Classification for Liver Diseases Based on Ultrasound Image Texture Features

Sendren Sheng-Dong Xu[1], Chien-Tien Su[2,*], Chun-Chao Chang[3,4], and Pham Quoc Phu[1]

[1]Graduate Institute of Automation and Control
National Taiwan University of Science and Technology, Taiwan
[2]School of Public Health, College of Public Health and Nutrition
Taipei Medical University, Taiwan
[3]Division of Gastroenterology and Hepatology, Department of Internal Medicine
Taipei Medical University Hospital, Taiwan
[4]Department of Internal Medicine, School of Medicine, College of Medicine
Taipei Medical University, Taiwan
*email: ctsu@tmu.edu.tw

Keywords: Classification; F-score; Gray-Level Co-occurrence Matrix (GLCM); Gray-Level Run-Length Matrix (GLRLM); Hepatocellular Carcinoma (HCC); Liver Cancer; Liver Abscess; Image Texture, Sequential Backward Selection (SBS); Sequential Forward Selection (SFS); Support Vector Machine (SVM); Ultrasound Image.

Abstract

This paper discusses the computer-aided (CAD) classification between Hepatocellular Carcinoma (HCC), i.e., the most common type of liver cancer, and Liver Abscess, based on ultrasound image texture features and Support Vector Machine (SVM) classifier. Among 79 cases of liver diseases, with 44 cases of HCC and 35 cases of liver abscess, this research extracts 96 features of Gray-Level Co-occurrence Matrix (GLCM) and Gray-Level Run-Length Matrix (GLRLM) from the region of interests (ROIs) in ultrasound images. Three feature selection models, i) Sequential Forward Selection, ii) Sequential Backward Selection, and iii) F-score, are adopted to determine the identification of these liver diseases. Finally, the developed system can classify HCC and liver abscess by SVM with the accuracy of 88.875%. The proposed methods can provide diagnostic assistance while distinguishing two kinds of liver diseases by using a CAD system.

I. Introduction

Liver diseases are ones of the most life-threatening causes worldwide. Among the different kinds of liver diseases, liver cancer and liver abscess, which have the high mortality rate, are the most dangerous ones. The distribution of liver cancer is irregular, with the high occurrence rate in countries of Eastern or South-eastern Asian, sub-Saharan Africa and Melanesia [1]. Liver abscess is not as popular as liver cancer. However if it is not detected in time and treated in the proper treatment protocol, the patients are uniformly fatal. The liver biopsy tests are often used to evaluate the disease. They permit the doctors to examine the liver, provide a lot of helpful information to give the high-accuracy predictions. In addition to those undeniable benefits, it could cause pain, infection to patients or other injuries to later treatments.

In order to reduce the unnecessary number of biopsy cases, other non-invasive methods for diagnosis have been applied popularly, especially using imaging technique such as Ultrasound (US) [2]-[3], Computed Tomography (CT) or Magnetic Resonance Imaging (MRI). Among those methods, the ultrasound imaging, with the unique advances such as no radiation, low-cost, easy operation, and non-invasive, is widely applied to visualizing the liver for clinical diagnoses. Therefore, it could provide the visual information for doctors to identify the state of disease. Nevertheless, the diagnoses are significantly affected by the quality of ultrasound images as well as the knowledge, experience of doctors. For inexperienced clinicians, it may be not easy to distinguish between liver cancers and liver abscess because of their similarity.

To overcome this obstacle, it will be helpful to develop a computer-aided diagnosis (CAD) system by using the image processing and pattern recognition techniques because the well-built system can help clinicians effectively and objectively recognize the distinction. Literature survey indicates that some scientists have studied about the liver-issue classification based on ultrasound images. For example, Nicolas *et al.* first exploited the textual features to discriminate between liver and spleen of normal humans [4]. Richard and Keen utilized the 5-by-5 Laws' feature mask and then apply the probabilistic relaxation algorithm for the segmentation [5]. Many textual features were used by Pavlopoulos for quantitative characterization of ultrasonic images [6]. Bleck*et al.* applied the random field model to distinguishing the four states of liver [7]. The models proposed by Gebbinck*et al.* and Kadahetal combined neural network and discriminant analysis to separate the different liver disease [8]-[9]. Pavlopoulos *et al.* improved their model by applying the fuzzy neural network to process the features [10]. Horng*et al.* evaluated the efficiency of the textual spectrum, the fractual dimension, the textual feature coding method, and gray level co-occurrence matrix in distinguishing cirrhosis, normal samples and hepatitis [11]. Yang *et al.* developed an algorithm for classifying cirrhotic and non-cirrhotic liver with the spleen-referenced approach [12].

As mentioned above, analyzing and classifing the images presenting organ lesion to differentiate its benign and malignant is the common purpose of many researchers, for example. So far in ultrasound imaging to explore the classification of Hepatocellular Carcinoma and liver abscess, we still have few clinically relevant studies, even though there are so many research about the analyzing the characteristics of Hepatocellular Carcinoma (HCC) (e.g., see [13]-[14]) or liver abscess (e.g., see [15]-[16]). Recently, many methods have been proposed to extract the features from the ultrasound images. For instance, the first and second order statistics are used (e.g., see [17]). Other approaches based on Wavelet Transform (e.g., see [18]), Gabor filter (e.g. see [19]), Monogenic Decomposition (e.g. see [20]), or Fractual Analysis (e.g. see [21]) were also proposed. There are various features could be extracted. However, the textual feature applied the most is Gray Level Co-occurrence Matrix (GLCM) (e.g., see [22]-[25]).

The main purpose of this research is to develop a reliable CAD system to provide the classification between two liver diseases, i) Hepatocellular Carcinoma (HCC), i.e., the most common type of liver cancer, and ii) liver abscess, based on Support Vector Machine (SVM) method and ultrasound image textual features. In this paper, we compare the results between the popular features GLCM with Gray Level Run Length Matrix (GLRLM). In this research, we totally have a large number of features, 96 features from each sample. If all of them are used to train a classifier, it not only costs too much time but also is hardly to achieve the high accuracy. To reduce the processing time as well as improve the accuracy, it is necessary to search the important features from the feature set. Then, the crucial features of the samples are used to train and test by SVM. There are some steps to achieve this goal. Firstly, in the ultrasound images, the liver lesions are marked by the experienced doctors and the region of interests (ROIs) are sampled inside the red boundary. Secondly, all features are extracted from the collected ROIs. Thirdly, several feature selection processes are carried out to optimize the feature set. Finally, the optimal feature sets are used to train and test by SVM.

II. Feature Extraction

Feature extraction is one of the most important stages in pattern recognition. It collects the input data for classifier, and thus can directly affect the performance of a CAD system. For example, with the same number of features, a better feature set could more exactly describe the special characteristics of each kind of liver diseases such that it can improve the diagnosis result. As mentioned in Section I, textual analysis of ultrasound (US) images is a very useful tool for liver diagnosis and two of the most effective method is Gray Level Co-occurrence Matrix (GLCM) and Gray Level Run-Length Matrix (GLRLM). In this research, we totally extract 96 features, including 52 features of GLCM and 44 features of GLRLM, for analysis.

A. Materials

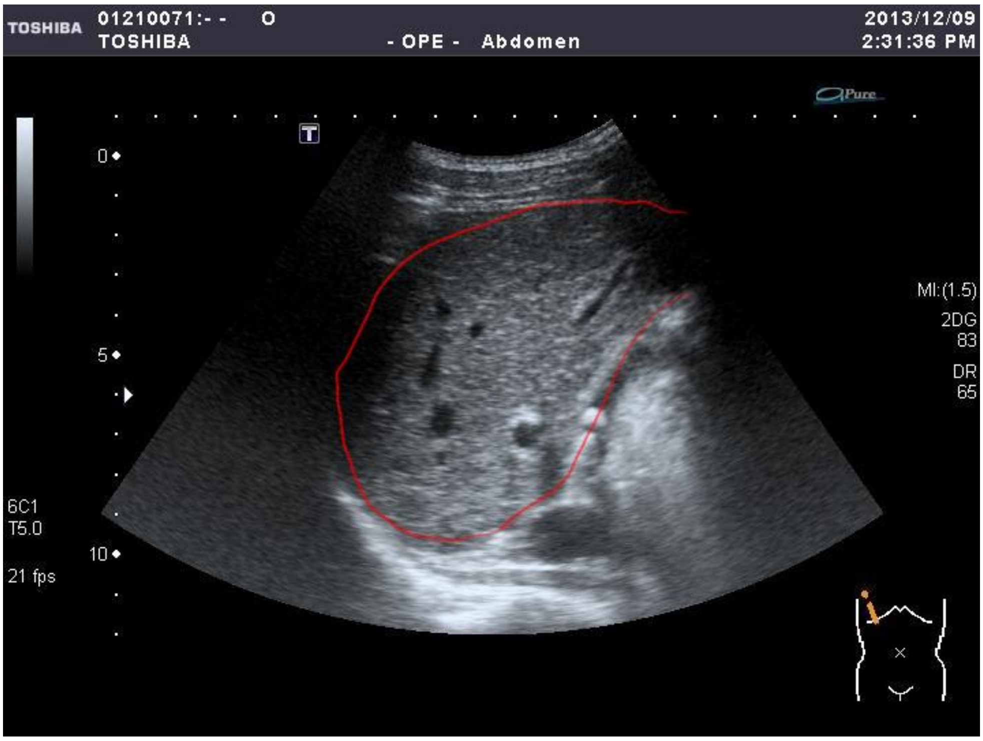


Figure 2-1. An example of the cropping process for the original US image

The original US images, supported by the Medical University Hospital in Taipei, were stored as JPG format. They were converted into 256-gray-scale BMP files by Matlab function for more convenient processing. All images are included in 79 cases of liver diseases, with 44 cases of Hepatocellular Carcinoma (HCC) and 35 cases of liver abscess. At first, the original images were marked by the experiences clinician and verified in clinical reality. Then, the 32*32 -pixel ROIs were selected inside the marked boundaries as presented in Fig. 2-1. In Fig. 2-2, the 32*32-pixel ROIs were sampled from the marked image. All samples were collected from the liver disease images for later procedure as Fig. 2-3. In this research, we sampled 400 ROIs of each kind of diseases for training and testing process.

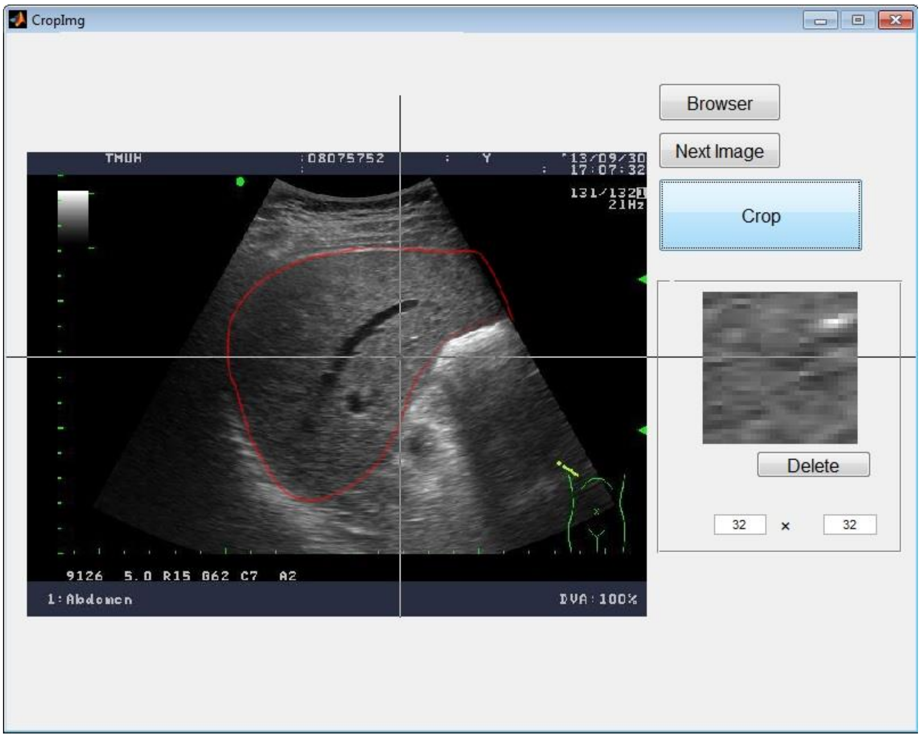


Figure 2-2. The cropping window

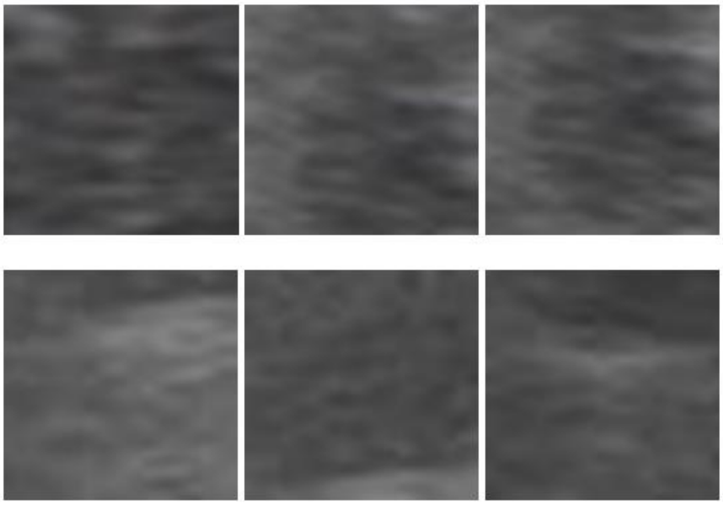


Figure 2-3. The ROIs taken by Matlab: The first row is the samples of HCC and the second row is the samples of liver abscess

B. Gray Level Co-occurrence Matrix (GLCM) and Haralick Beatures

In this step, ROIs are analyzed by GLCM, the most popular second order statistical features proposed by Haralick [26] in 1973. The Haralick-feature extraction could be done through two steps. In the first step, the co-occurrence matrix is calculated, and in the second, the texture features, which are very useful in a various kind of imaging application, especially biomedical images, are computed based on the co-occurrence matrix.

B-1. Gray Level Co-occurrence Matrix

The Gray Level Co-occurrence Matrix is a matrix that shows how often the different combinations of gray levels transpire in an image. It was widely applied to extracting the features, especially in the research of liver diseases (e.g., see [27]). In other words, it presents the relationship between two neighbour pixels. The whole procedure to extract the Haralick features is presented as Fig. 2-4 and the co-occurrence matrix can be calculated as the following equation (2.1):

$$C_{d,\theta}(i,j) = \sum_{x=1}^{N}\sum_{y=1}^{M}\begin{cases}1, \text{if } I(x,y) = i, I(x+\Delta x, y+\Delta y) = j\\ 0, \text{otherwise}\end{cases} \tag{2.1}$$

where $C_{d,\theta}(i,j)$ is the number of occurrences of the pair of gray levels i and j, with d, θ are the distance and angularity respectively; $I(x,y)$ is the intensity of a pixel at *xth* row and *yth* column in the image. We can see that, with the different pairs (d, θ), the image could be explored in different directions and distances. In this research, we choose the 4 directions $\theta = 0, 45^{o}, 90^{o}, 135^{o}$ with $d = 1$ (illustrated in Figure 2-5). Thus, the pair $(d = 1, \theta = 0)$ is the nearest horizontal pixel. Moreover, there are also co-occurrence matrixes for vertical $(\theta = 90^{o})$ and diagonal axes $(\theta = 45^{o}, 135^{o})$.

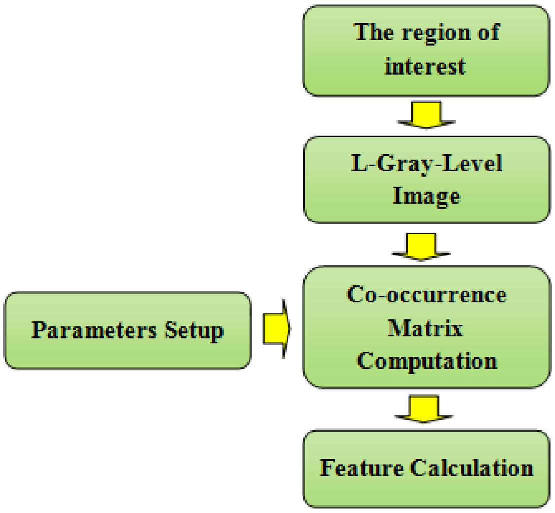


Figure 2-4. GLCM process

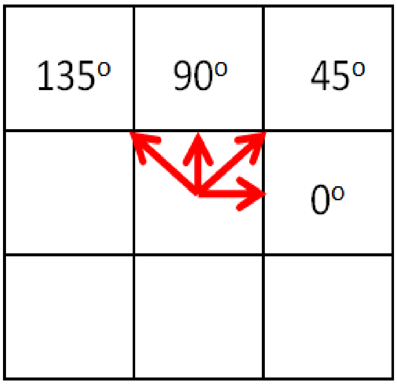


Figure 2-5. The directions of Gray Level Co-occurrence Matrix

For example in Figure 2-6, we calculate the Gray Level Co-occurrence Matrix of a 4*4-pixel image. In this case, 3 and 1 are nearby each other 2 times in the image, so the position (3;1) and (1;3) are filled with 2. Similarly, applying the same principle to all other pixels we could get the GLCM of the image.

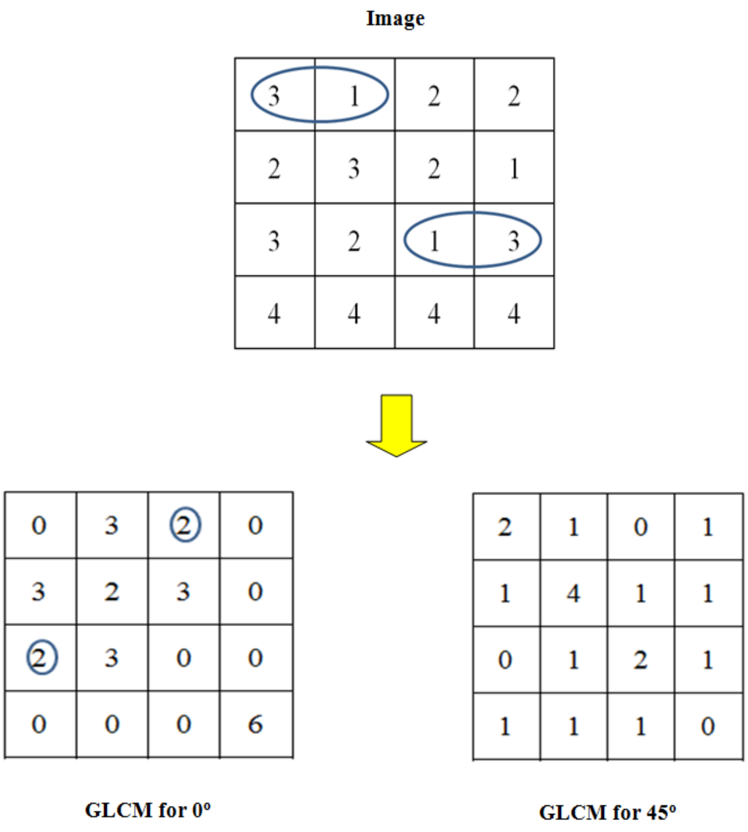
The ROI that we need to process is a 32*32 pixels with 256-gray level. Therefore, we will have a 256*256 matrix with totally 65,536 cells, but many cells are filled with zeros (because these combinations do not exist). This could lead to the bad approximation due to so many zeros cells). The solution for this problem is that the number of gray levels should be reduced, so this will decrease the number of zeros cells and the validity could be improved considerably. In this research, the ROIs were scaled to 16-gray-level images before computing GLCM. After that, it was normalized to be converted into probabilistic form by equation (2.2) and procedure is shown in Figure. 2-7.

$$P(i,j) = \frac{C(i,j)}{\sum_{i=0}^{l-1}\sum_{j=0}^{l-1} C(i,j)}. \qquad (2.2)$$

Therefore, we could get the GLCM as in equation (2.3):

$$G = \begin{bmatrix} P(0,0) & P(0,1) & \cdots & P(0,l-1) \\ P(1,0) & P(1,1) & \cdots & P(1,l-1) \\ \vdots & \vdots & \ddots & \vdots \\ P(l-1,0) & P(l-1,1) & \cdots & P(l-1,l-1) \end{bmatrix}. \qquad (2.3)$$

From co-occurrence matrix, the textual features proposed by Haralick could be calculated.



Image

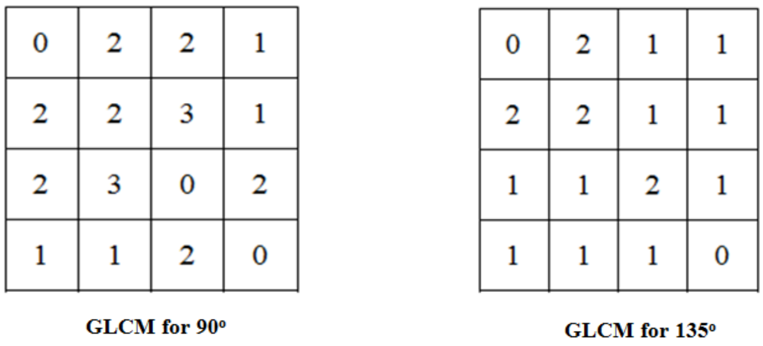
3	1	2	2
2	3	2	1
3	2	1	3
4	4	4	4

0	3	2	0
3	2	3	0
2	3	0	0
0	0	0	6

GLCM for 0°

2	1	0	1
1	4	1	1
0	1	2	1
1	1	1	0

GLCM for 45°



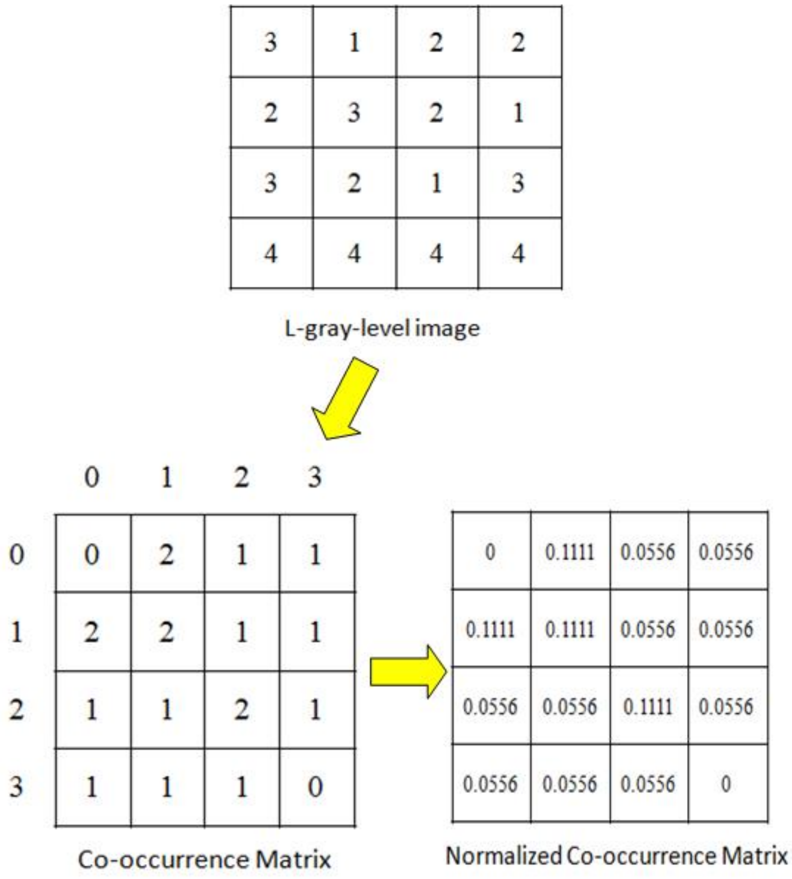
0	2	2	1
2	2	3	1
2	3	0	2
1	1	2	0

GLCM for 90°

0	2	1	1
2	2	1	1
1	1	2	1
1	1	1	0

GLCM for 135°

Figure 2-6. Extracting Gray Level Co-occurrence Matrix (GLCM) from an image



3	1	2	2
2	3	2	1
3	2	1	3
4	4	4	4

L-gray-level image

	0	1	2	3
0	0	2	1	1
1	2	2	1	1
2	1	1	2	1
3	1	1	1	0

Co-occurrence Matrix

0	0.1111	0.0556	0.0556
0.1111	0.1111	0.0556	0.0556
0.0556	0.0556	0.1111	0.0556
0.0556	0.0556	0.0556	0

Normalized Co-occurrence Matrix

Figure 2-7. Calculation for Gray Level Co-occurrence.

B-2. Haralick features

There are thirteen features which could be extracted from GLCM for an image. These features are presented as follows:

1. ***Energy feature:***

It is also called angular second moment (ASM) feature or uniformity. It describes the uniformity of an image. When the gray levels of pixels are similar, the energy value will be large.

$$f_1 = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} {P_{d,\theta}}^2(i,j) \tag{2.4}$$

2. ***Entropy feature:***

This concept comes from thermodynamics, which is a field of physics concern with heat, temperature and their relationship with energy and work. In our case, it could be considered as a chaotic or disordered quantity.

$$f_2 = -\sum_{i=0}^{l-1} \sum_{j=0}^{l-1} P_{d,\theta}(i,j).\log P_{d,\theta}(i,j) \tag{2.5}$$

3. ***Contrast feature:***

This measures the intensity variations between the pixels with the fixed direction and distance (d, θ). With the same gray level, the contrast value will be equal to 0. If $|i - j| = 1$, there is a little contrast so the weight is just 1. If $|i - j| = 2$, the contrast of gray level is higher, so the weight is bigger, 4. It means that the weight increase exponentially.

$$f_3 = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} |i - j|^2 . P_{d,\theta}(i,j) \tag{2.6}$$

4. ***Correlation feature:***

This feature describes the linear dependency of gray level in the Co-occurrence matrix. It shows how a centre pixel relates to each other.

$$f_4 = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} P_{d,\theta}(i,j) \frac{(i-\mu_x)(j-\mu_y)}{\sigma_x \sigma_y} \tag{2.7}$$

With μ_x, μ_y and σ_x, σ_y are the mean and standard deviations which could be calculated as following:

$$\sigma_x = \sqrt{\sum_{i=0}^{l-1} \sum_{j=0}^{l-1} P_{d,\theta}(i,j).(i - \mu_x)^2} \tag{2.8}$$

$$\sigma_y = \sqrt{\sum_{i=0}^{l-1} \sum_{j=0}^{l-1} P_{d,\theta}(i,j).(i - \mu_y)^2} \tag{2.9}$$

$$\mu_x = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} i.P_{d,\theta}(i,j) \tag{2.10}$$

$$\mu_y = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} j.P_{d,\theta}(i,j) \tag{2.11}$$

With the symmetrical Gray Level Co-occurrence Matrix, $\mu_x = \mu_y$and $\sigma_x = \sigma_y$.

5. Homogeneity feature:

This feature is also known as Inverse Difference Moment (IDM) feature which describes the local similarity of an image.

$$f_5 = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} \frac{1}{1+(i-j)^2} P_{d,\theta}(i,j) \tag{2.12}$$

The weight of IDM is the inverse of the weight of Contrast so it decreases away from the diagonal of GLCM. It means that the position which is nearer the GLCM diagonal will have the larger weight.

6. Sum average feature:

$$f_6 = \sum_{i=0}^{2(l-1)} i.P_{x+y}(i) \tag{2.13}$$

With:

$$P_{x+y}(k) = \sum_{i=0}^{l-1} \sum_{j=0}^{l-1} P_{d,\theta}(i,j), k = i + j \tag{2.14}$$

7. Sum entropy feature:

$$f_7 = -\sum_{i=0}^{2(l-1)} P_{x+y}(i).\log P_{x+y}(i) \tag{2.15}$$

8. Sum variance feature:

$$f_8 = \sum_{i=0}^{2(l-1)} (i - f_7)^2.P_{x+y}(i) \tag{2.16}$$

9. ***Difference average feature:***

$$f_9 = \sum_{i=0}^{(l-1)} i.P_{x-y}(i) \qquad (2.17)$$

With :

$$P_{x-y}(k) = \sum_{i=0}^{l-1}\sum_{j=0}^{l-1} P_{d,\theta}(i,j), \;\; k = |i-j| \qquad (2.18)$$

10. ***Difference variance feature:***

$$f_{10} = \sum_{i=0}^{(l-1)} (i - f_9)^2 . P_{x-y}(i) \qquad (2.19)$$

11. ***Difference entropy feature:***

$$f_{11} = -\sum_{i=0}^{(l-1)} P_{x-y}(i). \log P_{x-y}(i) \qquad (2.20)$$

12. ***Information measures of correlation feature 1:***

$$f_{12} = \frac{HXY - HXY1}{\max(HX,HY)} \qquad (2.21)$$

13. ***Information measures of correlation feature 2:***

$$f_{13} = (1 - \exp[-2(HXY2 - HXY)])^{1/2} \qquad (2.22)$$

With:

$$P_x(i) = \sum_{j=0}^{l-1} P_{d,\theta}(i,j) \qquad (2.23)$$

$$P_y(i) = \sum_{i=0}^{l-1} P_{d,\theta}(i,j) \qquad (2.24)$$

$$HX = -\sum_{i=0}^{l-1} P_x(i)\log(P_x(i)) \qquad (2.25)$$

$$HY = -\sum_{i=0}^{l-1} P_y(i)\log(P_y(i)) \qquad (2.26)$$

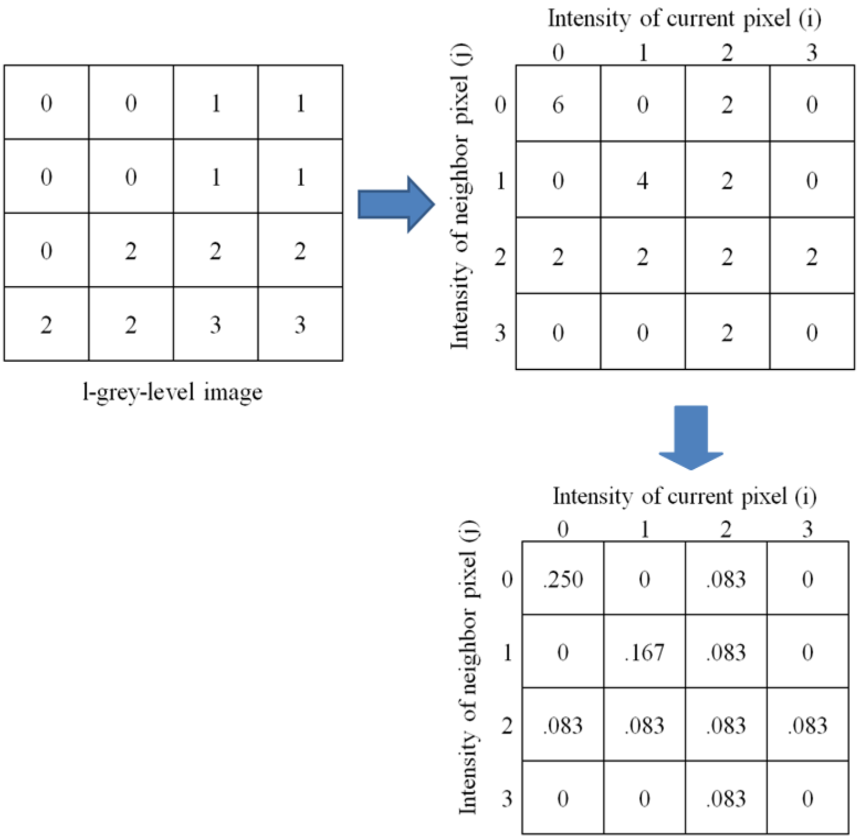
$$HXY = -\sum_{i=0}^{l-1}\sum_{j=0}^{l-1} P_{d,\theta}(i,j). \log(P_{d,\theta}(i,j)) \qquad (2.27)$$

$$HXY1 = -\sum_{i=0}^{l-1}\sum_{j=0}^{l-1} P_{d,\theta}(i,j). \log(P_x(i)P_y(j)) \qquad (2.28)$$

$$HXY2 = -\sum_{i=0}^{l-1}\sum_{j=0}^{l-1} P_x(i)P_y(j). \log(P_x(i)P_y(j)) \qquad (2.29)$$

The following example of 4x4 gray-scale image will illustrate the above equations as in Fig. 2-8.

After we get a co-occurrence matrix of the 4x4 gray-level image, 13 textual features can be calculated. The results of 13 textual features are shown in Table 2-1.



0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Intensity of current pixel (i)
Intensity of neighbor pixel (j)

	0	1	2	3
0	6	0	2	0
1	0	4	2	0
2	2	2	2	2
3	0	0	2	0

Intensity of current pixel (i)
Intensity of neighbor pixel (j)

	0	1	2	3
0	.250	0	.083	0
1	0	.167	.083	0
2	.083	.083	.083	.083
3	0	0	.083	0

Figure 2-8. GLCM example with $(\mathbf{d = 1, \theta = 90^{o}})$

Table 2-1. GLCM features of an image

	Textual Features	Value
1	Energy	0.1386
2	Entropy	2.0915
3	Contrast	0.9960
4	Correlation	0.5119
5	Homogeneity	0.7213
6	Sum average	2.3260
7	Sum variance	2.8829
8	Sum entropy	1.5155
9	Difference average	0.4980
10	Difference variance	0.5542
11	Difference entropy	1.0107
12	Information Measures of Correlation Feature 1	-0.3713
13	Information Measures of Correlation Feature 2	0.7840

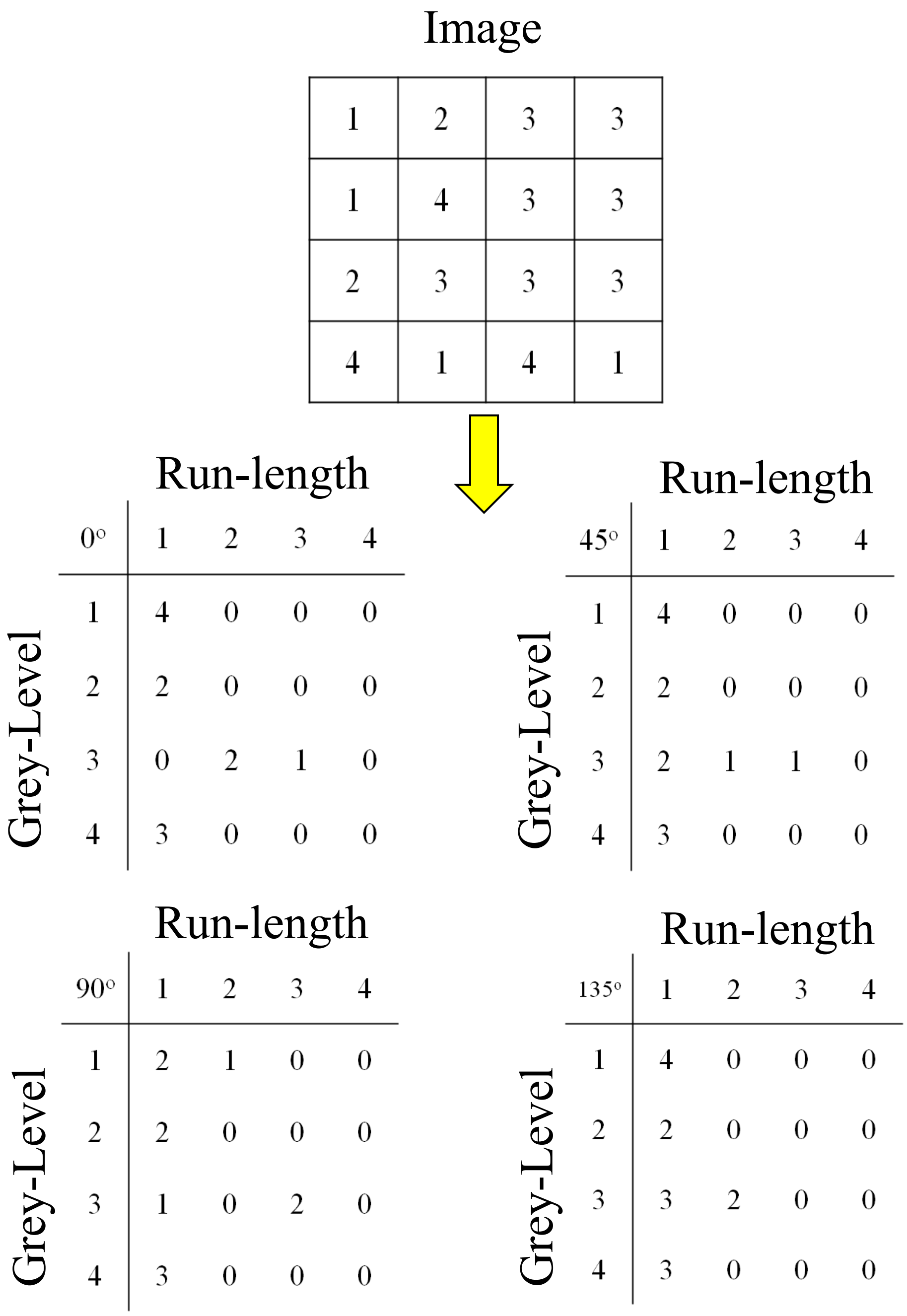
C. Gray Level Run-Length Matrix (GLRLM) and Textual Features

The other method which we applied in this paper to analyzing the ROIs is Gray Level Run-Length Matrix (GLRLM). It is firstly proposed by Galloway in 1975 with 5 features [26]. In 1990, Chu *et al.* [29] suggested two new features to extract gray level information in the matrix before Dasarathy and Holder [30] offered another four features following the idea of joint statistical measure of gray level and run length. Tang [31] provided a good summary of some features achieved from the GLRLM.

C-1. Gray Level Run-Length Matrix

Run-length statistics extract the coarseness of a texture in the different directions. A run is defined as a string of consecutive pixels which have the same gray level intensity along a specific linear orientation. Fine textures contain more short runs with similar gray level while coarse textures have more long runs with significantly different gray level intensities.

For a given image, the pair (i, j) of a run-length matrix $Q(i, j)$ is defined as the run-number of grey level i and run length j as described the following example in Figure 2-9. Hence, the RLM measures how many times there are runs of j consecutive pixels with the same value, with j going from 2 to the length of the longest in a fixed orientation. Even though there are many GLRLM can be defined for a given image, normally 4 matrices are computed, for the horizontal, vertical and diagonal directions. The matrix P has the size $(M \times N)$, where M is equal to the maximum gray level and N is the possible maximum run length in the corresponding image. The typical directions are 0°, 45°, 90°, and 135°, and calculating the run-length encoding for each orientation will produce a run-length matrix.



1	2	3	3
1	4	3	3
2	3	3	3
4	1	4	1

0°	1	2	3	4
1	4	0	0	0
2	2	0	0	0
3	0	2	1	0
4	3	0	0	0

45°	1	2	3	4
1	4	0	0	0
2	2	0	0	0
3	2	1	1	0
4	3	0	0	0

90°	1	2	3	4
1	2	1	0	0
2	2	0	0	0
3	1	0	2	0
4	3	0	0	0

135°	1	2	3	4
1	4	0	0	0
2	2	0	0	0
3	3	2	0	0
4	3	0	0	0

Figure 2-9. GLRLM example. the first row is a 4x4 image with 4 gray levels and the others are the corresponding GLRLMs in 4 directions.

C-2. GLRLM Features:

After a run-length matrix is calculated along a given direction, several texture descriptors are calculated to obtain the texture properties and differentiate among different textures. These descriptors can be used either with respect to each direction or by combining them if a global view of the texture information is required. There are eleven features which are typically extracted from the run-length matrices: Short Run Emphasis (SRE), Long Run Emphasis (LRE), High Gray-Level Run Emphasis (HGRE), Low Gray- Level Run Emphasis (LGRE), pair-wise combinations of the length and gray level emphasis (SRLGE, SRHGE, LRLGE, LRHGE), Run-Length Non-uniformity (RLN), Grey-Level Non-uniformity (GLN), and Run Percentage (RPC). These features describe specific characteristics in the image. For example, (SRE) measures the distribution of short runs in an image, while Run Percentage measures both the homogeneity and the distribution of runs of an image in a specific direction. The formulas for calculating the features and their explanation are provided as following:

1. *Short Run Emphasis:*

It describes the distribution of short runs. This value indicates how much is a texture composed of runs of short length in a given direction.

$$\mathrm{SRE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{Q(i,j)}{j^2} \tag{2.30}$$

With n_r denotes the total number of runs.

2. *Long Run Emphasis:*

Similar to SRE, it describes the distribution of long runs. This value indicates how much is a texture composed of runs of long length in a given direction. These two features give a more in depth information of the coarseness of an image.

$$\mathrm{LRE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}Q(i,j).j^2 \tag{2.31}$$

3. *Low Gray-Level Run Emphasis:*

It describes the distribution of low gray level value. The more low gray level values are in an image, the larger this value is.

$$\mathrm{LGRE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{Q(i,j)}{i^2} \tag{2.32}$$

4. High Gray-Level Run Emphasis:

In contrast with Low Gray-Level Emphasis, it describes the distribution of high gray level value. The higher gray-level values are in an image, the larger this value is.

$$\mathrm{HGRE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} Q(i,j).\,i^2 \tag{2.33}$$

5. Short Run Low Gray-Level Emphasis:

It describes the relative distribution of short runs and low gray level values. The SRLGE value is large for the image with many short runs and lower gray level values.

$$\mathrm{SRLGE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{Q(i,j)}{i^2.j^2} \tag{2.34}$$

6. Short Run High Gray-Level Emphasis:

It describes the relative distribution of short runs and high gray level values. The SRHGE value will be large for the image with many short runs and high gray level values.

$$\mathrm{SRHGE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{Q(i,j).i^2}{j^2} \tag{2.35}$$

7. Long Run Low Gray-Level Emphasis:

It describes the relative distribution of long runs and low gray level values. The LRLGE value will be large for the image with many short runs and high gray level values.

$$\mathrm{LRLGE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} \frac{Q(i,j).j^2}{i^2} \tag{2.36}$$

8. Long Run High Gray-Level Emphasis:

It describes the relative distribution of long runs and high gray level values. The LRHGE value will be large for the image with many short runs and high gray level values.

$$\mathrm{LRHGE} = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} Q(i,j).\,i^2.\,j^2 \tag{2.37}$$

9. Gray-Level Non-uniformity:

It describes the similarity of pixel values throughout the image in a given direction. It is expected small if the gray level values are similar throughout the image.

$$\mathrm{GLN} = \frac{1}{\mathrm{n_r}} \sum_{\mathrm{i=1}}^{\mathrm{M}} \left(\sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{Q(i,j)} \right)^2 \qquad (2.38)$$

10. Run Length Non-uniformity:

It describes the similarity of the length of runs throughout the image in a given direction. It is expected small if the run lengths are similar throughout the image.

$$\mathrm{RLN} = \frac{1}{\mathrm{n_r}} \sum_{\mathrm{j=1}}^{\mathrm{N}} \left(\sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{Q(i,j)} \right)^2 \qquad (2.39)$$

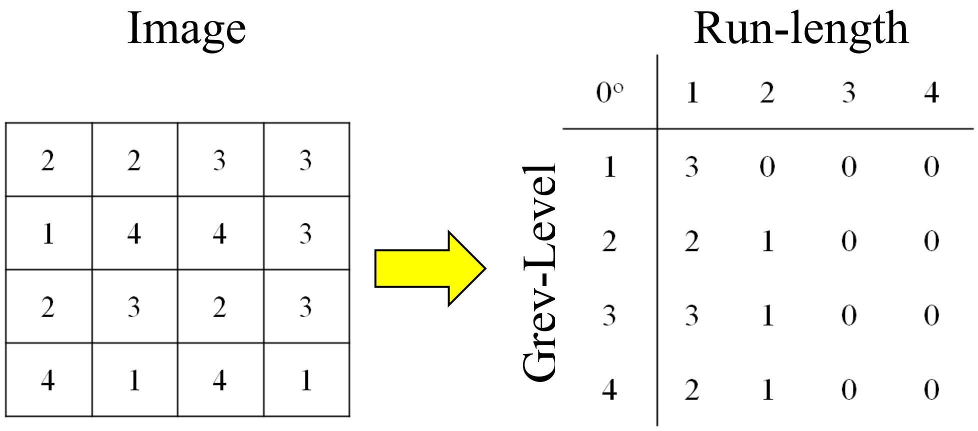
11. Run Percentage:

This feature is not a percentage in spite of its name. It presents the homogeneity and the distribution of runs of an image in a given direction. The RPC is the largest when the length of runs is 1 for all gray levels in a given direction.

$$\mathrm{RPC} = \frac{\mathrm{n_r}}{\mathrm{n_p}} \qquad (2.40)$$

With $\mathrm{n_p}$ is the number of pixels. Figure 2-10 shows an example about calculating the GLRLM from a 4x4 gray-scale image following horizontal direction. We will compute 11 features following the above formulas.

The results of 11 textual features, which can be computed, are shown in Table 2-2.



2	2	3	3
1	4	4	3
2	3	2	3
4	1	4	1

0°	1	2	3	4
1	3	0	0	0
2	2	1	0	0
3	3	1	0	0
4	2	1	0	0

Figure 2-10. A 4x4 gray-scale image and its corresponding GLRLM

Table 2-2. GLRLM features of an image

	Textual Features	Value
1	SRE	0.8269
2	LRE	1.6923
3	SRHGE	5.9423
4	LRLGE	0.4348
5	LRHGE	14.3077
6	LGRE	0.3371
7	HGRE	7.6154
8	SRLGE	0.3126
9	LRN	8.3846
10	GLN	3.3077
11	RPC	0.8125

III. Feature Selection

Feature selection has been an interesting research field in machine learning, pattern recognition, data mining, and statistics. The main idea of feature selection is to eliminate redundant features that contain little or no predictive information as well as keep the useful ones. In order to find optimal features for classification, researchers have proposed several methods to analyse the feature set. In fact, the effectiveness of features on classification is highly problem-dependent. Extracted features could perform very well for one problem, but may give poor performance for others. Hence, we have responsibility for picking proper features for the given problem at hand. Ultimately, we need to find a set of features from various kinds of feature extraction methods, which is optimal for the problem. In this research, we apply Sequential Forward Selection, Sequential Backward Selection and F-score to finding the optimal feature subset.

A. Sequential Forward Selection (SFS).

This method begins by evaluating all samples of dataset which consist of only one input attribute. In other words, we start from the empty set sequentially add the feature x_i which result has the highest objective function $J(Y_k + x_i)$ when combined with the set of features Y_k that have already been selected. Its algorithm is explained as following procedure:

- Step 1: Start with empty set $Y_o = \{\emptyset\}$.
- Step 2: Select the next best feature $x^* = \text{argmax}[J(Y_k + x)]$ with $x \notin Y_k$. In our case, the objective function bases on classification rate from cross validation test. The mean value of 10-fold cross validation test is used to evaluate feature subset.
- Step 3: Update Y_{k+1}=$Y_k + x^*$ and set $k = k + 1$.
- Step 4: Go to step 2.

This process continues until adding feature decreases the criterion. According to the above process, we see that the search space is drawn like an ellipse to emphasize the fact that there are fewer states towards the full or empty sets. For instance, the state space for 4 features is illustrated in the following figure.

To find the overall best input feature set, the easiest way is exhaustive search. However, it is very expensive. Compared with the exhaustive search, forward selection is much cheaper. SFS works best when the optimal subset has a small number of features and the main disadvantage of SFS is that it is unable to remove features that become obsolete after the addition of other features.

B. Sequential Backward Selection (SBS)

Contrary to Sequential Forward Selection, Sequential Backward Selection works in the opposite way. This starts from full of features and sequentially eliminates the worst feature x^* to result in the highest objective function $J(Y_k - x_i)$. Its algorithm is explained as following procedure:

- Step 1: Start with full set $Y_o = X$
- Step 2: Eliminate the worst feature $x^* = \mathrm{argmax}[J(Y_k - x)]$ with $x \in Y_k$. The objective function is the based on classification rate from cross validation test. The mean value of 10-fold cross validation test is used to evaluate feature subset.
- Step 3: Update Y_{k+1}=$Y_k - x^*$ and set $k = k + 1$.
- Step 4: Go to step 2.

This process continues until removing feature decreases the criterion. SBS usually works best when the optimal feature subset has a large number of features, since SBS spends most of its time visiting large subsets. Its procedure could be shown as in the following example.

C. F-score

F-score is a technique which measures the discrimination. Given training vectors x_k, $k = 1, \dots, m$, if the number of positive and negative instances are n+ and n-, respectively, then the F-score of the *ith* feature could be calculated as equation (3.1):

$$F(i) = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \tag{3.1}$$

With $\bar{x}_i$, $\bar{x}_i^{(+)}$, $\bar{x}_i^{(-)}$ are the averages of the *ith* feature of the whole, positive, and negative data and $x_{k,i}^{(+)}, x_{k,i}^{(-)}$ is the *ith* feature of *kth* positive and negative sample respectively. F-score indicates the discrimination between the positive and negative sets so the larger the F-score is, the more likely this feature is more discriminative. Thus, we could consider this score as a criterion for feature selection. A disadvantage of this method is that it cannot reveal mutual information among features as shown in the figure 3-5. In the example, both of features also have lower value of F-score; however, the set of them could classify two groups precisely.

In despite of this drawback, F-score is simple and generally quite effective. We order all features based on its F-scores and then apply a classifier for training/testing for the set included the feature with highest H-score. Then we add the second highest F-score feature to the feature set before training and testing all dataset again. The procedure is repeated until all features added to feature set.

IV. SVM Classification

Support Vector Machine (SVM), which was proposed by Vapnik et al, is a powerful machine learning method based on statistical learning theory. Its theory is based on the idea of finding an optimal hyper-plane to separate two classes. This produces a classifier that could perform well on unseen patterns. Nowadays, SVM has been widely applied in many fields such as regression estimation, environment illumination learning, object recognition, bioinformatics analysis and so forth. In each case, there are usually many possible hyper-planes to separate the groups, but there is only one that has maximal margin.

In this research, two kinds of liver diseases are needed to discriminate. Hence, LIBSVM, a popular machine learning tool for classification, regression, and other machine learning tasks were used for implement multi-class learning task. LIBSVM, which was proposed by Lin *et al* [32], is a library for support vector machines. It is a integrated library for support vector classification (C-SVC, nu-SVC), distribution estimation (one-class SVM). A typical use of LIBSVM includes two steps: the first step is training a data set to obtain a model and the second one is using the model to predict information of a testing data set.

V. Experimental Results and Discussion

A. Performance Evaluation

To reduce the variability of the prediction performance, Cross-validation test was usually applied to evaluate the performance of the proposed system. It is one of the most popular methods to evaluate a model’s prediction performance. If a model was trained and tested on the same data, it is easy to lead to an over-optimistic result. Therefore, the better approach, the holdout method, was to split the training data into disjoint subsets.

doi:10.20944/preprints201810.0073.v1

As it is a single train-and-test method, the error rate we got was resulted from an “unfortunate” split. Moreover, in some case of lacking samples, we cannot afford the "luxury' of setting. The drawbacks of the holdout can be overcome with a family of re-sampling methods, Cross Validation. Two well-known kinds of Cross-validation are Leave-one-out Cross-validation (LOOCV) and k-fold Cross-validation.

In k-Fold Cross-validation, the total samples are randomly partitioned into *k* groups which have the same size. Of the *k* group, one group is for testing the model while the remaining ($k - 1$) groups are used as training data. This process is repeated *k* times (the *folds*) until all groups are tested. Then the results from the *k* experiments can be averaged to produce a single estimation. Thus, the true accuracy is estimated as the average accuracy rate

$$\mathrm{Acc} = \frac{1}{\mathrm{K}} \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{Acc}_{\mathrm{i}} \tag{5.1}$$

The advantage of this method (see in the above figure) is that all samples are used for both training and validation, and each observation is used for validation only one time. Although 10-fold and 5-fold Cross-validation are commonly used, in general *k* is an unfixed parameter.

Leave-one-out Cross-validation could be considered as a degenerate case of k-fold Cross Validation with k is the total number of samples. Consequently, for a data with N samples, LOOCV performs N experiments. In each experiment, only one sample is used for testing while N-1 samples left are for training process.

In this research, we use 10-fold Cross-validation for performance evaluation. The classification result 4 kinds of values: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), which mean is described as Table 5-1:

Table 5-1. The confusion matrix

Condition Prediction	**Positive**	**Negative**
Positive	True Positive	False Positive
Negative	False Negative	True Negative

where true or false are intended for result correction while positive or negative signifies the tumor is either HCC or liver abscess. Based on the information, we could calculate the accuracy factors as (5.2) which describes the performance of classifiers

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}. \tag{5.2}$$

B. Result and Discusion

Various experiments were conducted. In this result, 2 kinds of features (GLCM and GLRLM) were calculated and selected by the feature selection methods (Sequential Forward Selection, Sequential Backward selection or F-score) before classified by Support Vector Machine or Neural Network. The following sections will present the results of the different combination for considering the optimal methods for CAD system.

B-1. Classification by all features

As mentioned in the previous parts, we totally extracted 96 features from a region of interest. They consist many characteristics of each kind of liver diseases. In this experiment, SVM was applied to discriminate the diseases by using each kind of features (GLCM or GLRLM) and using 2 kinds (GLCM and GLRLM) together. The following table shows the result of classification in this case.

Table 5-2. Accuracy rate without applying selection method

GLCM (52 features)	GLRLM (44 features)	GLCM+GLRLM (96 features)
75.75%	54.53%	61%

As can be seen from the Table 5-2, the classification results without applying any feature selection method of SVM are not good except the model, included GLCM and SVM, with the accuracy rate was 75.75%. The other combination of SVM just obtained the detection rate of 54.53% and 61%.

This result shows that both of GLCM and GLRLM features could be applied for classification of HCC and liver abscess; however, GLRLM seems to contain more noise than GLCM. It could adapt to the redundant features in both GLCM and GLRLM while SVM's performance was affected significantly. In the next section, classification was conducted by applying Sequential Forward Selection method.

B-2. Classifications by using Sequential Forward Selection

After applying the Sequential Forward Selection for the different kind of feature sets, the different numbers of features were selected. The results are shown as in figure 5-3. In SVM classification, we set one more condition for SFS process. That is the lower bound of selected features is 4 features because if the number of selected features is not large enough, the result of classification will be not good. For example, the process of SVM, which is used to trains and test all samples with only one features, just takes about 0.3s.

In regards to SVM, we could see that SFS give a slight improvement from 75.75% to 78% with GLCM, and a significant change of recognition rate with GLRLM and the combination of GLCM and GLRLM from 54.53% to 88.13% and from 61% to 89.25% respectively.

Table 5-3. Accuracy rate with applying Sequential Forward Selection method

	Without SFS	**With SFS**	**The number of selected features**
Accuracy rate with GLCM features	75.75%	78%	5
Accuracy rate with GLRLM features	54.53%	88.13%	7
Accuracy rate with GLCM and GLRLM features	61%	89.25%	9

B-3. Classifications by using Sequential Backward Selection

Table 5-4. Accuracy rate with applying Sequential Backward Selection method

	Without SBS	**With SBS**	**The number of selected features**
Accuracy rate with GLCM features	75.75%	75.5%	48
Accuracy rate with GLRLM features	54.53%	88.25%	28
Accuracy rate with GLCM and GLRLM features	61%	88.87%	68

After conducting the Sequential Backward Selection, we got the results shown as in figure 5-4. It is obviously that SBS enhanced the accuracy of all method with the different level. In case of GLCM features, the accuracy was slightly decreased 0.25% with SVM. As for GLRLM and the combination between GLCM and GLCM, the highest result achieved by SVM and the GLCM-GLRLM combination with SBS (88.87%). It is followed by 88.25% of the model which used GLRLM and SVM.

B-4. F-Score

Based on the equation (3.1), the F-score of each feature could be computed and shown as in Table 5-5 and Table 5-6.

Table 5-5. F-score of GLCM

F	F-score	F	F-score	F	F-score	F	F-score
1	0.094006	14	0.046439	27	0.041523	40	0.048657
2	0.000759	15	0.003384	28	0.00427	41	0.000678
3	0.222294	16	0.087401	29	0.104389	42	0.15354
4	0.166121	17	0.115392	30	0.114496	43	0.128984
5	0.225713	18	0.092026	31	0.093941	44	0.135722
6	0.175996	19	0.175218	32	0.176167	45	0.175231
7	0.129382	20	0.126291	33	0.123562	46	0.118854
8	0.144281	21	0.098294	34	0.09322	47	0.092418
9	0.225882	22	0.094314	35	0.100657	48	0.145838
10	0.219637	23	0.086611	36	0.104336	49	0.149344
11	0.22536	24	0.096301	37	0.10026	50	0.14694
12	0.083251	25	0.001884	38	0.00122	51	0.007818
13	0.000225	26	0.003192	39	0.003984	52	0.000644

Table 5-6. F-score of GLRLM

F	F-score	F	F-score	F	F-score	F	F-score
53	0.309454	64	0.108676	75	0.087955	86	0.135784
54	0.105037	65	0.026292	76	0.019915	87	0.031407

55	0.112888	66	0.12426	77	0.136258	88	0.116255
56	0.258012	67	0.097652	78	0.092349	89	0.140583
57	2.50E-05	68	1.20E-05	79	4.00E-06	90	1.20E-06
58	0.113405	69	0.111519	80	0.110956	91	0.110342
59	0.187865	70	0.183068	81	0.17762	92	0.175827
60	0.064373	71	0.108361	82	0.109202	93	0.107328
61	0.11555	72	0.053881	83	0.062097	94	0.083418
62	0.503327	73	0.297173	84	0.266923	95	0.271874
63	0.009315	74	0.001695	85	0.003806	96	0.002537

As shown in Fig. 5-1, when we added more GLCM feature based on their F-score, the accuracy increased and reached the peak at the 36th feature with 77.25% accuracy before slightly decreasing to the end. Meanwhile, SVM achieved 87.125% at the 87th feature (including 34 features) before dropping rapidly from the 63th feature to the end of GLRLM features as shown in Fig. 5-2. In this case, we also can get 86.625% accuracy with only 16 features at the 61st feature of GLRLM. In case of using all features, the trend of accuracy chart was quite similar with the case of applying GLRLM features. SVM can obtain the classification result up to the best accuracy of 88.875%, before plunging to the end as shown in Fig. 5-3.

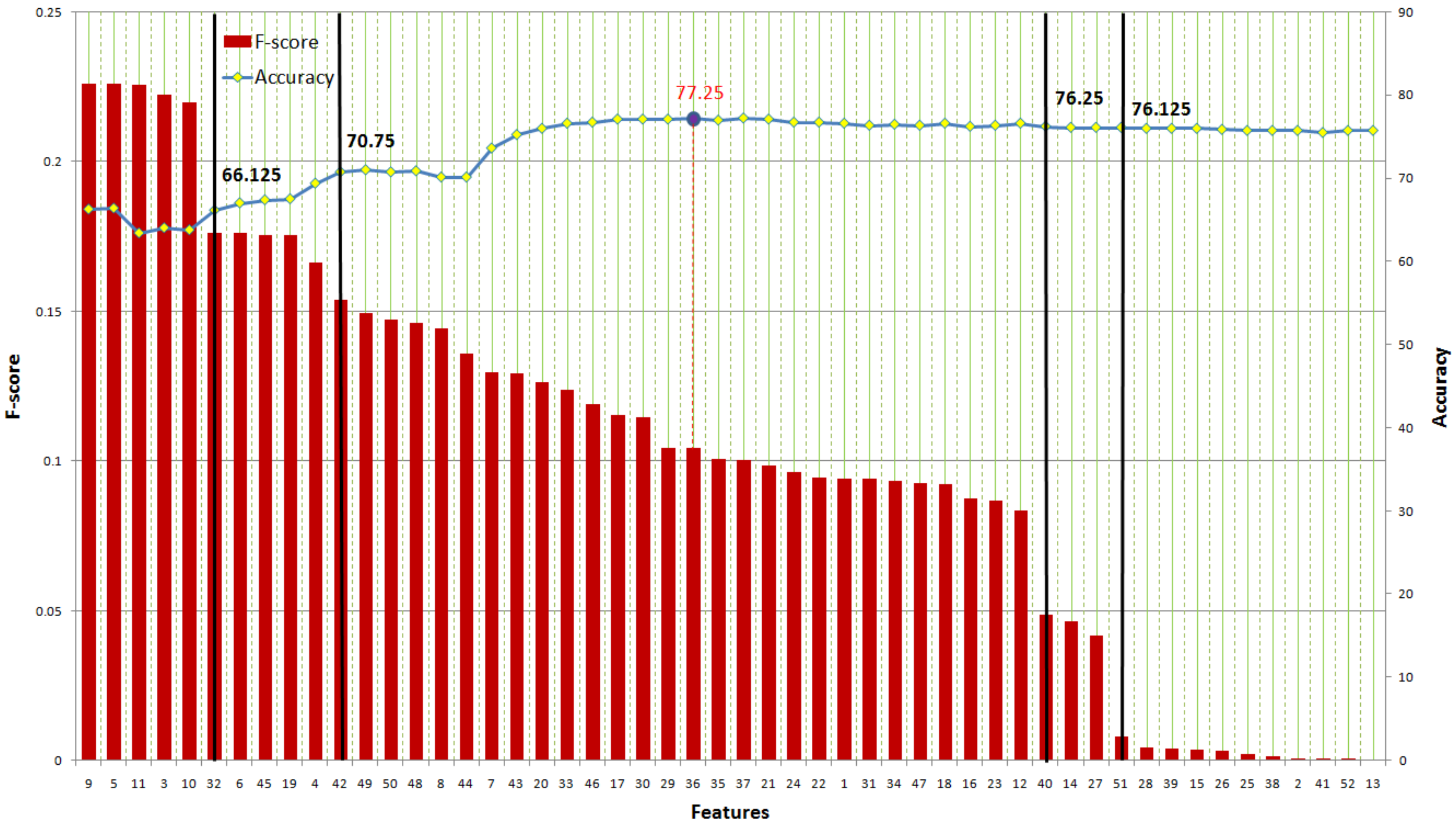


Figure 5-1. The result of GLCM features and SVM

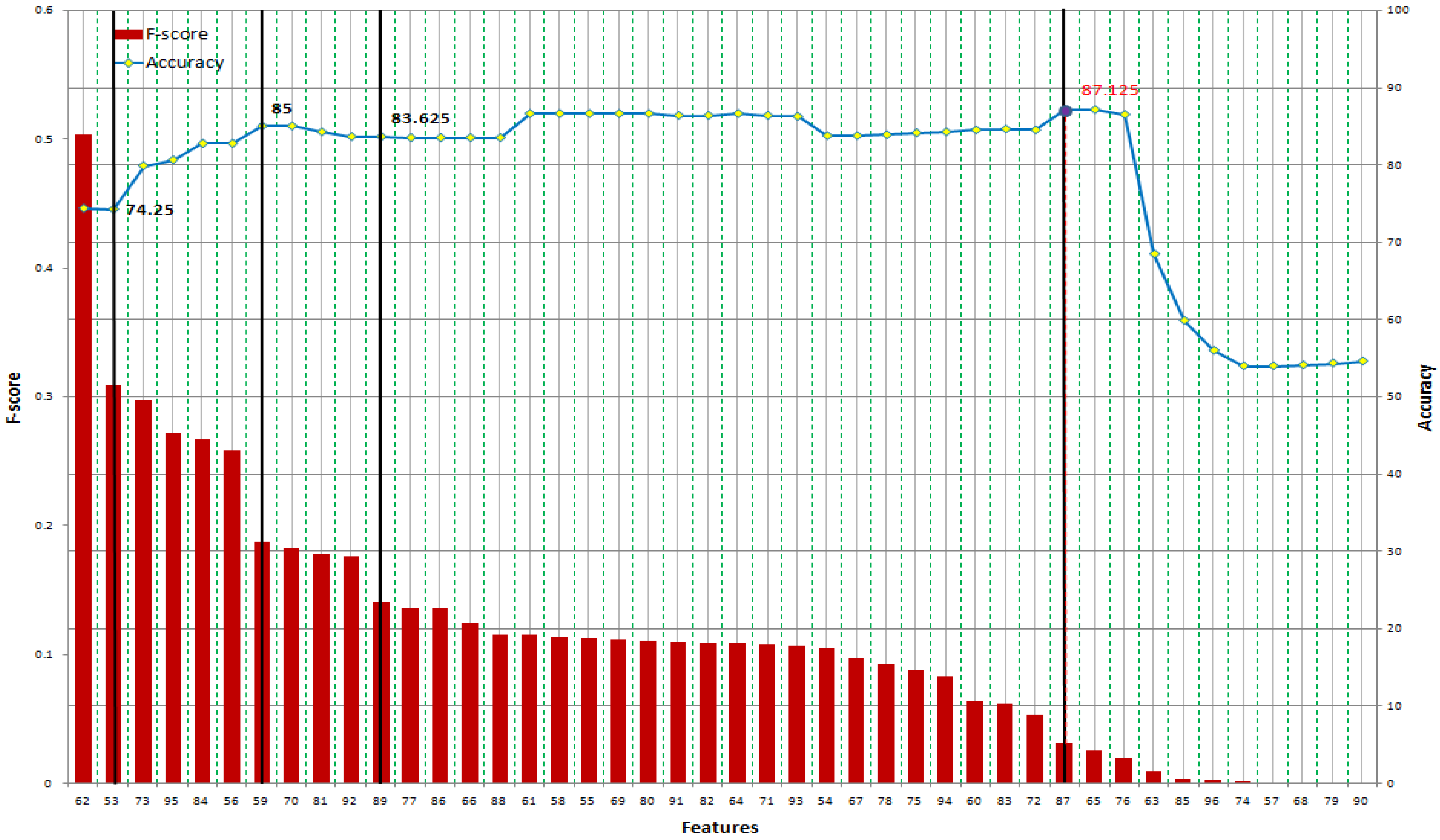


Figure 5-2. The result of GLRLM features and SVM

doi:10.20944/preprints201810.0073.v1

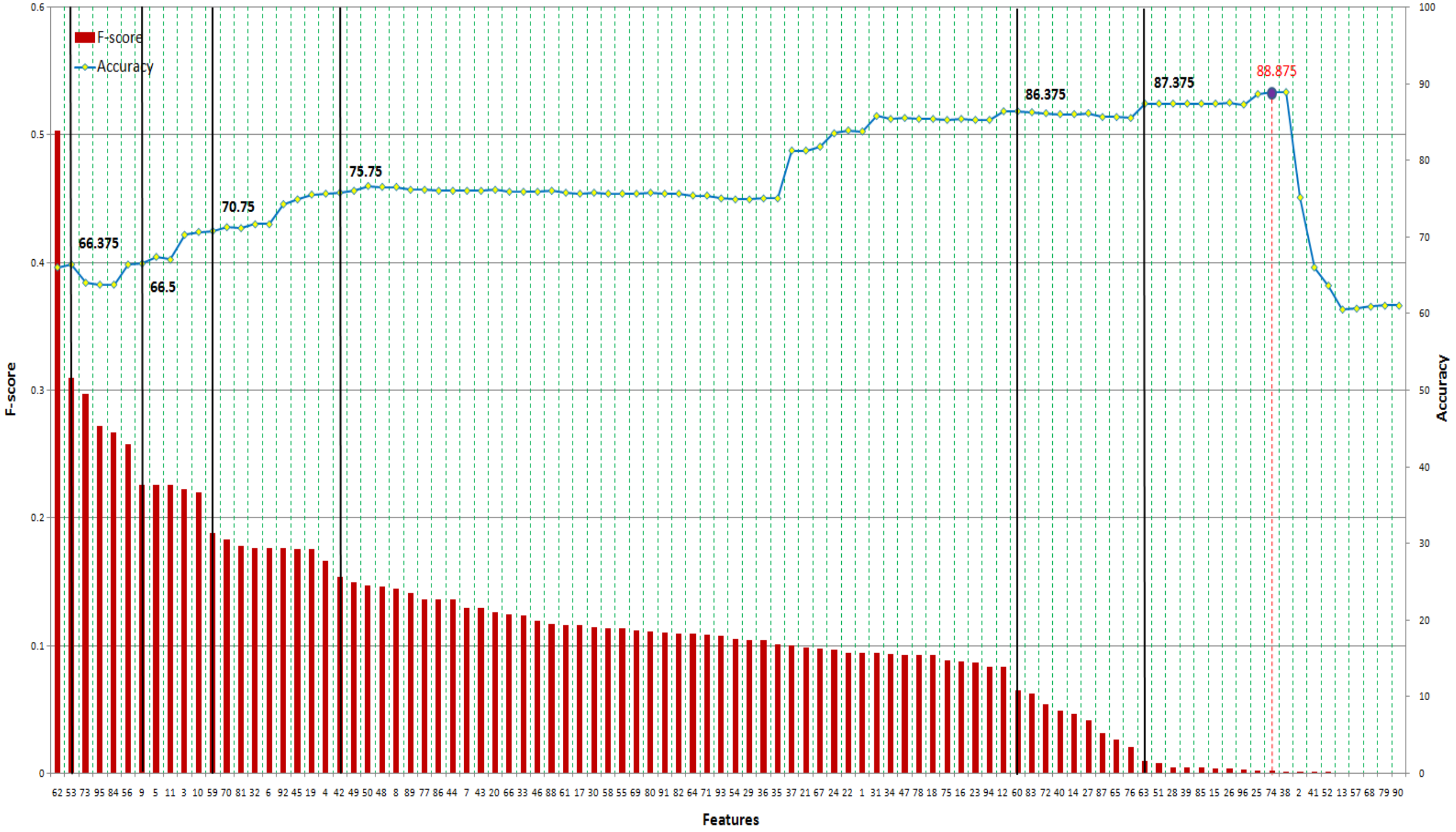


Figure 5-3. The result of all features and SVM

The other way to apply F-score is just picking some thresholds where the gap between low F-score and high F score is considerable. We chose 4 thresholds for each kind of feature and 6 thresholds for all features. The results are shown in Table 5-7.

Table 5-7. Comparison between searching all and using threshold

	Threshold	Search all
GLCM	**76.25%**	**77.25%**
GLRLM	**87.125%**	**87.125%**
GLCM+GLRLM	**87.375%**	**88.875%**

As presented in the Table 5-7, the threshold method could achieve the results which are as good as by searching all features in case of SVM.

C. Performance Analysis

The performance of all methods was summarized as in following Table 5-8. The models using SVM and GLRLM or all features with any selection method gave the most outstanding accuracy, about 88%. For example, SVM tried all cases of features set based on F-score during about 40s. Finally, the best accuracy obtained by the model, which contains SVM and all features selected by SFS, was 89.25%. Nevertheless, in case of reducing the processing load, the threshold of F-score for SVM could be considerd because it also gives the good performance. It took only 1.8s to achieve the classification rate of 87.375% compared with 40s of SFS. It will have the significant meaning for processing a large dataset.

Table5 -8. Overall results

	No Feature Selection	SFS	SBS	F-score (Threshold)	F-score (Search all)
GLCM+SVM	75.75%	78%	75.5%	76.25%	77.25%
GLRLM+SVM	54.53%	88.13%	88.25%	87.125%	87.125%
All features +SVM	61%	89.25%	88.87%	87.375%	88.875%

VI. Conclusion

Hepatocellular Carcinoma (HCC) and liver abscess are ones of the most dangerous diseases all over the world. Due to the requirement of diagnosis of liver disease based on ultrasound images, it is very necessary to develop a CAD system to assist the inexperienced physicians in their decision making. Therefore, this research proposes a system to reduce the erroneous diagnosis for classification of HCC and liver abscess. First, 96 textural features, including 52 features of Gray-Level Co-occurrence Matrix (GLCM) and 44 features of Gray-Level Run-Length Matrix (GLRLM), were extracted from the Region of Interests (ROIs), which were verified by radiologist and recognized by biopsy. In order to obtain the important features, we applied the feature selection (SFS, SBS, and F-score) and select the most discriminative feature set. Finally, the classifiers Support Vector Machine (SVM) were trained from the features of training set and test by 10-fold Cross-validation to get a reliable result. It shows that the proposed system can identify two types of liver disease with high accuracy (up to 89.25%). This research can provide diagnostic assistance while distinguishing two kinds of liver diseases by using the proposed CAD system.

Acknowledgement

This research was supported by the Taipei Medical University (TMU), Taiwan, and the National Taiwan University of Science and Technology (NTUST), Taiwan, under Grants TMU-NTUST-100-11, TMU-NTUST-101-10, and TMU-NTUST-103-12, and the Ministry of Science and Technology (MOST), Taiwan, under the Grant MOST 107-2221-E-011-145.

References

[1] P. Pisani, D. M. Parkin, F. Bray, and J. Ferlay, "Estimates of the worldwide mortality from 25 cancers in 1990," *International Journal of Cancer*, vol. 83, pp. 18-29, 1999.

[2] M. Mozumi and H. Hasegawa, "Adaptive beamformer combined with phase coherence weighting applied to ultrafast ultrasound, " *Applied Sciences*, vol. 8, no. 2, pp. 1-13, 2018.

[3] J. Albinsson, H. Hasegawa, H. Takahashi, E. Boni, A. Ramalli, Å. R. Ahlgren, and M, Cinthio, "Iterative 2D tissue motion tracking in ultrafast ultrasound Imaging," *Applied Sciences*, vol. 8, no. 5, pp. 1-16, 2018.

[4] C. J. Chen and D. S. Chen, "Interaction of hepatitis B virus, chemical carcinogen, and genetic susceptibility: multistage hepatocarcinogenesis with multifactorial etiology," *Hepatology*, vol. 36, pp. 1046-1049, 2002.

[5] W. D. Richard and C. G. Keen, "Automated texture-based segmentation of ultrasound images of the prostate," *Computerized Medical Imaging and Graphics,* vol. 20, pp. 131-140, 1996.

[6] S. Pavlopoulos, G. Konnis, E. Kyriacou, D. Koutsouris, P. Zoumpoulis, and I. Theotokas, "Evaluation of texture analysis techniques for quantitative characterization of ultrasonic liver images, " in *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, 1996, pp. 1151-1152, vol. 3.

[7] J. Bleck, U. Ranft, M. Gebel, H. Hecker, M. Westhoff-Bleck, C. Thiesemann*, et al.*, "Random field models in the textural analysis of ultrasonic images of the liver," *IEEE Transactions on Medical Imaging,* vol. 15, pp. 796-801, 1996.

[8] Y. M. Kadah, A. A. Farag, J. M. Zurada, A. M. Badawi, and A. Youssef, "Classification algorithms for quantitative tissue characterization of diffuse liver disease from ultrasound images," *Medical Imaging, IEEE Transactions on,* vol. 15, pp. 466-478, 1996.

[9] M. K. Gebbinck, J. Verhoeven, J. Thijssen, and T. E. Schouten, "Application of neural networks for the classification of diffuse liver disease by quantitative echography," *Ultrasonic Imaging,* vol. 15, pp. 205-217, 1993.

[10] S. Pavlopoulos, E. Kyriacou, D. Koutsouris, K. Blekas, A. Stafylopatis, and P. Zoumpoulis, "Fuzzy neural network-based texture analysis of ultrasonic images," *Engineering in Medicine and Biology Magazine, IEEE,* vol. 19, pp. 39-47, 2000.

[11] M.-H. Horng, Y.-N. Sun, and X.-Z. Lin, "Texture feature coding method for classification of liver sonography," *Computerized Medical Imaging and Graphics,* vol. 26, pp. 33-42, 2002.

[12] P.-M. Yang, C.-M. Chen, T.-W. Lu, and C.-P. Yen, "Computer-aided diagnosis of sonographic liver cirrhosis: A spleen-reference approach," *Medical Physics,* vol. 35, pp. 1180-1190, 2008.

[13] A. Laghi, R. Iannaccone, P. Rossi, I. Carbone, R. Ferrari, F. Mangiapane*, et al.*, "Hepatocellular Carcinoma: detection with triple-phase multi–detector row helical CT in patients with Chronic Hepatitis 1," *Radiology,* vol. 226, pp. 543-549, 2003.

[14] B. I. Choi, "The current status of imaging diagnosis of hepatocellular carcinoma, "*Liver Transplantation,* vol. 10, pp. S20-S25, 2004.

[15] J. A. A. Pérez, J. J. González, R. F. Baldonedo, L. Sanz, G. Carreño, A. Junco*, et al.*, "Clinical course, treatment, and multivariate analysis of risk factors for pyogenic liver abscess," *The American Journal of Surgery,* vol. 181, pp. 177-186, 2001.

[16] C. Cruz-Gomez, P. Wiederhold, and M. Gudino-Zayas, "Automatic liver tissue segmentation in microscopic images using fusion color space and multiscale morphological reconstruction," in *Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), 2013 International Conference on*, 2013, pp. 88-92.

[17] Z. Jiang, K. Yamauchi, K. Yoshioka, K. Aoki, S. Kuroyanagi, A. Iwata, *et al.*, "Support vector machine-based feature selection for classification of liver fibrosis grade in chronic hepatitis C," *Journal of Medical Systems*, vol. 30, pp. 389-394, 2006.

[18] D. Sabih and M. Hussain, "Automated classification of liver disorders using ultrasound images," *Journal of Medical Systems*, vol. 36, pp. 3163-3172, 2012.

[19] C. R. Hatt, G. Ng, and V. Parthasarathy, "Enhanced needle localization in ultrasound using beam steering and learning-based segmentation," *Computerized Medical Imaging and Graphics*, vol. 41, pp. 46-54, 2015.

[20] R. T. Ribeiro, R. Tato Marinho, and J. M. Sanches, "An Ultrasound-Based Computer-Aided Diagnosis Tool for Steatosis Detection," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, pp. 1397-1403, 2014.

[21] S. Moldovanu, L. Moraru, and D. Bibicu, "Computerized decision support in liver steatosis investigation," *Int. J. Biol. Biomed. Eng*, vol. 6, pp. 69-76, 2012.

[22] S. Poonguzhali and G. Ravindran, "Automatic classification of focal lesions in ultrasound liver images using combined texture features," *Information Technology Journal*, vol. 7, pp. 205-209, 2008.

[23] W. Gomez, W. C. A. Pereira, and A. F. C. Infantosi, "Analysis of Co-occurrence texture statistics as a function of gray-level quantization for classifying breast ultrasound," *IEEE Transactions on Medical Imaging*, vol. 31, pp. 1889-1899, 2012.

[24] S. Onal, S. K. Lai-Yuen, P. Bao, A. Weitzenfeld, and S. Hart, "MRI-based segmentation of pubic bone for evaluation of pelvic organ prolapse," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 1370-1378, 2014.

[25] G. Yi, W. Yuanyuan, K. Dehong, and S. Xianhong, "Automatic classification of intracardiac tumor and thrombi in echocardiography based on sparse representation," *Biomedical and Health Informatics, IEEE Journal of*, vol. 19, pp. 601-611, 2015.

[26] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, pp. 610-621, 1973.

[27] D. Mitrea, S. Nedevschi, M. Lupsor, M. Socaciu, and R. Badea, "Experimenting various classification techniques for improving the automatic diagnosis of the malignant liver tumors, based on ultrasound images," *International Congress on Image and Signal Processing (CISP)*, 2010, pp. 1853-1858.

[28] M. Galloway, "Texture analysis using grey level run lengths," *NASA STI/Recon Technical Report N*, vol. 75, p. 18555, 1974.

[29] A. Chu, C. M. Sehgal, and J. F. Greenleaf, "Use of gray value distribution of run lengths for texture analysis," *Pattern Recognition Letters*, vol. 11, pp. 415-419, 1990.

[30] B. V. Dasarathy and E. B. Holder, "Image characterizations based on joint gray level—run length distributions," *Pattern Recognition Letters*, vol. 12, pp. 497-502, 1991.

[31] X. Tang, "Texture information in run-length matrices," *IEEE Transactions onI mage Processing*, vol. 7, pp. 1602-1609, 1998.

[32] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *The Journal of Machine Learning Research*, vol. 6, pp. 1889-1918, 2005.