The following three features were computed from the difference of $H(y_q, y_r, \mathbf{d})$ statistics. The difference of $H(y_q, y_r, \mathbf{d})$ represents the probability of occurrence of differences, $| y_q - y_r | = i$, in the gray level values of two pixels separated by a specific distance **d**. It is defined as

$$H_{diff}(i, \mathbf{d}) = \sum_{y_q = y_1 | \, y_q - y_r \, | = i}^{y_t} \; \sum_{y_r = y_1}^{y_t} H(y_q, y_r, \mathbf{d}).$$

8. The entropy of $H_{diff}(i, \mathbf{d})$ is defined as

$$-\sum_{i=i_1}^{i_t} H_{diff}(i, \mathbf{d}) \log H_{diff}(i, \mathbf{d}).$$

9. The energy of $H_{diff}(i, \mathbf{d})$ is defined as

$$\sum_{i=i_1}^{i_t} [H_{diff}(i, \mathbf{d})]^2.$$

10. The mean of $H_{diff}(i, \mathbf{d})$ is defined as

$$\sum_{i=i_1}^{i_t} i H_{diff}(i, \mathbf{d}).$$

To determine **d** for calculating the GLCM, we performed the calculations shown below, following Dhawan et al. [4]. In one direction, we computed the correlation coefficients of the 10 texture features over all combinations of the 10 distances by taking 2 distances at a time $(\mathbf{d} = (1, -1), (2, -2), ..., (10, -10))$. Thus, for each feature, a total of 45 different correlation coefficients were computed. Table 1 shows the averages and standard deviations of the correlation coefficients over all different combinations of the 10 distances. Similar calculations were carried out for the other directions. For one distance, we computed the correlation coefficients of 10 features over all combinations of the four directions by taking two directions at a time $(\mathbf{d} = (5, 0), (0, 5), (5, 5), (5, -5))$. In Table 2, we show the results of this computation.

As seen from the results, for each feature, a high average value and a low standard deviation value of the correlation coefficients are observed in the

case of both distance and direction. This observation implies that the GLCM is almost independent of the **d** value used for its determination. Therefore, we decided to use only one distance and one direction $(\mathbf{d} = (5, -5))$. By using $\mathbf{d} = (5, -5)$, we calculated the 10 texture features described above.

2.2.2. Extraction of the region of interest

In order to extract the ROI from the whole image, first, binary processing was performed by using only the Otsu method. By performing binary processing on the images, we aimed to distinguish parts with high gray levels (like fat-containing parts and the heart) from those with low gray levels (like the lungs and objects outside the body). The ROI was extracted by labelling the lung portion of the binary-processed images.

The Otsu method determines the threshold for binary processing as follows (Otsu [11]). Let $n_1, n_2, ..., n_t$ denote the number of pixels with gray levels $y_1, y_2, ..., y_t$ in the image matrix, respectively. Total number of pixels is given by $N = n_1 + n_2 + \cdots + n_t$. Then, the probability distribution of each gray level is given by

$$p_i = \frac{n_i}{N} \quad \text{for } i = 1, 2, ..., t.$$

Table 1. Averages and standard deviations of the correlation coefficients over all different combinations of the 10 distances

Features	Average of Corr.	Standard Deviation of Corr.
1. Contrast	0.9123	0.0952
2. Correlation	0.9094	0.0986
3. Energy	0.9976	0.0022
4. Homogeneity	0.9881	0.0141
5. Entropy	0.9923	0.0107
6. Mean	0.9999	0.0001

7. Deviation	0.9977	0.0025
8. Entropy (diff)	0.9785	0.0243
9. Angular second moment	0.9860	0.0166
10. Mean	0.9583	0.0450

Table 2. Averages and standard deviations of the correlation coefficients over all different combinations of the four directions

Features	Average of Corr.	Standard Deviation of Corr.
1. Contrast	0.9558	0.0160
2. Correlation	0.9526	0.0176
3. Energy	0.9983	0.0006
4. Homogeneity	0.9960	0.0017
5. Entropy	0.9979	0.0010
6. Mean	1.0000	0.0000
7. Deviation	0.9992	0.0006
8. Entropy (diff)	0.9890	0.0046
9. Angular second moment	0.9951	0.0023
10. Mean	0.9809	0.0072

Suppose that we dichotomize the pixels into two classes C_0 and C_1 by considering the threshold to correspond to gray level y_k. C_0 has pixels with gray levels $y_1, y_2, ..., y_k$, and C_1 has pixels with gray levels y_{k+1}, $y_{k+2}, ..., y_t$. Let σ_W^2, σ_B^2, and σ_T^2 denote the within-class variance, the between-class variance, and total variance of gray levels, respectively. Then, the following basic relation always holds:

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2.$$

The within-class variance and the between-class variance depend on the threshold y_k, but the total variance does not depend on it. Therefore, there is a trade-off between the within-class variance and the between-class variance in terms of y_k. Moreover the within-class variance is based on the second-order statistics, while the between-class variance is based on the first-order statistics. Therefore, it is simpler to use the between-class variance for the criterion measure in order to evaluate the goodness of the threshold y_k than the within-class variance. Accordingly, Otsu used the following discriminant criterion measure:

$$\eta(y_k) = \frac{\sigma_B^2(k)}{\sigma_T^2}. \tag{1}$$

The optimal threshold y_{k^*} is the value that maximizes this equation (1), i.e., the between-class variance σ_B^2. In order to find the optimal threshold, we calculate the following equation for all the gray levels in the image matrix:

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)(1-\omega(k))},$$

where $\mu_T = \sum_{i=1}^{t} y_i p_i$, $\omega(k) = \sum_{i=1}^{k} p_i$, and $\mu(k) = \sum_{i=1}^{k} y_i p_i$. The optimal threshold y_{k^*} satisfies the

$$\sigma_B^2(k^*) = \max_{1 \le k < t} \sigma_B^2(k).$$

However, there were several cases in which it was difficult to extract the ROI solely through binary processing. An example is shown in Figure 1. One reason for the difficulty is that if a tumor is near the lung wall, then there is no significant difference between gray levels of fat and the tumor. To overcome this problem, we used the edge-detection process involving on LoG filter. An edge in the image is a place where the gray level changes rapidly. By combining a binary image and the edge-detected image, we tried to extract the ROI.

An LoG filter performs smoothing with a Gaussian filter and quadratic differential processing with a Laplacian filter simultaneously when applied to an image. This filter is obtained as follows (Marr and Hildreth [8]). As mentioned at the beginning of this section, an image matrix $f(x, y)$ is obtained by sampling the corresponding continuous image function $I(x, y)$. The Laplacian of $I(x, y)$ is defined by

$$\nabla^2 I(x, y) = \frac{\partial^2}{\partial x^2} I(x, y) + \frac{\partial^2}{\partial y^2} I(x, y).$$

By taking the two-dimensional Taylor expansion of $I(x+1, y)$, $I(x-1, 0)$, $I(x, y+1)$ and $I(x, y-1)$ around (x, y) and adding the four resulting equations, the following approximate expression is obtained:

$$\nabla^2 I(x, y) \approx I(x+1, y) + I(x-1, y) + I(x, y+1) + I(x, y-1) - 4I(x, y).$$

By expressing this approximate equation in a filter form for using an image matrix $f(x, y)$, the Laplacian filter is obtained as

$$\text{Laplacian filter} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The output image matrix obtained by the convolution of an input image matrix with the Laplacian filter is the approximation of the Laplacian of the input image matrix. Because the convolution involves quadratic differential processing of the image matrix, the sign of the output image changes from positive to negative (or negative to positive) at the edge, which is a position where the slope of a plane increases or decreases rapidly. Therefore, edge detection can be done by detecting the zero-crossing points of the output image. A zero crossing point is a position where the signs of two adjoining pixels are opposite to each other.

However only the Laplacian filter is used and when noise is present in the input image, the result will emphasize the noise. In order to overcome this problem, the input image is smoothed using a Gaussian filter before

applying the Laplacian filter. The Gaussian filter is given by a two-dimensional Gaussian function $G(x, y)$ with mean 0 and variance σ^2. The Laplacian of a smoothed image $F(x, y)$ can be obtained by calculating the convolution of the Laplacian of this two-dimensional Gaussian function with the continuous image $I(x, y)$:

$$\nabla^2 F(x, y) = \nabla^2 G(x, y) * I(x, y).$$

The Laplacian of the two-dimensional Gaussian function, $\nabla^2 G(x, y)$, is given by

$$\nabla^2 G(x, y) = \frac{1}{2\pi\sigma^6}\left\{(x^2 + y^2 - 2\sigma^2)\exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)\right\}, \tag{2}$$

where σ^2 represents the degree of smoothing performed by the Gaussian function. If the value of σ^2 is large, then the output image $\nabla^2 F(x, y)$ represents the Laplacian of a blurred image that has been smoothed considerably. On the other hand, if the value of σ^2 is small, then output image represents the Laplacian of an input image that is almost identical to the output image. In this study, a filter of size 13×13 was constructed using the Laplacian of the two-dimensional Gaussian function (2), and the edge was detected from the zero-crossing points of the output image, which obtained by convolving the filter and $f(x, y)$. The value of σ^2 was set to 2 in this study.

Figure 1. Input image.

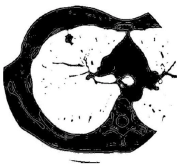


Figure 2. Combination of a binary image and the edge-detected image.

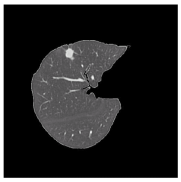


Figure 3. Extracted ROI the combination of the binary image and edge-detected image.

An image obtained by combining a binary image and the edge-detected image is shown in Figure 2. The ROI extracted by labelling the lung portion in Figure 2 is shown in Figure 3. It can be seen that the ROI has been extracted well in this case.

However, there were a few images in which the ROI could not be extracted by using this combination method. One reason for this was that the area of contact between the lung wall and tumor was very wide. That is, the lung boundaries could not be distinguished by using the edge-detection process. There were five such images out of the 39 images used in this study. For these five images, interpolation was carried out by using the cursor, and ROI images were obtained.

The value, except for the extracted ROI, was set to gray level 0, and the GLCM was constructed by using $\mathbf{d} = (5, -5)$. The first row and first column of the GLCM were removed, and the 10 features described in Subsection 2.2.1 were obtained.

2.2.3. Features extracted by wavelet transformation

In the analysis of an image given in a spatial domain that we have seen in general, the location of each pixel and the magnitude of each gray level is focused upon. On the other hand, in frequency domain analysis, by performing the Fourier transform of the image, it becomes possible to focus on the periodicity of each pixel in the image; however, spatial information is lost, and if the periodicity is changed at a specific location on the image, the image cannot be recognized. For the reasons mentioned above, we used the wavelet transformation for the purpose of the analysis in both domains. In the following, we describe the wavelet transformation discussed by Mallat [7] and Addison [1]. For simplicity, we first describe the wavelet transformation for a one-dimensional signal vector $f(x)$, where $x = 0, 1, ..., N_x - 1$, and then extend the description to a two-dimensional signal matrix (image matrix) $f(x, y)$.

The wavelet transformation of a signal $f(x)$ is its decomposition with a family of real orthonormal bases $\psi_{m,n}(x)$ obtained through the translation

and dilation of a function known as the mother wavelet $\psi(x)$:

$$\psi_{m,n}(x) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{x - n2^m}{2^m}\right), \tag{3}$$

where m and n are the translation and dilation parameters, respectively. Assume that the size of the signal vector N_x is sampled at a multiple of 2, $N_x = 2^M$, then the translation parameter is $m = 0, 1, ..., M$ and the dilation parameter is $n = 0, 1, ..., \left\lfloor \frac{N_x - 1}{2^m} \right\rfloor$, where $\lfloor x \rfloor = \max\{n \in \mathbf{Z} \mid n \le x\}$. The translation parameter represents the degree of decomposition by wavelet transformation. By convolution with the wavelet (3), the signal vector is decomposed, and the wavelet coefficients $T_{m,n}$ are obtained as

$$T_{m,n} = \sum_{x=0}^{N_x - 1} f(x)\psi_{m,n}(x).$$

The wavelet transformation is associated with a scaling function that has the same form as that of the wavelet function. The scaling function $\phi_{m,n}(x)$ is obtained by using a function known as the father scaling function $\phi(x)$:

$$\phi_{m,n}(x) = \frac{1}{\sqrt{2^m}}\phi\left(\frac{x - n2^m}{2^m}\right). \tag{4}$$

By the convolution of the signal with the scaling function (4), approximation coefficients $S_{m,n}$ are obtained:

$$S_{m,n} = \sum_{x=0}^{N_x - 1} f(x)\phi_{m,n}(x).$$

The decomposition of the signal vector $f(x)$ to a certain level m_0 is described as

$$f(x) = \sum_n S_{m_0,n}\phi_{m_0,n}(x) + \sum_{m}^{m_0}\sum_n T_{m,n}\psi_{m,n}(x).$$

The mother wavelet can be constructed using the father scaling function, which satisfies the two-scale equation represented as

$$\phi(x) = \sum_{k=1}^{N_k} c_k \phi(2x - k),$$

$$\psi(x) = \sum_{k=1}^{N_k} b_k \phi(2x - k),$$

where the coefficient c_k is called *scaling coefficient*. Coefficients c_k and b_k are related as

$$b_k = (-1)^k c_{N_k - 1 - k},$$

where N_k is the number of coefficients c_k. The coefficients c_k and N_k are given in advance for various wavelets. Using the scaling coefficients, the wavelet coefficients $T_{m+1,n}$ and the approximation coefficients $S_{m+1,n}$ are given by

$$T_{m+1,n} = \frac{1}{\sqrt{2}} \sum_k b_k S_{m,2n+k} \tag{5}$$

and

$$S_{m+1,n} = \frac{1}{\sqrt{2}} \sum_k c_k S_{m,2n+k}. \tag{6}$$

These relational expressions are known as the expressions of decomposition algorithm, and they indicate that if an approximation coefficient $S_{0,n}$ is given, then wavelet coefficients and approximation coefficients at each level can be obtained sequentially. In general, the sampled signal vector $f(x)$ is treated as the approximation coefficients $S_{m,n}$ at $m = 0$ and $n = 0, 1, ..., N_x - 1,$ and the signal vector is decomposed sequentially by the expressions (5) and (6).

The wavelet transformation of the two-dimensional signal matrix $f(x, y)$ is obtained by extending the one-dimensional decomposition. The two-dimensional decomposition algorithm can be written as

$$S_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} c_{k_1} c_{k_2} S_{m,(2n_1+k_1,\,2n_2+k_2)},$$

$$T^{h}_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} b_{k_1} c_{k_2} S_{m,(2n_1+k_1,\,2n_2+k_2)},$$

$$T^{v}_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} c_{k_1} b_{k_2} S_{m,(2n_1+k_1,\,2n_2+k_2)},$$

$$T^{d}_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} b_{k_1} b_{k_2} S_{m,(2n_1+k_1,\,2n_2+k_2)},$$

where k_1 and k_2 are indices of scaling coefficients and n_1 and n_2 are dilation parameters. The coefficients c_k and b_k act as a high-pass filter and low-pass filter for the signal, respectively. Therefore, the approximation coefficient and wavelet coefficients mentioned above have the following meaning: $S_{m,(n_1,n_2)}$ is obtained by horizontal and vertical low-pass filtering, $T^{h}_{m,(n_1,n_2)}$ is obtained by horizontal low-pass filtering and vertical high-pass filtering, $T^{v}_{m,(n_1,n_2)}$ is obtained by horizontal high-pass filtering and vertical low-pass filtering, and $T^{d}_{m,(n_1,n_2)}$ is obtained by horizontal and vertical high-pass filtering.

In this study, we have obtained one signal approximation coefficient and three wavelet coefficients from the image matrix by two-dimensional wavelet transformation (that is $m = 1$). From each of these coefficients we obtained the energy and entropy by using the method of Laine and Fan [6] and Dhawan et al. [4]:

1. The energy of coefficient is defined as

$$\frac{\sum_{n_1}\sum_{n_2} x^2_{m,n_1n_2}}{\text{length} * \text{breadth}},$$

where x_{m,n_1n_2} is the computed signal approximation coefficient or signal wavelet coefficient value of the n_1 th row and n_2 th column and length and breadth are the dimensions of each coefficient.

2. The entropy of coefficient is defined as

$$-\sum_{n_1}\sum_{n_2}\left[\frac{x^2_{m,n_1n_2}}{\text{norm}}\right]\log_{10}\left[\frac{x^2_{m,n_1n_2}}{\text{norm}}\right],$$

where $\text{norm} = \sum_{n_1}\sum_{n_2} x^2_{m,n_1n_2}$.

We used Daubechies wavelets D_6 and D_{20}, following Laine and Fan [6]. As described in the introduction section, these authors successfully classified 25 natural textures without any error by using these wavelets. For each of these wavelets, the signal approximation coefficients and horizontal, vertical, and diagonal signal wavelet coefficients of the decomposition level $m = 1$ were calculated. From each of these coefficients, we obtained the energy and entropy, and we used a total of 16 features for classification.

A total of 36 features were used in this study, and they were standardized to the range [0, 1].

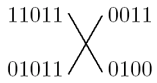
2.3. Feature selection

Some of the features that are not suitable for classification are present in the obtained features. The main purpose of feature selection is to reduce the number of features used in classification while maintaining an acceptable classification accuracy. Feature selection results in a better classification rate and facilitates faster analysis. The GA used for feature selection in this study is as follows (Whitley [16]).

2.3.1. Genetic algorithm

The first step in the implementation of the GA is to generate an initial population. Each member of this population will be represented as a binary string of length L, which corresponds to the problem encoded. Each string is referred to as a "chromosome." In most cases, the initial population is generated randomly, and each chromosome is then evaluated. The execution of the GA can be expressed as a two-stage process. In the first stage, it starts with the current population, and selection is applied to create an intermediate population. The selection is carried out on the basis of the evaluation of f_i for each chromosome a_i. In the second stage, crossover and mutation are applied to the intermediate population to create the next population. The process of proceeding from the current population to the next population is one generation in the GA. In the first generation, the current population is the initial population, which is generated randomly.

Crossover is applied by randomly selecting a pair of chromosomes from the intermediate population. In the case of crossover with a probability P_c, these chromosomes recombine to form two new chromosomes. The position of recombination (crossover point) is randomly chosen. For example, assume that the two chromosomes 110110011 and 010110100 are chosen from the intermediate population and that the crossover point is chosen to be between the fifth and sixth points in the chromosomes. Then, crossover is performed as follows:



Swapping the chromosomes between the two parents produces the offsprings 110110100 and 010110011. Generally, the probability of crossover, P_c, is high near bits with a value of 1. After the crossover operation, we apply a mutation operator. The mutation operation flips each bit in the population with some low probability P_m. For example, if the third and fifth points in the chromosome 110110100 are chosen, then mutation is carried out as

$$110110100 \rightarrow 111100100.$$

Typically, the mutation rate is applied with a probability less than 0.01.

After the process of selection, crossover, and mutation is complete, the next population will be evaluated. The process of evaluation, selection, recombination, and mutation corresponds to one generation in the execution of a GA.

An algorithmic description of the GA used in this study is given below:

1. Generate the initial population randomly for the chromosomes a_i.
2. The current population: $\Pi = \{a_i\}$, $i = 1, 2, ..., n$.
3. **For** $j \leftarrow 1$ **to** the number of generations M **do**
4. Initialize the intermediate population I and the offspring population O.
5. **For** $i \leftarrow 1$ **to** the number of chromosomes n **do**
6. Evaluate the chromosome a_i in the population Π.
7. Copy to the intermediate population I from Π on the basis of the evaluation of chromosomes $(I = \{a_i'\}, i = 1, 2, ..., n)$.
8. **For** $k \leftarrow 1$ **to** $n/2$ **do**
9. Choose the two parents a_q' and a_r' at random from M, and apply performed with the probability P_c $(a_q'', a_r'' = (a_q', a_r') \bigcup$ crossover $(a_q', a_r'))$.
10. Add a_q'' and a_r'' to the offspring population O.
11. The offspring population: $O = \{a_i''\}$, $i = 1, 2, ..., n$.
12. **For** $i \leftarrow 1$ **to** the number of chromosomes n **do**
13. **For** $b \leftarrow 1$ **to** the number of bits L **do**
14. Apply with mutation probability P_m to the bth bit from the chromosome a_i'' in O.

15. Replace population Π by the offspring population O.

16. Evaluate each chromosome in population Π of the last generation and get the best chromosome.

In the GA, M is 200, n is 100, P_c is 0.8, and P_m is 0.01.

In algorithm 7, we used the remainder stochastic selection method. Let f_i is the evaluation of the chromosome a_i. Then the value of $f_i / \bar{f}$, where $\bar{f}$ is the average evaluation of all chromosomes in the population of the generation j, is calculated in this method. For each chromosome a_i, the integer portion of the value of $f_i / \bar{f}$ indicates how many copies of this chromosome are directly placed in the intermediate population. In addition to this, with a probability corresponding to the fractional portion of $f_i / \bar{f}$, an additional copy is made in the intermediate population. For example, a chromosome with $f_i / \bar{f} = 1.56$ places 1 copy in the intermediate population and has a 0.56 chance of placing another one chromosome in the intermediate population. This operation is repeated until the number of chromosomes in the intermediate population is n.

2.3.2. Nearest neighbor method

The k-nearest neighbor method (k-nn) is a quite simple and widely used classification method. This method begins by putting the learning data points in the feature space represented by a chromosome a_i. Let N denote the number of learning images, and let $x_l = (x_{l1}, x_{l2}, ..., x_{lL})$ represent the coordinate of image l in the feature space $(l = 1, 2, ..., N)$. Furthermore, the chromosome a_i for each generation j $(j = 1, 2, ..., M)$ is represented as $\{f_{i1}, f_{i2}, ..., f_{iL}\}$, where f_{ib} is defined as

$$f_{ib} = \begin{cases} 1, & \text{when feature } b \text{ is included,} \\ 0, & \text{when feature } b \text{ is not included} \end{cases}$$

$(b = 1, 2, ..., L)$. Then, the classification rate for the chromosome a_i by k-nn

is defined as

$$CR_i = \frac{1}{N}\sum_{l=1}^{N} I_{il},$$

where I_{il} represents whether or not the classification has been successful at image *l*, and it is given as follows:

$$I_{il} = \begin{cases} 1, & \sum_{a=1}^{k} \alpha_{ila} > \frac{k}{2}, \\ 0, & \sum_{a=1}^{k} \alpha_{ila} < \frac{k}{2}, \end{cases}$$

where *k* is an odd number fixed in advance. α_{ila} is defined to be 1 when in the feature space represented by the chromosome a_i, the *a*th neighbor data point *m* of *l* is in the same class as the *l*; otherwise, is defined to be 0. The *a*th neighbor data point *m* is the point where the value of

$$\sqrt{\sum_{b=1}^{L} (f_{ib} \cdot x_{lb} - f_{ib} \cdot x_{mb})^2} \quad (m \neq l)$$

is the *a*th smaller for $m = 1, 2, ..., N$. We used $k = 1$ and 3 in this study.

Apart from this, the nearest neighbor method that involves the use of the center of each class was applied. In this method, we begin by the finding the center of each class for the learning images other than image *l*. That is to say, for a certain class C_1, each element c_{1b} of the center $c_1 = (c_{11}, c_{12}, ..., c_{1L})$ in the feature space represented by the chromosome a_i is defined as

$$c_{1b} = \frac{1}{N_1}\sum_{n=1}^{N_1} x_{nb} f_{ib},$$

where $n = 1, 2, ..., N_1$ represents the image of the member of C_1. Then, image *l* is classified in the same class as the nearest neighbor center point in the feature space at the Euclidean distance.

2.3.3. Evaluation function

We used the evaluation function that was used by Siedlecki et al. [14], to obtain the evaluation value for each chromosome in algorithm 7 of Subsection 2.3.1. This evaluation function uses a penalty function that increases exponentially when a chromosome has a error rate higher than a fixed threshold. This penalty function is defined as

$$p(e_i) = \frac{\exp((e_i - t)/m) - 1}{\exp(1) - 1},$$

where $e_i = 1 - CR_i$ is the error rate of a chromosome a_i, t is the threshold, and m is the scale factor. We note that $p(t) = 0$ and $p(t + m) = 1$ and that for higher values of the error rate, the penalty value quickly increases toward infinity. By adding the number of features used in a_i to the penalty value $p(e_i)$, the score $J(a_i)$ can be obtained:

$$J(a_i) = p(e_i) + N_i.$$

Using this score, the evaluation function is defined as

$$f_i = (1 + \varepsilon) \max_{a_i \in \Pi} J(a_i) - J(a_i),$$

where ε is a small positive constant. This ε assures that $\min f_i > 0$. In other words, even the chromosome with the least value of the evaluation function gets a chance to survive in the intermediate population. We set the parameters as $t = 0.01$, $m = 0.1$, and $\varepsilon = 0.05$.

2.4. Weighting method

In feature selection, we focus on deciding whether to include each feature, and the classification is carried out in the selected feature space. However, a higher classification rate is expected by weighting each feature and classifying in the feature space. In this subsection, we describe the feature weighting method used to optimize classification used by Punch et al. [12] and the method proposed in this work.

2.4.1. Weighting method used by Punch et al.

In Punch et al. [12], weights were chosen to have a value from among 256 values between 0 and 10, using 8 bits/feature to represent each weight. The string length in each chromosome was represented as $8 \times L$, and an optimal chromosome was searched for by using a GA. That is, the elements of the chromosome a_i were represented as $\{f_{i11}, f_{i12}, ..., f_{i18}, f_{i21}, f_{i22}, ..., f_{i28}, \cdots, f_{iL1}, f_{iL2}, ..., f_{iL8}\}$, where each element f_{ibc} was 0 or 1. For feature *b*, the weight was defined as

$$10 \times \frac{y_{ib}}{256},$$

where y_{ib} is the value obtained by translation from the binary representation $\{f_{ib1}, f_{ib2}, ..., f_{ib8}\}$ corresponding to feature *b* in the chromosome a_i to decimal representation. Using this weight, the *a*th neighbor data point *m* of *l* is the point where the value of

$$\sqrt{\sum_{b=1}^{L}\left(10 \times \frac{y_{ib}}{256} \cdot x_{lb} - 10 \times \frac{y_{ib}}{256} \cdot x_{mb}\right)^2} \quad (m \neq l)$$

is smaller in the *a*th for $m = 1, 2, ..., N$. They evaluated the each chromosome using *k*-nn and searched an optimal weighting by GA.

2.4.2. Weighting method proposed in this study

The proposed weighting method begins by randomly dividing the learning images into two sets: for training and testing. Let $l_1 = 1, 2, ..., N_1$ and $l_2 = 1, 2, ..., N_2$ denote the training images and test images, respectively $(N_1 + N_2 = N)$. Then, an optimal combination of features for classifying the test images by *k*-nn, which involves the use of the training images, is obtained by applying the GA. As discussed in Subsection 2.3.2, the classification rate for the chromosome a_i when *k*-nn is used is defined as

$$CR_i = \frac{1}{N_2}\sum_{l_2=1}^{N_2} I_{il_2},$$

where I_{il_2} is given by

$$I_{il_2} = \begin{cases} 1, & \sum_{a=1}^{k} \alpha_{il_2a} > \frac{k}{2}, \\ 0, & \sum_{a=1}^{k} \alpha_{il_2a} < \frac{k}{2}. \end{cases}$$

α_{il_2a} is defined to be 1 when in the feature space represented by the chromosome a_i, the ath neighbor training data point m of the testing data point l_2 is in the same class as l_2; otherwise, it is defined to be 0. The ath neighbor training data point m is the point where the value of

$$\sqrt{\sum_{b=1}^{L} (f_{ib} \cdot x_{l_2b} - f_{ib} \cdot x_{mb})^2}$$

is the ath smaller for $m = 1, 2, ..., N_1$. After an optimal combination of features is selected, again by the iterative process of randomly dividing the images into two groups and obtaining an optimal combination, the proportion of features selected from each group at the end by the GA is calculated. That is, if $\{f_{z1}, f_{z2}, ..., f_{zL}\}$ denote the elements of the chromosome corresponding to the maximum evaluation in the last generation $j = M$ at iteration z $(z = 1, 2, ..., Z)$, the proportion of each feature that are selected at the end by the GA is defined as

$$\left\{ \frac{1}{Z} \sum_{z=1}^{Z} f_{z1}, \frac{1}{Z} \sum_{z=1}^{Z} f_{z2}, ..., \frac{1}{Z} \sum_{z=1}^{Z} f_{zL} \right\}.$$

We directly used this proportion as the weight of each feature. The aim of using this iteration method is that a more robust and more reliable conclusion could be arrived at owing to the randomicity of the method. An algorithm of proposed method is described below:

1. **For** $z \leftarrow 1$ **to** the number of iterations Z **do**
2. The N_1 training images are selected from N learning images at random $(l_1 = 1, 2, ..., N_1)$.

3. The remaining learning images are treated as test images $(l_2 = 1, 2, ..., N_2)$.

4. L features are extracted from each image matrix during training and testing.

5. An optimal feature set $\{f_{z1}, f_{z2}, ..., f_{zL}\}$ for classifying the test images is selected by using the GA and k-nn.

6. The mean probability of each feature selected at the end by the GA is calculated:

$$\left(\left\{\frac{1}{Z}\sum_{z=1}^{Z} f_{z1}, \frac{1}{Z}\sum_{z=1}^{Z} f_{z2}, ..., \frac{1}{Z}\sum_{z=1}^{Z} f_{zL}\right\}\right).$$

In this method, Z was 500, the number of generations in the GA was 150, and the values of the other parameters like the probability of crossover and mutation were the same as those specified in Subsection 2.3.

2.5. Cross-validation

The classification accuracy of the malignancy was examined for the no-feature selection method (that is, when all features are used for classification), the method using only feature selection by GA, the with-weighting method used by Punch et al., and the with-weighting method proposed in this study. The accuracy was calculated by using the leave-one-out cross-validation method. Namely, the available 39 images were divided into 38 learning images and 1 verifying image. Then, an optimal feature set (or an optimal feature weight) for classification was searched for through feature extraction and feature selection by using the learning images. After that, the verifying image was classified using the resulting optimal feature set. This operation was performed on all 39 images, and the classification accuracy was calculated.

In Table 3, we show the results of the cross-validation when all features were used (that is 36 features). The results show the correctness of CT image classification for malignant grade performed through feature extraction and selection in this study. The abbreviations used in Table 3 are as follows: NFS

refers to the no-feature selection method, OFS refers to the method using only feature selection by a GA, WWPU refers to the with-weighting method used by Punch et al., and WWPR refers to the with-weighting method proposed in this study. In the $k = 1$ nearest neighbor method, the with-weighting methods have a higher classification rate than no-weighting methods. Additionally, there is no difference between the method used by Punch et al. and the method proposed in this study with respect to the classification rate. In the $k = 3$ nearest neighbor method, on the other hand, the methods of no-weighting showed slightly better results compared to the methods of with-weighting. In the nearest neighbor method, which uses the center of each class, the no-feature selection method shows a lower classification rate, about 72%. In contrast, the method proposed in this study shows a higher classification rate, about 90%, and this is the highest classification rate in this study.

Table 3. Results of cross-validation for about 36 features

	$k = 1$	$k = 3$	Center
NFS	82.05%	89.74%	71.79%
OFS	84.62%	89.74%	87.18%
WWPU	87.18%	87.18%	87.18%
WWPR	87.18%	87.18%	89.74%

Table 4. Results of cross-validation for about 35 features

	$k = 1$	Center
NFS	69.23%	71.79%
OFS	74.36%	82.05%
WWPU	82.05%	79.49%
WWPR	82.05%	76.92%

Furthermore, we focused on the number of features that are finally selected in the feature selection process by the GA. The average number of features that are finally selected for $k = 1$, $k = 3$, and the center of nearest neighbor method were 1.74, 1.35, and 1.13, respectively. From this result, it is clear that most of the 36 features extracted from each image were rarely selected by GA as a suitable. Moreover, the percentage of feature number 12 (the correlation of $H(y_q, y_r, \mathbf{d})$ in the ROI) selected at the end of the feature selection process was 100% in the case of each of the nearest neighbor methods. Thus, in the case of this classification, feature number 12 is a very good feature. On the other hand, considering a case where a such a good feature could not be obtained by the feature extraction step, we performed the cross-validation by using the 35 features left by ignoring feature number 12 by $k = 1$ and the center of nearest neighbor method. The results are shown in Table 4. Predictably, for all methods, the classification rate decreased compared to the case where all features were considered. In the $k = 1$ nearest neighbor method, the classification rate of the no-feature selection was reduced by about 13%. On the other hand, for the with-weighting methods, the decrease was only about 5%. That is, in the case of the $k = 1$ nearest neighbor method, the with-weighting methods are more effective and more robust for the classification. In the case of the center of nearest neighbor method, however, the percentage of reduction is higher for the with-weighting methods. Therefore with regard to this nearest neighbor method, the feature weighting show be performed more carefully. We think that the reason for the reduction in the classification rate is that the parameter setting of the GA for searching for the optimal weighting was not suitable. So we must continue to looking for an optimal method for appropriate parameter setting.

3. Conclusion

In this paper, we verified the correctness of CT image classification for malignant grade tumors through feature extraction and selection. In the feature extraction step, we focused on the texture features and extracted 36

features from each image by using a GLCM, wavelet transformation, binary processing, and edge detection. In particularly, in those features, it was found that the correlation of the elements of the GLCM in the ROI is very effective for the classification. In the feature selection step, we used a genetic algorithm to search for a feature set or a weighting that was effective for the classification of the images. In this process, we proposed a new method for feature weighting using the iteration of GA. We performed a comparison of our method with the method of no weighting and the method of weighting using GA, which is proposed in another paper.

The correctness of CT image classification for malignant grade tumors in this study indicate that the classification rate can be expected to reach nearly 90% when the optimal method is used. For the proposed method, some cases showed good results. However, there was also a case where the performance of our method was worse than that of the method of no weighting. In future studies, there is a need to examine whether the proposed method is good for all situations. We believe that the iteration method involving a GA has the potential to be improved for achieving better feature weighting.

References

[1] P. S. Addison, The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance, Institute of Physics Publishing, London, 2002.

[2] E. J. Ciaccio, S. M. Dunn and M. Akay, Biosignal pattern recognition and interpretation systems. Part 1. Fundamental concepts, IEEE Eng. Med. Biol. 12(3) (1993), 89-95.

[3] E. J. Ciaccio, S. M. Dunn and M. Akay, Biosignal pattern recognition and interpretation systems. Part 2. Methods for feature extraction and selection, IEEE Eng. Med. Biol. 12(4) (1993), 106-113.

[4] A. P. Dhawan, Y. Chitre and C. Kaiser-Bonasso, Analysis of mammographic microcalcifications using gray-level image structure features, IEEE Trans. Med. Imag. 15(3) (1996), 246-259.

[5] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[6] A. Laine and J. Fan, Texture classification by wavelet packet signatures, IEEE Trans. Pattern Anal. Machine Intell. 15(11) (1993), 1186-1191.

[7] S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Machine Intell. 11(7) (1989), 674-693.

[8] D. Marr and E. Hildreth, Theory of edge detection, Proc. R. Soc. Lond. B 207(1167) (1980), 187-217.

[9] A. Meyer-Base, Pattern Recognition for Medical Imaging, Academic Press, 2004.

[10] M. T. Miller, A. K. Jerebko, J. D. Malley and R. M. Summers, Feature Selection for Computer-Aided Polyp Detection Using Genetic Algorithms, Medical Imaging: Physiology and Function: Methods Systems, and Applications. Edited by Clough, Anne V. Clough, Amir A. Amini, Editors, Proc. SPIE 5031 (2003), 102-110.

[11] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9(1) (1979), 62-66.

[12] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland and R. Enbody, Further research on feature selection and classification using genetic algorithms, Proc. Int. Conf. Genetic Algorithms (1993), 557-564.

[13] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn and A. K. Jain, Dimensionality reduction using genetic algorithms, IEEE Trans. on Evolutionary Computation 4(2) (2000), 164-171.

[14] W. Siedlecki and J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recogn. Lett. 10(5) (1989), 335-347.

[15] M. Vasantha, V. S. Bharathi and R. Dhamodharan, Medical image feature, extraction, selection and classification, Internat. J. Eng. Sci. Tech. 2(6) (2010), 2071-2076.

[16] D. Whitley, A genetic algorithm tutorial, Statistics and Computing 4 (1994), 65-85.

[17] C. C. Wu, W. L. Lee, Y. C. Chen and K. S. Hsieh, A GA-based multiresolution feature selection for ultrasonic liver tissue characterization, Fourth International Conference on Innovative Computing, Information and Control (ICICIC), 7-9 December 2009, pp. 1542-1545.