Feature Selection and Classification of Diffraction Images

Author: Sai Kiran Thati

Advisor: Dr. Junhua Ding

Contents

List of Figures

List of Tables

Abstract — We present to you our latest research work on diffraction images captured using flow cytometer. In flow cytometry, the investigation of 3D morphological features of biological cells can provide the basis to deeply study the biochemical processes that a cell undergoes during various activities such as apoptosis and mitosis. Also, automatic classification of biological cells is highly desired, as in most of the cases, manual analysis of image data is often necessary to study the underlying morphological processes due to the complex cell structure. In previous studies, we have already shown the potential to automatically classify diffraction images using GLCM and SVM. In this paper, our primary objective is to select the most discriminating GLCM features, and deeply investigate the quantization and displacement factors of GLCM on the performance of two classifiers, Support Vector Machine (SVM) and Logistic Model Trees (LMT) for diffraction images. We computed a total of 20 GLCM textural features for 6 types of cultured cells (100 samples per cell). Without any feature selection the maximum classification accuracy was 90.8%, 86.83% with stratified 10 fold cross validation (10FCV) using SVM and LMT respectively. An optimal subset of features was selected using 2 different approaches. In approach 1, feature selection was done using Extensive Feature Correlation Study (EFCS) resulting in 7 GLCM textural features. In approach 2, feature selection was done using correlation based feature selection algorithm resulting in 8 features. Using features from approach 1, the classification accuracy of SVM increased from 90.8% to 91.16% while it remained the same in the case of LMT. Also, features selected using approach 1 yielded better classification rates than features selected using approach 2 for both classifiers. The findings in this experiment concludes that a set of 7 features selected using approach 1 with a 6-bit(64 levels) quantization scheme along with a displacement of $d=2$ are best to analyze diffraction images using GLCM.

Keywords— Diffraction imaging; GLCM; Extensive Feature correlaion study(EFCS); CFS(Correlation based feature selection algorithm); SVM(Support Vector Machine); Logistic Model Trees(LMT).

I. INTRODUCTION

In molecular cell biology, the study of cellular shape and structure is becoming a major subject for analysis. For example, observing the temporal sequence of morphological changes in cells undergoing apoptosis (programmed cell death) can provide critical information to study how some cancers originate and progress. Not only various cellular activities, but extraction of morphological features are very important for cell study, differentiation and clinical diagnosis. In most of the cases, morphological study of cells relies on conventional microscopy. This is usually labor intensive and relies on fluorescent signals from stained cells as the molecular labels for analysis and classification. Also, it reveals very limited morphological and produces qualitative results rather than quantitative. In [1, 2, 3, 4, 5], through modelling the light scattering by single cells, it was shown that the spatial distribution of elastically scattered light intensity correlates with the 3D morphology of biological cells in the form of intracellular distribution of refractive index. It is in this aspect that many investigators are interested in a flow cytometry method for rapid and label-free classification of biological cells.

However, in current flow cytometry, the technology is designed on the concept to detect angularly integrated signals of scattered light to extract molecular and limited morphological information for cell differentiation. Also, it requires cell staining with multiple fluorescent dyes, thereby increasing the complexity in sample preparation, measurement and data analysis. It is in this context, that we have recently developed a polarization diffraction imaging flow cytometry (p-DIFC) method and shown its potential for cell characterization and classification [6,7,8,9,12,13]. p-DIFC method captures the spatial distribution of coherent light scatters by simultaneously acquiring two cross polarized diffraction images or a polarization image pair from single cells excited by a laser beam. Also, this method doesn't require cell staining with multiple fluorescent reagents. The diffraction pattern or the light scattering occurs due to the varying refractive indices and can be captured as a diffraction image using a camera. Diffraction imaging technique is not new. In fact, diffraction images captured using X-ray crystallography are used to study arrangement of crystalline atoms in three dimensions. Also, diffraction imaging is used in coherent diffraction imaging or lens-less imaging for 2D and 3D reconstruction of various samples ranging from nanoparticles, nanotubes, bio materials and whole cells using X-rays, electrons, and tabletop lasers. To quantitatively characterize the fringe patterns on the diffraction images, we have developed image processing software based on Grey Level Co-occurrence Matrix (GLCM) algorithm for automated classification of cells [7, 10, 11].

In [7], we have clearly demonstrated that p-DIFC method has the capacity for rapid and label-free cell classification using GLCM algorithm. In [10], we have investigated and presented a new label-free method for the measurement of apoptosis in single HL-60 cells and found that 4 GLCM parameters extracted from the diffraction images present strong correlation with the apoptosis rates. Also in [11], we have demonstrated a robust way to classify Jurkat T and Ramos B cells with high classification rates using GLCM and SVM.

Our primary aim is to find an optimal subset of GLCM features that are more related to the class in terms of differentiating them and less correlated among each other. Our secondary aim is to find the optimal displacement factor d and an efficient quantization scheme that can best capture the fringe patterns on the diffraction image with minimal information loss and at the same time is computationally fast. The quantitative evaluation of the set of experiments on GLCM computing parameters, GLCM features and two learning algorithms would help achieve the aforementioned aims.

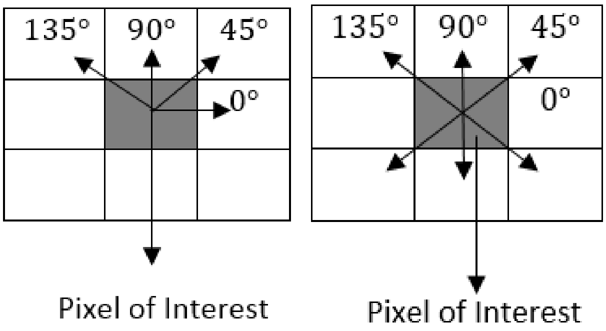
The organization of the paper goes by briefly describing the GLCM algorithm and its application in section 2, then the data set, software tools used, and the method of experiment in section 3. The features selection approaches are discussed in section 4. The classification results and conclusions are discussed in section 5. Lastly, the paper concludes by discussing and concluding the whole study in section 6 and 7 respectively.

II. GLCM AND APPLICATIONS

A. *GLCM Algorithm*

One of the most commonly used technique to study texture analysis is GLCM. The Grey level co-occurrence matrix was proposed by Haralick et al in 1973[26], describing 14 computable textural features based on grey-tone spatial dependencies. GLCM calculates second order statistics which depend on the co-occurrence of the pixel pairs for a given image.

The GLCM defines how often different combinations of gray level pixels occur in an image for a given distance d in a particular angle θ. The distance d here refers to the distance between the pixel under observation and its neighbor. For a given Image I with g number of gray levels, GLCM matrix P will have $g*g$ number of rows and columns. The element $P(i,j)$ of the GLCM specifies the number of times a pixel with intensity i occurs next to a pixel with intensity j in a specified angular direction θ and an offset distance d between the pixels. The angular direction θ is usually calculated using $[0°, 45°, 90°, 135°]$. Fig. 1 shows the possible orientation.



(a) (b)

Fig. 1. (a) Asymmetric computation, (b) Symmetric computation.

GLCM matrices are symmetric i.e. *P(i,j;d,*0°*)= P(j,i;d,*0°*)*. Fig. 1. (b) explains the symmetric computation of GLCM.

For example consider an image having 9 pixels with following gray level intensities. Fig. 2. (a)

1	0	2
1	1	0
2	0	2

(a)

0	2	3
2	2	0
3	0	0

(b)

0	0.16	0.25
0.16	0.16	0
0.25	0	0

(c)

Fig. 2. (a) 3*3 Image, (b) GLCM, (c) Normalized GLCM

Then the GLCM matrix (symmetric) for d=1 and $\theta = 0°$ would be Fig. 2(b). Once the occurrences of all possible gray level pairs are computed, then the GLCM matrix is normalized by the following equation. See Fig. 2. (C)

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}}$$

where, i is the row number (actual gray level of the reference cell) and j is the column number (actual gray level value of the neighboring cell).

B. *Applications*

Texture analysis techniques can be broadly classified into four categories, namely - statistical methods (GLCM, Gray Level run length matrix –GLRLM, Statistical feature Matrix-SFM), Geometrical methods (Voronoi tessellation features, structural models), Model based methods (Markov random fields, Fractals), and Signal processing methods (Spatial domain filtering, Fourier domain filtering, Gabor & Wavelet models) [37].

Among the statistical methods, GLCM is the most commonly used approach for texture analysis. The GLCM has been widely used in medical image analysis for automatic diagnosis of abdominal tumors [28], automatic detection of tumor subtypes in mammograms [29], and prediction of Cirrhosis in ultrasound images [30]. It has also been extensively used in the field of Remote Sensing to differentiate land categories [26] and analyze texture of SAR (Synthetic aperture radar) images [31] [32]. It has also been used in auto classification of insects [34], automatic classification of woven fabrics [33], and many other areas.

The reason behind the popularity of GLCM is that the algorithm is very simple and at the same time provides good results for many kinds of images.

III. EXPERIMENT METHOD

For this experiment, 6 types of cultured cells (100 diffraction images per each cell) were used. The method used for capturing these images can be found in [6] [7] [9] [10] [11]. The 6 cells used are:

a) *Jurkat cells (pronounced yur'kat) are an immortalized line of human T lymphocyte cells*
b) *Ramos is cell line of human B lymphocyte cells.*
c) *HL-60 is a promyelocytic cell line of human promyeloblast cells.*
d) *PC3 (PC-3) is a human prostate cancer cell lines.*
e) *PCS- Primary Dermal Fibroblasts cells derived from normal human skin.*
f) *WBC –Human white blood cells.*

A set of 20 GLCM features (listed below) were computed for a total of 600 images using Matlab (R2013b).The fringe patterns on diffraction images do not have a specific orientation. Instead, the texture is distributed randomly across the image. Hence, for every image four GLCM's were calculated using angles $[0°, 45°, 90°, 135°]$ respectively. Then the mean of the features calculated from these 4 matrices was used as an input to the classifiers. This process was repeated for different combinations of displacement (d=1, 2, 4, 8, 16, and 32) and gray levels (g=8, 16, 32, 64, 128, and 256) yielding 36 combinations in total.

All the feature values computed for 36 different combinations were stored in Excel and then converted to .csv (comma separated value) format.

The Correlation based feature selection (CFS, Approach 2) and the classification analysis using SVM (Support Vector Machine) and Logistic Model trees (LMT) were done in Weka (Version 3.6.12). SVM classification in Weka was done using an open source library for SVM, named LIBSVM (Version 3.20). However, LMT and CFS are in-built in Weka and do not require any external libraries or sources to run.

The classification analysis using SVM and LMT was done in three different scenarios. In the first scenario, all the 20 GLCM features were used. In the second scenario, only features selected using Extensive Feature Correlation Study (EFCS, Approach 1) were used. In the third scenario, only features selected using correlation based feature selection algorithm were used. In all of the scenarios, the classification process was repeated for 36 different combinations with stratified 10 fold cross validation. The classification results were tabulated in excel.

IV. FEATURE SELECTION

Feature selection is the process of selecting a subset of total features present that are highly correlated with the subject or class and yet uncorrelated with each other. For real time classification of biological cells, it is important to select the most relevant features that best capture various textures present on the diffraction images. However, the feature set calculated from GLCM often contains highly correlated features and creates difficulties like increase in computation time, lower learning speed of the machine learning algorithms, and in some cases decrease classification accuracy. Thus, it is important to remove redundant features from the feature set computed from GLCM.

Many researches use feature selection algorithms to select an optimal subset of features. Existing feature selection algorithms for machine learning can be broadly classified into two categories, namely - wrappers and filters. The former are those which evaluate the features using feedback from a learning algorithm, and the latter are those which evaluate features using heuristic process based on the general characteristics of data [27].

Wrappers give better results than filters in terms of the accuracy of a learning algorithm because feature selection is based upon on the feedback received from that particular algorithm. Since the feature selection process is bound to a learning algorithm, wrappers are expensive to run and have to be re-executed each time when switching from one learning algorithm to another [27].

Filters are way faster than wrappers and do not require re-execution when working with different learning algorithms. Correlation based feature selector (CFS) was one such filter algorithm that was used to select an optimal set of features for this experiment.

However, in this experiment we selected two sets of features using two approaches. The first approach which we call it Extensive Feature Correlation Study (EFCS) aims to study 20 most commonly used GLCM features by many researches and select an optimal set based on previous research and numerical results on diffraction images.

In the second approach, we used correlation based feature selection algorithm (CFS) which is one of the most commonly used feature selection algorithm.

A. *Extensive Feature Correlation Study (EFCS) of 20 Textural Features calculated from GLCM (Approach 1):*

GLCM features broadly measure three kinds of texture concepts namely Contrast, Uniformity or Orderliness of pixels, and Correlation among the pixels. This method aims to study 20 most commonly used GLCM features, organize them into one of the texture group they belong to, and then select one or more features from each group that best represent them.

Notation:

Let *p (i, j)* be *(i, j)* th entry in a normalized GLCM, μ_x, μ_y be the mean for the rows and columns of GLCM matrix, σ_x, σ_y be the standard deviations for rows and columns of the GLCM matrix. Let μ be the mean of the entire normalized GLCM.

The features are as follows.

1) *Angular second moment or Energy or Uniformity:*

$$f_1 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j)^2$$

Energy measures the texture uniformity and randomness. Energy calculates the sum of squared elements in GLCM. If an image contains a homogenous texture, it will have very few gray –tone transitions, giving the GLCM only few entries but of very high values. Hence, the energy is high.

2) *Contrast:*

$$f_2 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-j)^2 p(i,j)$$

The diagonal elements of the GLCM represent pixel pairs with no grey level variation (0-0, 1-1, 2-2, 3-3 etc.). If there are large number of entries in the diagonal elements of the matrix, then it means that the image does not have many gray-tone transitions and hence, the contrast is very less. If the number of entries away from the diagonal are more (implying more gray-tone transitions, then the contrast is high due to many gray tone transitions in the image. Contrast increases exponentially as one moves away from the GLCM diagonal.

3) Correlation:

$$f_3 = \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Correlation measures the linear dependence of neighboring gray tones. This features depends on the organization of texture in an image. Given a gray tone in any resolution cell, using a linear function, higher the probability of predicting gray tone in a neighboring resolution cell, higher the value of correlation [14]. This feature is uncorrelated with GLCM energy and entropy (which depend on pixel pair repetition) as it reaches maximum regardless of pixel pair occurrences. This feature is also uncorrelated with GLCM contrast, as the probability of predicting a gray level of one pixel from the second in a given pixel pair is completely independent from contrast [16].

4) Homogeneity or Inverse Difference Moment:

$$f_4 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1+(i-j)^2} p(i,j)$$

IDM measures image homogeneity. It takes large values for smaller gray tone differences in pair elements. It is more sensitive to entries present in or near the diagonal elements of GLCM. It is easy to guess that this feature is inversely correlated to contrast. But, in fact it depends on the combination of energy and contrast. IDM increases when contrast decreases and energy is kept constant. IDM increases when energy decreases and contrast is kept constant. IDM can be constant if there is an increase in contrast and decrease in energy [16]. So, this feature is dependent on their combination. Homogeneity decreases exponentially as one moves away from the GLCM diagonal.

5) Entropy:

$$f_5 = -\sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j) \log(p(i,j))$$

Entropy measures the heterogeneity or the disorder of an image. It is inversely correlated to energy (or ASM). For an image which is not texturally uniform, there will be large number of entries of small values. Hence, entropy reaches maximum at this point. Entropy reaches maximum for an image having pixels of random gray level intensities (white noise).

6) Dissimilarity:

$$f_6 = \sum_i \sum_j |i-j| p(i,j)$$

Dissimilarity measures variation of gray level pairs in an image. This feature is very close to contrast as both of them calculate the same parameter with different weights. Dissimilarity always produces values lower than Contrast as it increases linearly rather than exponentially. This feature is highly correlated with contrast [17].

7) Sum of squares: Variance

$$f_7 = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i-\mu)^2 p(i,j)$$

Variance measures how homogenous areas are spread out on the image. It increases when gray level values differ from their mean. Low variance implies large areas of uniform homogeneity while high variance might indicate many small areas of homogenous regions [15, 16]. It is positively correlated to contrast in many practical cases but they measure different texture concepts. Variance is associated with the average gray level difference between pixels and their mean, unlike contrast where it is associated with the average gray level difference between neighboring pixels [16].

8) Maximum Probability:

$$f_8 = MAX_{i,j}p(i,j)$$

This features measures texture uniformity solely on the basis of the highest probability. This feature was formulated in [18].

Features calculated from Gray level sum histogram (GLSH) and Gray level difference histogram (GLDH)

The gray level sum histogram is a histogram of the sum of the pairs of pixels. It is computed from GLCM by summing the two-dimensional density $p(i,j)$ over constant value of$(i+j)$, i.e.

$$S_k = \sum_{\substack{i=0 \\ i+j=k}}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j)\,, k = 0,..,2N_g - 2.$$

The gray level difference histogram is a histogram of the absolute differences of gray levels from pairs of pixels. It is computed from GLCM by summing the two-dimensional density $p(i,j)$ over constant value of $|i-j|$.

$$D_k = \sum_{\substack{i=0 \\ |i-j|=k}}^{N_g-1} \sum_{j=0}^{N_g-1} p(i,j)\,, k = 0,..,N_g - 1.$$

It is obvious that features f_{10}, f_{11} andf_{14}, f_{15} are conceptually same as Entropy and Variance calculated from GLCM except that they are computed from GLSH and GLDH, respectively. In [21], Weszka compared the classification performance of texture measures computed from GLCM, GLDH, FPS (Fourier Power Spectrum), and GLRLM (Gray level run length matrix) on aerials and LANDSAT imagery. He found that GLCM and GLDH features were most useful and were equal in performance. In [22], Conners & Harlow made a theoretical comparison of four texture analyses to determine the amount of information captured in each of the algorithms. The algorithms that were examined are the spatial gray level dependence method (SGLDM) or GLCM, gray level run length method (GLRLM), gray level difference method (GLDM), and power spectral method (PSM). The results indicate that GLCM performed very well of the four considered.

The advantage of using features from the sum and difference histograms matrices is that, it has lower storage and lower computational requirements. But, deep investigation is necessary to check if there is any information loss when converting GLCM to GLSH and GLDH [23]. Also, a careful feature selection procedure may be required in many cases.

9) Sum Average or GLCM Mean:

$$f_9 = \sum_{k=0}^{2N_g-2} kS_k = \mu_x + \mu_y$$

This feature weighs the pixel value by frequency of its occurrence in combination with a certain neighbor pixel value. This is one of the descriptive statistic on the GLCM.

10) Sum Entropy:

$$f_{10} = -\sum_{k=0}^{2N_g-2} S_k \log S_k$$

11) Sum Variance:

$$f_{11} = \sum_{k=0}^{2N_g-2} (k-f_9)^2 S_k$$

12) Cluster Shade:

$$f_{12} = \sum_{k=0}^{2N_g-2} (k - f_9)^3 S_k$$

13) Cluster Prominence:

$$f_{13} = \sum_{k=0}^{2N_g-2} (k - f_9)^4 S_k$$

Cluster shade and Cluster Prominence are used to measure the skewness (measure of the asymmetry of the probability distribution of pixels) of the matrix. It is believed to emulate human perceptual behavior. The values of these features depend on the sums of rows and columns of the GLSH matrix. These features are close to features that measure the uniformity. When an image has little variations, then the value of these features are low. These features were formulated by Conners and Harlow in [24], in a search to be able to use GLCM not only to discriminate the patterns in texture but to be able to use them to characterize the structure in textures.

14) Difference Variance:

$$f_{14} = \sum_{k=0}^{N_g-1} K^2 D_k$$

15) Difference Entropy:

$$f_{15} = -\sum_{k=0}^{N_{g-1}} D_k \log D_k$$

Other features:

16) Information Measure of Correlation 1:

$$f_{16} = \frac{HXY - HXY1}{\max\{HX, HY\}}$$

Where,

$HXY = -\sum_i \sum_j p(i,j)\log\big(p(i,j)\big)$,

$HXY1 = -\sum_i \sum_j p(i,j)\log\{p_x(i)p_y(j)\}$,

HX and *HY* are entropies of p_x and p_y.

17) Information Measure of Correlation 2:

$$f_{17} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}$$

Where,

$HXY2 = -\sum_i \sum_j p_x(i)p_y(j)\log\{p_x(i)p_y(j)\}$

Correlation measure f_3 is the most popular measure of linear dependence between two random variables but it provides limited information about their dependence structure. Also, it cannot measure dependence when there exists a nonlinear structure between two random variables. In order to overcome these deficiencies, f_{16} f_{17} were introduced by Linfoot in [25]. These features successfully preserve the properties of mutual information and can measure dependence between two random variables even when there exists a nonlinear structure between them. However, these two features are correlated and only usage of f_{16} is recommended [20].

18) Inverse Difference normalized (INN):

$$f_{18} = \sum_{i,j=1}^{N_g} \frac{p(i,j)}{1 + |i-j|/{N_g}^2}$$

This feature is inversely proportional to dissimilarity. This feature is modified by normalizing the gray level difference $|i-j|$ by the number of gray levels N_g.

19) Inverse Difference moment normalized (IDN):

$$f_{19} = \sum_{i,j=1}^{N_g} \frac{p(i,j)}{1 + (i-j)^2/{N_g}^2}$$

This feature is the normalized statistic of f_4. This is done by normalizing the gray level difference $(i-j)$by the number of gray levels N_g.

Without normalization, the above two features $f_{18}f_{19}$ are based on the sum of $p(i,j)$ weighted by a numerical series($\left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}\right\}$ $for\ f_{18}\ or$ $\left\{1, \frac{1}{2}, \frac{1}{5}, \frac{1}{10} \ldots\right\}$ $for\ f_4$.For smooth texture, the un-normalized features are likely to sum $p(i,j)$values that are close to one while coarse textures tend to sum $p(i,j)$ values close to zero. This results in sparse cluster for a smooth texture and tight clusters for coarse texture. This is opposite to the expected behavior and hence normalization provides a series that is more linear in nature. Hence, preventing this drawback [23].

20) Auto Correlation:

$$f_{20} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i,j).p(i,j)$$

This feature measures the amount of regularity as well as fineness/ coarseness of the texture present in the image. Like GLCM correlation it takes high values for image having repetitive or predictive pattern of texture elements. Auto correlation and GLCM correlation provide same information [19].

Table I. Classification of GLCM features into groups

Features	**Contrast Measures Group 1**	Recommended
Contrast(CON)	Measures the gray level local variations in an Image.	Contrast and IDM.
Dissimilarity(DIS)	Measures the same texture concept like Contrast and always produces values lower than Contrast. See Fig. 3.	
Inverse Difference Normalized(INN)	Inversely proportional to Dissimilarity	
Inverse Difference Moment(IDM)	Measure the extent to which similar gray tones tend to be neighbors. Dependent on the combination of Contrast and Energy	
Inverse Difference Moment Normalized(IDN)	Normalized statistic of IDM	
Variance(VAR)	Measures the average gray level difference between pixels and their mean. Close to Contrast in many cases, but both measure different texture concepts.	
Sum Variance(SV)		
Difference Variance(DF)		

Uniformity Or Orderliness Group 2		
Maximum probability(MAX)	Measures texture uniformity solely on the basis of the highest probability of one pair of pixels that dominates other pairs.	Entropy and Difference Entropy
Energy(ENE)	Measures image homogeneity. The more homogenous the larger the values.	
Entropy(ENT)	Measures image Heterogeneity. The more homogenous the lower the values. Inversely proportional to Energy	
Difference Entropy(DENT)		
Sum Entropy(SENT)		
Cluster Shade(CS)	Measures image uniformity and can characterize structure in textures	
Cluster Prominence(CP)		
Correlation and other descriptive statistics Group 3		
Correlation(COR)	Measures the linear dependency between two pixels in an image. 0 is uncorrelated. 1 is perfectly correlated.	Correlation, IMC1 & Sum Average.
Information Measure of correlation 1(IMC1)	Can measures linear dependence between two pixels even when there exists a nonlinear structure between them. Both these features are correlated [20]. Using one of them is recommended.	
Information Measure of correlation 2(IMC2)		
Auto Correlation(AC)	Measure the same information provided by GLCM correlation. For details see [19].	
Sum Average(SA)	This feature weighs the pixel value by frequency of its occurrence in combination with a certain neighbor pixel value.	

Group 1: From Fig. 3, it is evident that, when CON increases exponentially, DIS increases linearly. Choosing one among CON, DIS, and INN is recommended as all the three measure the same texture concept. VAR and CON measure different texture concepts but they hold similar results in many practical situations [16]. So, we ignore VAR. IDM is dependent on the combination of ENE and CON. IDN is a normalized feature of IDM. Hence, we choose CON and IDM to represent group 1 and ignore the rest.

Group 2: From Fig. 4, it is evident that the behavior of ENT, SENT, and DENT’s feature curves for 50 images of HL_60 diffraction images are very similar except for the difference in feature weights. DENT seems to have a normalized range of values compared to the other two. MAX, ENE, and ENT are homogeneity statistics and choosing one of them is recommended.

On the other hand CS and CP are very close to features measuring uniformity but have some interesting properties compared to them. But both these features take high values and often reduce classification performance. In [32] the author reported that due to the high range of values and thus their influence on other features, exclusion of CS and CP increased classification accuracy. Hence, we chose ENT and DENT to represent group 2. Although ENT and DENT are same from a theoretical point of view, the reason for considering the latter is due to its normalized range of values.

Group 3: COR, AC, IMC1, and IMC2 measure the linear dependence between the pixels. But the latter two have some interesting properties compared to others and choosing one of them is recommended. On the other hand SA is uncorrelated with all features in this group and is one of the descriptive statistic of GLCM. Hence we chose COR, IMC1, and SA from group 3.

A total of 7 features were selected using this approach {Contrast, Inverse Difference Moment, Entropy, Difference Entropy, Correlation, Information Measure of Correlation 1 and Sum Average}.

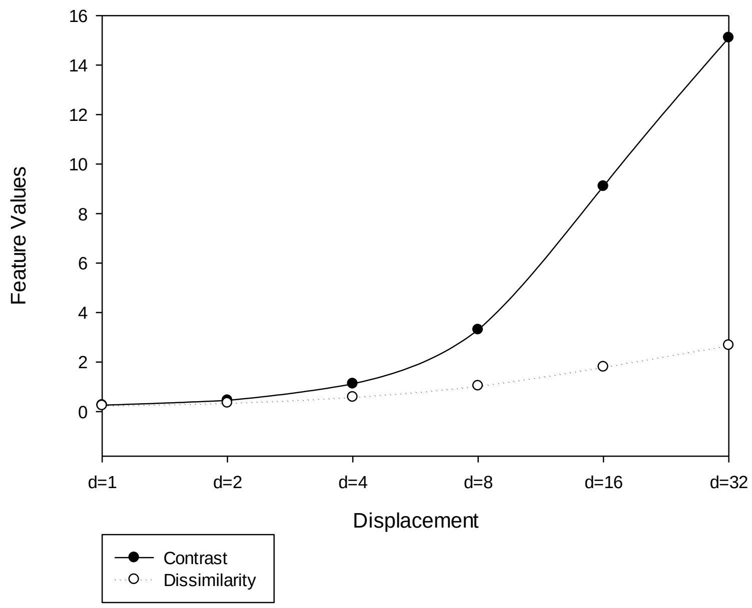


Fig. 3. Feature values for CON and DIS computed for HL_60 cells (*g=64, d=2*). When CON increases exponentially, DIS increases linearly

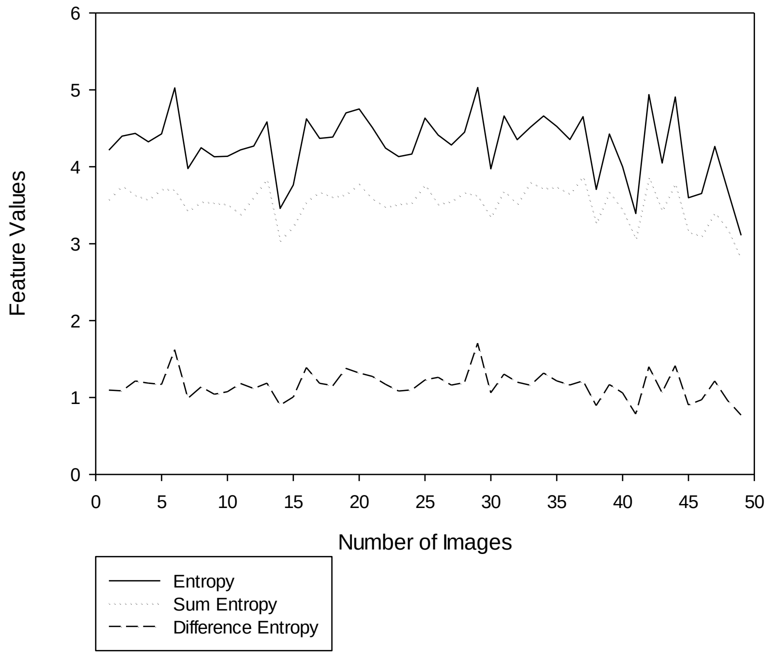


Fig. 4. Feature values of ENT, SENT, and DENT computed for 50 images of HL-60 cells *(g=64, d=2).*

B. *Feature Selection using CFS(Approach 2)*

CFS is a fully automatic algorithm that doesn't require the user to specify any thresholds or the number of features desired. The output of any learning algorithm that uses features selected using CFS can be represented in terms of original features, not in terms of a transformed space as CFS operates on original feature space.

CFS ranks feature subsets according to a correlation based heuristic evaluation function. The evaluation function produces subsets that contain highly correlated features with the class and uncorrelated with each other. Irrelevant features are removed as they have low correlation with the class. Redundant features that are highly correlated with one or more features are also removed. The feature is selected based upon the extent to which it predicts classes in areas of the instance space not already predicted by other features [27].

CFS's feature subset evaluation function is given by:

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \qquad (1)$$

Where, M_s is the heuristic "merit" of a feature subset s containing k features. $\overline{r_{cf}}$ is the average feature correlation ($f \in S$) and $\overline{r_{ff}}$ is the average feature- feature inter correlation. The numerator of the CFS's feature subset evaluation function provides information on how predictive the set of features are in terms of the class, whereas the denominator provides information on how much redundancy there is among the features [27].

The implementation of CFS used in this experiment was done using best first heuristic search strategy. Best first strategy may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).

A total of 8 features were selected using CFS algorithm {Autocorrelation, Correlation, Energy, Homogeneity, Maximum Probability, Sum Average, Information Measure of Correlation 1, and Information Measure of Correlation 2}.

V. Classification results

A. SVM

Support Vector Machines are one of the most successful tool for data classification. It was introduced by Vladimir Vapnik in 1995. A classification problem usually involves test data and training data. The goal of SVM is to build a model based on training data where each instance has a target value (or class labels) with a set of attributes (or features) and predict the target values of the test data given only it's attributes [36]. SVM basically performs binary classification, however several SVM classifiers can be combined to do multiclass classification by comparing 'one against the rest' or 'one against one'. However in this experiment we have used LIBSVM which is an open source library for SVM. LIBSVM is an integrated software for SVM and has wide range of features like allowing the user to set different parameters, try different kernels, display useful statistics, cross validation for mode selection, efficient multiclass classification by comparing 'one against one' rather 'one against the rest' as the former has shorter training time.

The basic idea of SVM is to map the input data on to higher dimensional feature space nonlinearly related to the input space and determine a maximum margin hyper plane or decision boundary to separate the two classes of feature space, where margin is the distance between the hyper plane and the closest data point.

Let $\{x_1,..,x_n\}$ be our training set and let $y_i \in \{1,-1\}$ be the class label of x_i. To find optimal hyper plane $w.x+b=0$, the optimization problem in its basic form is given as

$$min_{w,b}\frac{1}{2}|w|^2+C\sum_{i=1}^{m}\xi_i \qquad (1)$$

subject to $y_i(wx_i+b)\geq 1-\xi_i, \xi\geq 0$

The value of 'C' can be used to set the relative performance of maximizing the margin. In most of the real world problems, boundaries separating the two classes are highly nonlinear. To handle this kind of problems the input vector are mapped into a higher dimensional space where two classes can then be linearly separable [35]. The dot product in the feature space is given by function called kernel 'K'. Then the lagrangian dual of the nonlinear mapping can be given as

$$min_\alpha\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}y_iy_j\alpha_i\alpha_jK(x_i,x_j)-\sum_{i=1}^{m}\alpha_i \qquad (2)$$

subject to $\sum_{i=1}^{m}y_i\alpha_i=0$ and $C_i\geq\alpha_i\geq 0, i=1,..m$

where $\alpha_{1,}\dots\alpha_m\geq 0$ are the lagrane multipliers and 'K' is the kernel function[35]. In this experiment polynomial kernel function was used. It is given as

$$K(x_i,x_j)=(\gamma x_i^t x_j+r)^d, \gamma>0 \qquad (3)$$

Here, r, d & γ are the kernel parameters.

Table II. Classification Results of SVM (no feature selection)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	71.33	70.33	77.33	76.83	75.16	64.66
G=16	82.16	83.66	85.83	82.5	79.83	68.16
G=32	86.33	89	88	85	78.83	71.16
G=64	88	89.66	89.83	85.33	80	72.33
G=128	89	90.83	90.00	87.5	80.66	73.16
G=256	90	90.16	90.66	88.66	82.33	76.66

Table III. Classification Results of SVM (features selected using EFCS- Approach 1)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	69	71.83	75.16	76.33	73.33	64
G=16	79.66	82	83	82.33	77.83	70
G=32	86.16	89.33	89.33	83.83	80.16	70
G=64	88.16	91.16	89.5	84	79.16	69.16
G=128	89.5	91	89.16	84	79.33	71.5
G=256	89.83	90.83	89.5	86.16	83.33	74.5

Table IV. Classification Accuracy of SVM (features selected using CFS- Approach 2)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	68.83	69.16	75.66	75.83	72.66	64.66
G=16	74.83	80.33	81.83	81.66	78.66	68.5
G=32	80.83	86.16	86.66	84.5	80.33	72.5
G=64	83.66	87.66	87.16	83.66	80.33	67.16

G=128	85.16	85.83	86.16	84	79.83	68
G=256	84.33	87.5	86.33	83.83	80.5	70.5

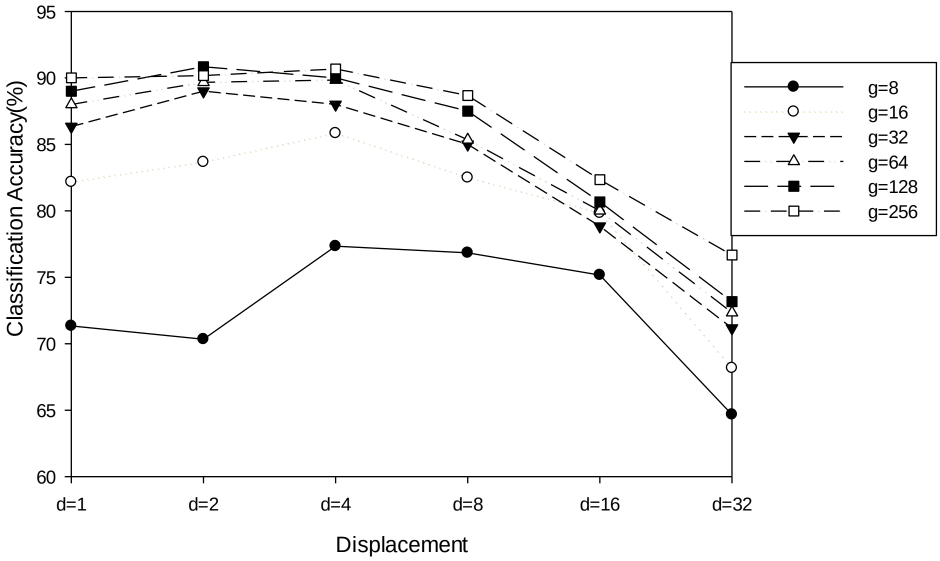


Fig. 5. Classification Accuracy of SVM (no feature selection) for all displacement values across all quantization levels

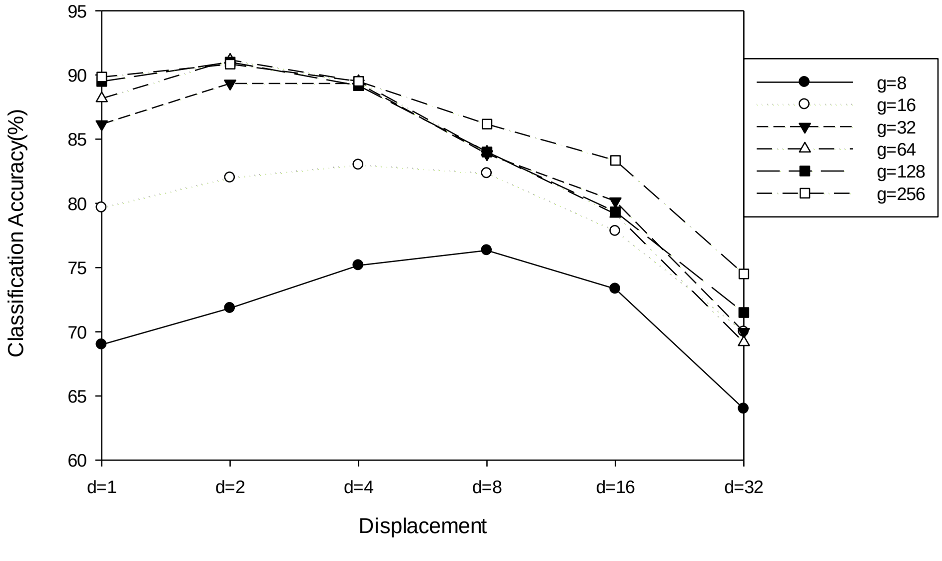


Fig. 6. Classification Accuracy of SVM (features selected using EFCS- Approach 1) for all displacement values across all quantization levels

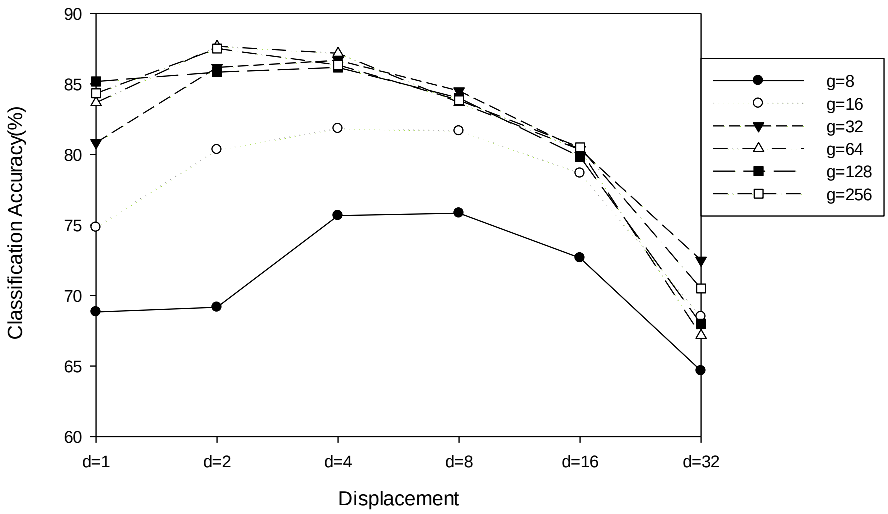


Fig. 7. Classification Accuracy of SVM (features selected using CFS- Approach 2) for all displacement values across all quantization levels

B. *Logistic Model Trees*

A logistic model tree is basically a standard decision tree with logistic regression function at leaves. As in the case of ordinary decision tree a test on one of the attribute is associated with every inner node of the tree. For numeric values, the node has two child nodes and the testing is done by comparing the attribute value with the threshold value. If the value of the instance is smaller than the threshold it is sorted down the left branch and sorted down to the right branch otherwise [38].

The tree structure of the logistic model tree consists of non- terminal (or inner nodes) *N* and a set of leaves or terminal nodes *T*. Let *S*= $D_1 *..* D_m$ denote the whole instance space, spanned by all attributes *V*= $\{v_1, \ldots, v_m\}$ that are present in the dataset. Then every region is represented by a leaf in the tree[38]:

$$S = \bigcup_{t \in T} S_t, \qquad S_t \bigcap S_{t'=\emptyset} \; for \; t \neq t'$$

The leaves of the Logistic Model trees have logistic regression function f_t instead of having a class label. The logistic regression function models the class membership probabilities of all attributes present in the data and is given as

$$\Pr(G = j|X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$$

where $F_j(x) = \alpha_0^j + \sum_{v \in V_t} \alpha_v^j . v$

if $\alpha_{v_k}^j = 0 \; for \; v_k \notin V_t$, the model represented by the whole logistic model tree is then given by

$f(x) = \sum_{t \in T} f_t\,(x).I(x \in S_t)$

where $I = (x \in S_t)$ is 1 if $x \in S_t$ and 0 otherwise.

Table V. Classification Results of LMT (no feature selection)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	66.33	66.83	74.66	74.16	72.66	64.66
G=16	72.16	76.83	80.83	80.66	80	71.16
G=32	80.83	85.16	85.16	81.83	79.16	69.83
G=64	81.83	86.5	85.66	83.33	78.83	67.5
G=128	82.33	85.66	86	83.16	79.66	69
G=256	84.66	86	86.83	82.83	80	73.33

Table VI. Classification Results of LMT (features selected using EFCS- Approach 1)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	64.5	69	74.66	75.16	75.5	66.33
G=16	71	77.33	82	82.16	78.5	68.66
G=32	80.16	85.16	86.33	81.66	77.66	67.5
G=64	82.16	86.83	85.5	82.5	75.66	62.83
G=128	83.33	86.16	85	81.66	76.83	65.33
G=256	82.33	86.16	82.16	79.5	77.83	69.33

Table VII. Classification Results of LMT (features selected using CFS- Approach 2)

Distance	d=1	d=2	d=4	d=8	d=16	d=32
G=8	63.66	69.33	74	67.5	70.16	62.33
G=16	70.5	76.16	80.5	78.5	77.5	69.16
G=32	76.16	83	82.16	81.83	79.16	69.16
G=64	77.5	84.33	83.16	80.33	77.5	67
G=128	80.83	84	82.16	80.66	77.33	65
G=256	80.33	83	81.33	80.83	78.33	68.66

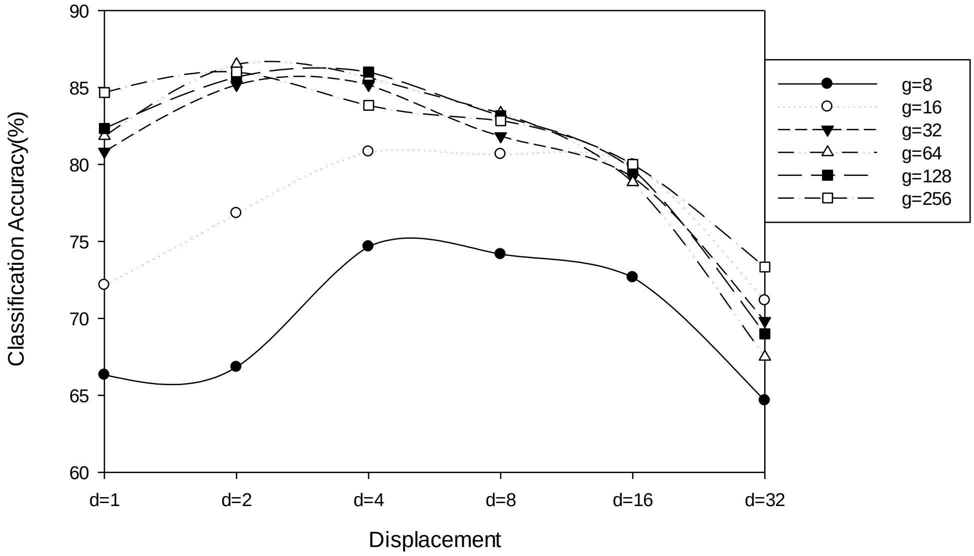


Fig. 8. Classification Accuracy of LMT (no feature selection) for all displacement values across all quantization levels

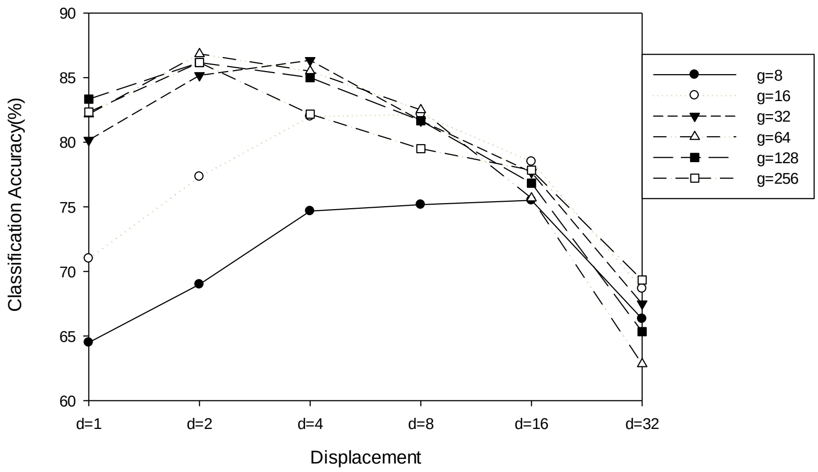


Fig. 9. Classification Accuracy of LMT (features selected using EFCS- Approach 1) for all displacement values across all quantization levels

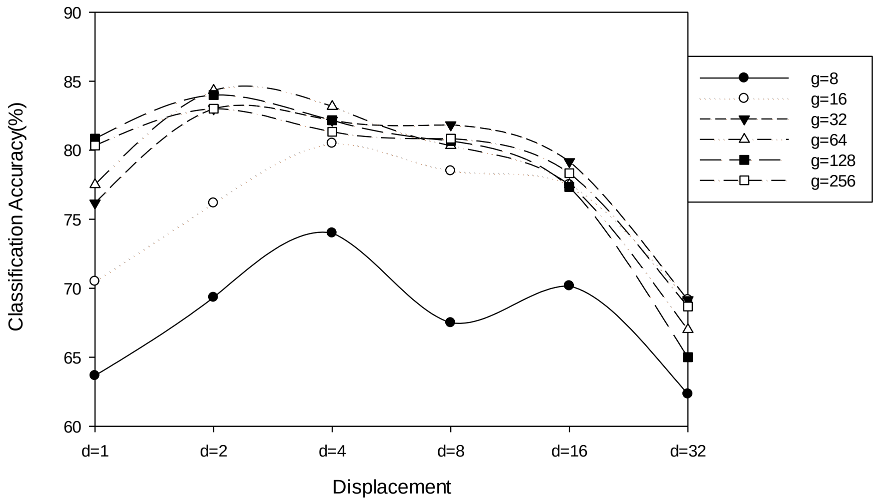


Fig. 10. Classification Accuracy of LMT (features selected using CFS- Approach 2) for all displacement values across all quantization levels

Several conclusion have been drawn from the classification results of SVM and LMT

1. Classification accuracy decreased when the gray level quantization was less than 32. This means that the low quantized versions of GLCM features are not sufficient enough to capture all the information from diffraction images.

2. Classification accuracy was maximum when the gray level quantization was 32 or more. Specifically, the accuracy was maximum when *g=64*. This hints us that 64-level scheme is sufficient to represent the diffraction images. Thus, we eliminate the need for 128-level and 256-level schemes which would computationally be very costly in a context of real time analysis.

3. Classification accuracy was maximum when displacement *d=2 or 4* and decreased when *d>4 or d<2*. This hints us that *d=2 or 4* is best to capture the fringe patterns on the diffraction images. Specifically, the maximum accuracy achieved was 91.16%, 86.83% using SVM and LMT respectively, when *g=64* and *d=2*.

4. Using EFCS approach (7 features) the classification accuracy achieved was 91.16% using SVM, which is more than what was achieved using all the 20 features and features selected using CFS (8 features).

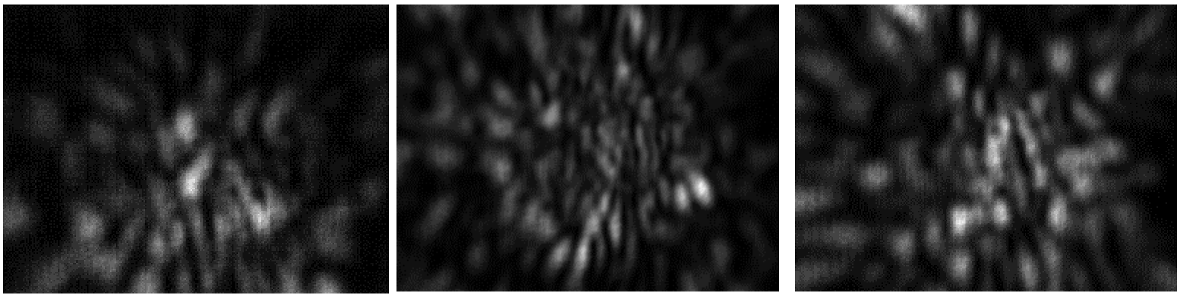
5. Using EFCS approach (7 features) the classification accuracy achieved was 86.83 % using LMT, which is same as what was achieved using all the 20 features and more than features selected using CFS (8 features).

6. The classification performance of SVM outperforms LMT in all cases.

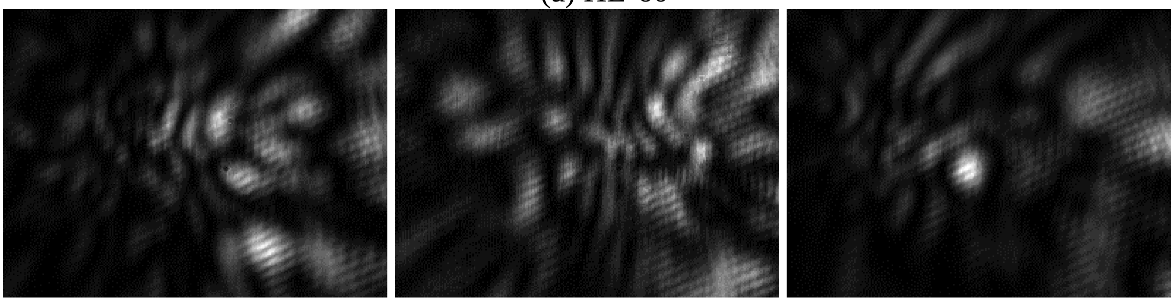
VI. DISCUSSIONS

The study of morphological changes in biological cells is highly desired and can lead insights to many underlying activities. Second order GLCM statistics computed from diffraction images strongly provide the basis to distinguish different kinds of cells according to their diffraction patterns captured in flow cytometer settings. In [7], [10], and [11], although we have shown the potential of GLCM to classify diffraction images, much investigation was necessary to find the optimal feature set, effect of quantization level and displacement vector *d*. Since the dimension of GLCM is dependent on the number of gray levels, if the number is large, the computation of the GLCM would be very costly. On the other hand if the number of gray levels were too low it would fail to capture the texture elements in an image. Hence, through this extensive experimentation, it is clearly evident that a quantization level of 64-level with a displacement of $d=2$ best represents the diffraction images.

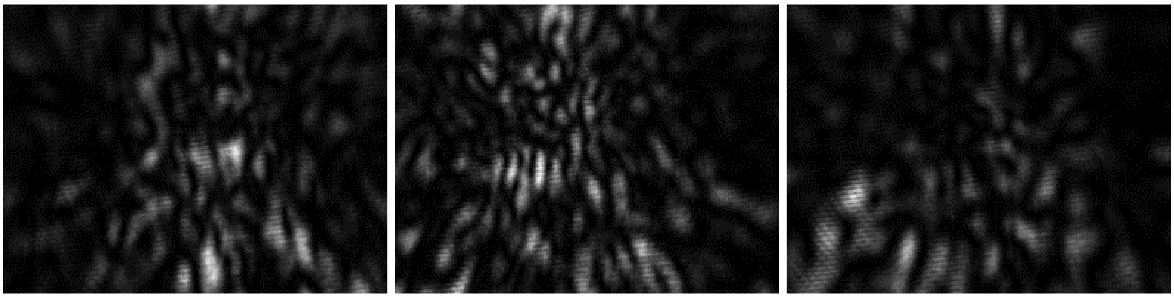
A close observation of the diffraction images of 6 type of cells in Fig. 11 discloses noticeable differences in their fringe patterns. We strongly argue that an optimal set of seven features selected using EFCS approach {Contrast, Correlation, Inverse Difference Moment, Entropy, Difference Entropy, Information Measure of Correlation 1 and Sum Average} are sufficient to quantify these differences. The validity of this argument can be verified by observing the classification results not just based on a single classifier, but both SVM and LMT, where the classification accuracy was better than the CFS approach and using all the 20 features.



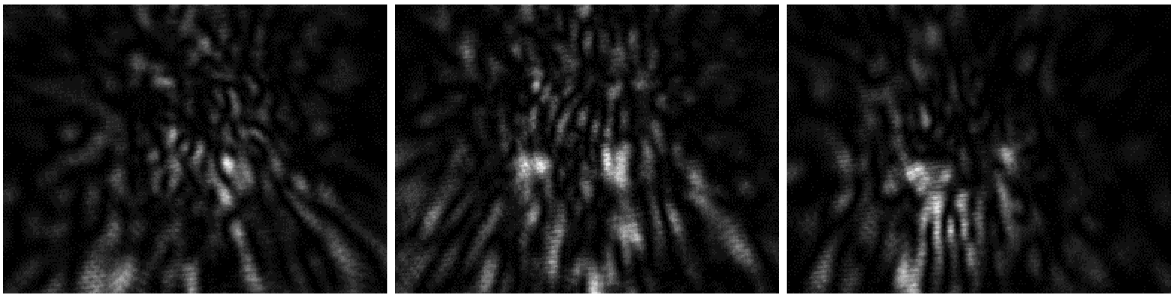
(a) HL-60



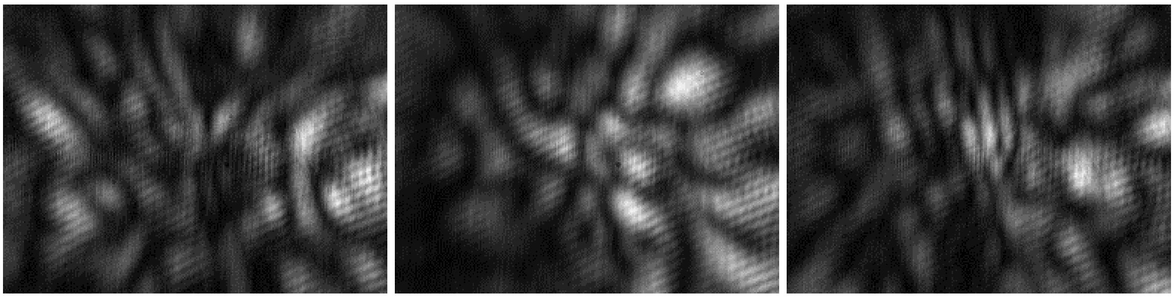
(b) JURKAT



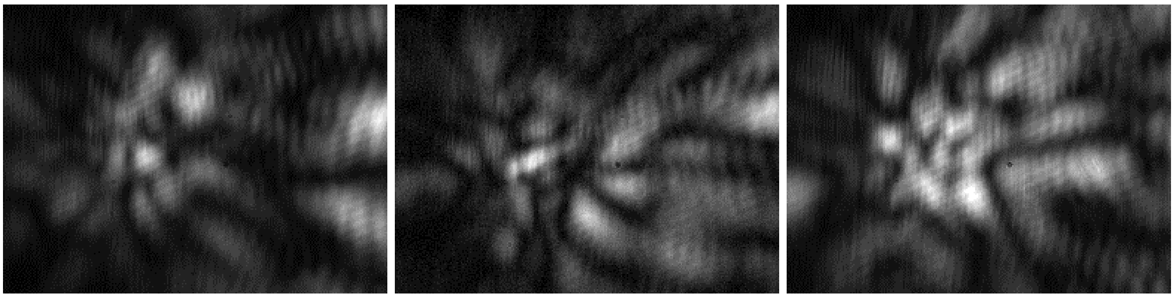
(c) PC3



(d) PCS



(e) RAMOS



(f) WBC

Fig. 11. 8-bit Diffraction images of 6 types of cells captured using (p-DIFC).

Note* the original raw images were normalized from 12-bit to 8-bit to eliminate the images of over-exposure and under-exposure.


(a) HL-60


(b) JURKAT


(c) PC3


(d) PCS


(e) RAMOS

(f) WBC

Fig. 12. 6-bit (64 levels) uniform quantized diffraction images in Matlab.

From Table 8, the CON values for PCS, PC3 and RAMOS are higher than the other cells. This is expected as one can observe from Fig. 12(c), (d), and (e) that the number of local variations are more compared to HL-60, JURKAT and WBC cells. COR and IMC1 values for HL-60 and WBC are somewhat higher than the other cells. This could be due to the predictive nature of the pattern or the linear dependence between the pixels. Also as the features values of COR and IMC1 are mean of four orientations, as opposed to a single orientation, hence it is difficult to understand the nature of values.

ENT values for RAMOS & WBC are higher than other cells implying the texture is less homogenous than other cells. On the other hand, SA values are similar for RAMOS &WBC as well as for PC3 & PCS and different for HL-60 & JURKAT.

Table VIII. Mean & Standard deviation values of 7 GLCM features selected using EFCS approach (*g=64, d=2)*

C #	CON	COR	IDM	ENT	DENT	SA	IMC1
1	1.66±0.751	0.983±0.006	0.673±0.074	4.274±0.0393	1.153±0.16	15.084±2.721	-0.52±0.061
2	2.45±1.012	0.974±0.007	0.619±0.074	4.389±0.46	1.299±0.17	14.309±3.055	-0.453±0.052
3	3.18±1.165	0.957±0.008	0.575±0.065	4.282±0.412	1.412±0.161	11.576±2.064	-0.371±0.038
4	3.01±1.028	0.957±0.007	0.6±0.051	4.181±0.352	1.385±0.131	11.132±1.835	-0.384±0.026
5	3.72±1.392	0.974±0.008	0.496±0.063	5.268±0.311	1.501±0.151	26.024±3.797	-0.409±0.045
6	2.486±1.14	0.985±0.007	0.563±0.067	5.109±0.32	1.33±0.154	26.736±4.384	-0.48±0.05

Table IX. Confusion Matrix for SVM (*g=64, d=2*, accuracy=91.16%)

a	b	c	d	e	f	←classified as
95	4	0	0	1	0	a=HL-60
3	93	4	0	0	0	b=JURKAT
0	1	84	15	0	0	c=PC3
0	1	18	81	0	0	d=PCS
0	0	0	0	99	1	e=RAMOS
0	0	0	0	5	95	f=WBC

Table X. Confusion Matrix for LMT *(g=64, d=2,* accuracy=86.83%)

a	b	c	d	e	f	←classified as
90	7	2	0	1	0	a=HL-60
9	87	4	0	0	0	b=JURKAT
0	4	81	15	0	0	c=PC3
0	1	19	80	0	0	d=PCS
3	0	0	0	92	5	e=RAMOS
0	0	0	0	9	91	f=WBC

From Table 9 and 10, it is evident that most of the misclassification occurs for PC3 and PCS cells. On the other hand the average classification accuracy for other cells is around 95.5 %(SVM) and 90 %(LMT).

VII. CONCLUSION

In this paper, we have accomplished two tasks. First, we deeply investigated 20 commonly used GLCM parameters and proved that an optimal set of 7 features {Contrast, Correlation, Inverse Difference Moment, Entropy, Difference Entropy, Information measure of Correlation 1, and Sum Average} are sufficient to describe the diffraction images. Also, we have proved that these features perform better than those selected using CFS approach.

Second, we investigated that effect of quantization and displacement factors of GLCM on the performance of two classifiers, SVM and LMT. The findings conclude that a 6-bit (64 levels) quantization scheme along with a displacement of *d=2* are best to analyze diffraction images using GLCM.

REFERENCES

[1] A. Dunn, C. Smithpeter, A. J. Welch, and R. Richards-Kortum , "Finite-difference time-domain simulation of light scattering from single cells," *J. Biomed. Opt. 2(3),* 262–266, 1997.

[2] R. S Brock, X. H. Hu, P. Yang , and J. Q Lu, "Evaluation of a parallel FDTD code and application to modeling of light scattering by deformed red blood cells," *Opt. Express 13(14),* 5279–5292, 2005.

[3] J. Q. Lu, P. Yang, and X. H. Hu, "Simulations of light scattering from a biconcave red blood cell using the finite-difference time-domain method," *J. Biomed. Opt. 10(2),* 024022, 2005.

[4] R. S. Brock, X. H. Hu, D. A. Weidner, J. R. Mourant, and J. Q. Lu, "Effect of detailed cell structure on light scattering distribution: FDTD study of a B-cell with 3D structure constructed from confocal images," *J. Quant. Spectrosc. Radiat. Transf. 102(1),* pp. 25–36, 2006.

[5] H. Ding et al., "Angle-resolved Mueller matrix study of light scattering by B-cells at three wavelengths of 442, 633, and 850 nm," *J. Biomed. Opt. 12(3),* 034032, 2007.

[6] K. M. Jacobs, J. Q. Lu, and X. H. Hu, "Development of a diffraction imaging flow cytometer," *Opt. Lett.34 (19),* pp. 2985–2987, 2009.

[7] K. Dong et al., "Label-free classification of cultured cells through diffraction imaging," *Biomed. Opt. Express 2(6),* pp. 1717–1726, 2011.

[8] S. Yu et al., "A novel method of diffraction imaging flow cytometry for sizing microspheres," *Opt. Express 20(20),* pp. 22245–22251, 2012.

[9] J. Zhang et al., "Analysis of cellular objects through diffraction images acquired by flow cytometry," *Opt. Express 21(21),* pp. 24819–24828, 2013.

[10] Xu Yang et al., "A quantitative method for measurement of HL-60 cell apoptosis based on diffraction imaging flow cytometry technique," *Biomed Opt Express, 5(7):* pp. 2172–2183, Jul. 2014.

[11] Yuanming Feng et al.,"Polarization Imaging and Classification of Jurkat T and Ramos B Cells Using a Flow Cytometer," *Cytometry Part A, 85,* pp. 817-826, June 2014.

[12] Y. Sa et al., "Study of low speed flow cytometry for diffraction imaging with different chamber and nozzle designs*." Cytometry A 2013; 83A*,pp. 1027–1033, 2013.

[13] Jacobs KM.et al., “Diffraction imaging of spheres and melanoma cells with a microscope objective,” *J. Biophotonics 2(8-9),* pp. 521–527, 2009.

[14] Robert Haralick, K. Shanmugam and I. Dinstein, “On Some Quickly Computable Features for Texture,” in *Proceedings of the 1972 Symposium on Computer Image Processing and Recognition*, University of Missouri, Vol.2, pp. 12-2-1 to 12-2-10, August, 1972.

[15] Robert Haralick, “On a Texture-Context Feature Extraction Algorithm for Remotely Sensed Imagery,” *Proceedings of the IEEE Computer Society Conference on Decision and Control*, Gainesville, FL, pp. 650-657, Dec. 1971.

[16] A. Baraldi and F. Parmiggiani, “An investigation of the textural characteristics associated with gray level co-occurrence matrix statistical parameters,” *IEEE Trans. Geosci. Remote Sensing*, vol. 33, pp. 293–304, Mar. 1995.

[17] A. Gebejes and R. Huertas, “Texture Characterization based on Grey-Level Co-occurrence Matrix,” *Proceedings in Conference of Informatics and Management Sciences*, 2013.

[18] Robert Haralick, “Statistical and Structural Approaches to Texture,” *Proceedings of the IEEE*, Vol. 67, No. 5, May, 1979, pp. 786-804.

[19] Van der Sanden, J. J. and D. H. Hoekman, “Review of relationships between grey-tone co-occurrence, semivariance and autocorrelation based image texture analysis approaches.” *Canadian Journal of Remote Sensing*, vol. 38 no. 3, pp. 207-213, 2005.

[20] J. M. Carstensen, “Description and simulation of visual texture,” Ph.D. dissertation, Inst. Math. Modeling, Technical Univ. Denmark, Lyngby, Denmark, 1992.

[21] J. S. Weszka, C. R. Dyer, and Azriel Rosenfeld, “A comparative study of texture measures for terrain classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, 6(4), pp. 269-285, 1976.

[22] R.W Conners and C.A. Harlow, “A theoretical comparison of texture algorithms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence 2*, pp. 204-222, 1980.

[23] David Anthony Clausi, “Texture Segmentation of SAR sea ice imagery,” Phd thesis, University of Waterloo, 1996.

[24] R. W. Conners, M. M. Trivedi, and C. A. Harlow, “Segmentation of a high-resolution urban scene using texture operators,” *Comput. Vision, Graph., Image Processing*, vol. 25, pp.273, 1984.

[25] E.H Linfoot, “An information measure of correlation,” *Inform Contr*, Vol. 1, pp. 85-89, 1957.

[26] R. M. Haralick, K. Shanmugan, and I. H. Dinstein, "Textural features for image classification", *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp.610 -621, 1973.

[27] M. A Hall, " Correlation-based Feature Selection for Machine Learning," Unpublished doctoral dissertation, The University of Waikato, Hamilton, Newzealand, 1999.

[28] D. Mitrea, M. Socaciu, R. Badea and A. Golea, " Texture based characterization and automatic diagnosis of the abdominal tumors from ultrasound images using third order GLCM features," in*: Proceedings of the Fourth International Congress on Image and Signal Processing,* pp. 1558-1562, 2011.

[29] M. Mohamed Fathima, D. Manimegalai, and S. Thaiyalnayaki, "Automatic detection of tumor subtype in mammograms based On GLCM and DWT features using SVM," *Information Communication and Embedded Systems (ICICES),* vol., no., pp.809,813, 21-22, Feb. 2013.

[30] Jitendra Virmani, Vinod Kumar, Naveen Kalra, and Niranjan Khandelwal, "Prediction of Cirrhosis Based on Singular Value Decomposition of Gray Level Co-occurence Marix and a neural Network Classifier," *Developments in E-systems Engineering (DeSE),* p.p 146 – 151, 2011.

[31] Jia Longhao, Zhou Zhongfa, and Li Bo, "Study of SAR Image Texture Feature Extraction Based on GLCM in Guizhou Karst Mountainous Region," *Remote Sensing, Environment and Transportation Engineering (RSETE),* 2012 2nd International Conference on, 2012.

[32] L. Soh and C. Tsatsoulis, "Texture analysis of sar sea ice imagery using gray level co-occurances matrices," *IEEE Trans. on Geoscience and Remote Sensing*, 37(2):pp. 780-795, 1999.

[33] Yassine Ben Salem, and Salem Nasri, "Automatic Classification of Woven Fabrics using Multi-class Support Vector Machine," *J RJTA,* 13 (2), pp. 28–36, 2009.

[34] L.Q. Zhu and Z. Zhang, "Auto-classification of Insect Images Based on Color Histogram and GLCM," *Seventh International Conference on Fuzzy Systems and Knowledge Discovery,* 2010.

[35] Lawrence R. Rabiner, "A tutorial on HMMs and selected applications in speech recognition," *Proceedings of IEEE,* Vol77. N02, Feb. 1989.

[36] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, " A Practical Guide to Support Vector Classification," Technical Report, National Taiwan University, 2003.

[37] M. Tuceryan and A. K. Jain, “Texture Analysis,” In *Handbook of Pattern Recognition and Computer Vision*, Second Edition, C. H. Chen, L. F. Pau, and P. Wang (editors), World Scientific Publishing, Co., pp. 235 – 276,1998.

[38] L. Niels, H. Mark, and F. Eibe, “Logistic model trees”. Mach. Learn., pp. 161–205,2005.

APPENDIX 1 CLASSIFICATION STATISTICS AND RESULTS

64, d=2, SVM no feature selection

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         538               89.6667 %
Incorrectly Classified Instances        62               10.3333 %
Kappa statistic                          0.876
Mean absolute error                      0.0344
Root mean squared error                  0.1856
Relative absolute error                 12.4    %
Root relative squared error             49.7996 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.95      0.012      0.941     0.95      0.945      0.969    HL_60
                 0.9       0.012      0.938     0.9       0.918      0.944    JURKAT
                 0.88      0.044      0.8       0.88      0.838      0.918    PC3
                 0.83      0.02       0.892     0.83      0.86       0.905    PCS
                 0.91      0.02       0.901     0.91      0.905      0.945    RAMOS
                 0.91      0.016      0.919     0.91      0.915      0.947    WBC
Weighted Avg.    0.897     0.021      0.898     0.897     0.897      0.938

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 95  3  1  0  1  0 |  a = HL_60
  5 90  4  1  0  0 |  b = JURKAT
  0  3 88  9  0  0 |  c = PC3
  0  0 17 83  0  0 |  d = PCS
  1  0  0  0 91  8 |  e = RAMOS
  0  0  0  0  9 91 |  f = WBC
```

64,d=2, SVM CFS feature selection

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         514               85.6667 %
Incorrectly Classified Instances        86               14.3333 %
Kappa statistic                          0.828
Mean absolute error                      0.0478
Root mean squared error                  0.2186
Relative absolute error                 17.2    %
Root relative squared error             58.6515 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.9       0.014      0.928     0.9       0.914      0.943    HL_60
                 0.88      0.02       0.898     0.88      0.889      0.93     JURKAT
                 0.73      0.056      0.723     0.73      0.726      0.837    PC3
                 0.77      0.052      0.748     0.77      0.759      0.859    PCS
                 0.95      0.022      0.896     0.95      0.922      0.964    RAMOS
                 0.91      0.008      0.958     0.91      0.933      0.951    WBC
Weighted Avg.    0.857     0.029      0.858     0.857     0.857      0.914

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 90  7  1  0  2  0 |  a = HL_60
  6 88  5  1  0  0 |  b = JURKAT
  0  2 73 25  0  0 |  c = PC3
  0  1 22 77  0  0 |  d = PCS
  1  0  0  0 95  4 |  e = RAMOS
  0  0  0  0  9 91 |  f = WBC
```

64,d=2, SVM EFCS feature selection

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         547               91.1667 %
Incorrectly Classified Instances        53                8.8333 %
Kappa statistic                          0.894
Mean absolute error                      0.0294
Root mean squared error                  0.1716
Relative absolute error                 10.6    %
Root relative squared error             46.0435 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.95      0.006      0.969     0.95      0.96       0.972    HL_60
                 0.93      0.012      0.939     0.93      0.935      0.959    JURKAT
                 0.84      0.044      0.792     0.84      0.816      0.898    PC3
                 0.81      0.03       0.844     0.81      0.827      0.89     PCS
                 0.99      0.012      0.943     0.99      0.966      0.989    RAMOS
                 0.95      0.002      0.99      0.95      0.969      0.974    WBC
Weighted Avg.    0.912     0.018      0.913     0.912     0.912      0.947

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 95  4  0  0  1  0 |  a = HL_60
  3 93  4  0  0  0 |  b = JURKAT
  0  1 84 15  0  0 |  c = PC3
  0  1 18 81  0  0 |  d = PCS
  0  0  0  0 99  1 |  e = RAMOS
  0  0  0  0  5 95 |  f = WBC
```

64,d=2, LMT no feature selection

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         519               86.5    %
Incorrectly Classified Instances        81               13.5    %
Kappa statistic                          0.838
Mean absolute error                      0.0664
Root mean squared error                  0.1866
Relative absolute error                 23.9145 %
Root relative squared error             50.07   %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.88      0.018      0.907     0.88      0.893      0.967    HL_60
                 0.87      0.028      0.861     0.87      0.866      0.974    JURKAT
                 0.8       0.046      0.777     0.8       0.788      0.961    PC3
                 0.81      0.034      0.827     0.81      0.818      0.964    PCS
                 0.93      0.024      0.886     0.93      0.907      0.978    RAMOS
                 0.9       0.012      0.938     0.9       0.918      0.986    WBC
Weighted Avg.    0.865     0.027      0.866     0.865     0.865      0.972

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 88  9  1  0  2  0 |  a = HL_60
  8 87  4  1  0  0 |  b = JURKAT
  0  4 80 16  0  0 |  c = PC3
  0  1 18 81  0  0 |  d = PCS
  1  0  0  0 93  6 |  e = RAMOS
  0  0  0  0 10 90 |  f = WBC
```

64,d=2, LMT CFS feature selection

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	506	84.3333 %
Incorrectly Classified Instances	94	15.6667 %
Kappa statistic	0.812	
Mean absolute error	0.0752	
Root mean squared error	0.1996	
Relative absolute error	27.0629 %	
Root relative squared error	53.5662 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.9	0.022	0.891	0.9	0.896	0.962	HL_60
	0.81	0.016	0.91	0.81	0.857	0.966	JURKAT
	0.77	0.054	0.74	0.77	0.755	0.949	PC3
	0.85	0.042	0.802	0.85	0.825	0.968	PCS
	0.9	0.034	0.841	0.9	0.87	0.987	RAMOS
	0.83	0.02	0.892	0.83	0.86	0.987	WBC
Weighted Avg.	0.843	0.031	0.846	0.843	0.844	0.97	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 90  5  2  0  1  2 |  a = HL_60
  7 81 11  1  0  0 |  b = JURKAT
  1  2 77 20  0  0 |  c = PC3
  0  1 14 85  0  0 |  d = PCS
  2  0  0  0 90  8 |  e = RAMOS
  1  0  0  0 16 83 |  f = WBC
```

64,d=2, LMT EFCS feature selection

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	521	86.8333 %
Incorrectly Classified Instances	79	13.1667 %
Kappa statistic	0.842	
Mean absolute error	0.0606	
Root mean squared error	0.1813	
Relative absolute error	21.811 %	
Root relative squared error	48.6593 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.9	0.024	0.882	0.9	0.891	0.972	HL_60
	0.87	0.024	0.879	0.87	0.874	0.969	JURKAT
	0.81	0.05	0.764	0.81	0.786	0.959	PC3
	0.8	0.03	0.842	0.8	0.821	0.975	PCS
	0.92	0.02	0.902	0.92	0.911	0.979	RAMOS
	0.91	0.01	0.948	0.91	0.929	0.99	WBC
Weighted Avg.	0.868	0.026	0.87	0.868	0.869	0.974	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 90  7  2  0  1  0 |  a = HL_60
  9 87  4  0  0  0 |  b = JURKAT
  0  4 81 15  0  0 |  c = PC3
  0  1 19 80  0  0 |  d = PCS
  3  0  0  0 92  5 |  e = RAMOS
  0  0  0  0  9 91 |  f = WBC
```

64,d=4, SVM no feature selection
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	539	89.8333 %
Incorrectly Classified Instances	61	10.1667 %
Kappa statistic	0.878	
Mean absolute error	0.0339	
Root mean squared error	0.1841	
Relative absolute error	12.2 %	
Root relative squared error	49.3964 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.94	0.016	0.922	0.94	0.931	0.962	HL_60
	0.9	0.01	0.947	0.9	0.923	0.945	JURKAT
	0.85	0.034	0.833	0.85	0.842	0.908	PC3
	0.86	0.03	0.851	0.86	0.856	0.915	PCS
	0.91	0.018	0.91	0.91	0.91	0.946	RAMOS
	0.93	0.014	0.93	0.93	0.93	0.958	WBC
Weighted Avg.	0.898	0.02	0.899	0.898	0.899	0.939	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 94  5  0  0  1  0 |  a = HL_60
  6 90  3  0  1  0 |  b = JURKAT
  0  0 85 15  0  0 |  c = PC3
  0  0 14 86  0  0 |  d = PCS
  2  0  0  0 91  7 |  e = RAMOS
  0  0  0  0  7 93 |  f = WBC
```

64,d=4, SVM CFS feature selection
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	523	87.1667 %
Incorrectly Classified Instances	77	12.8333 %
Kappa statistic	0.846	
Mean absolute error	0.0428	
Root mean squared error	0.2068	
Relative absolute error	15.4 %	
Root relative squared error	55.4977 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.91	0.018	0.91	0.91	0.91	0.946	HL_60
	0.88	0.014	0.926	0.88	0.903	0.933	JURKAT
	0.85	0.044	0.794	0.85	0.821	0.903	PC3
	0.82	0.03	0.845	0.82	0.832	0.895	PCS
	0.9	0.026	0.874	0.9	0.887	0.937	RAMOS
	0.87	0.022	0.888	0.87	0.879	0.924	WBC
Weighted Avg.	0.872	0.026	0.873	0.872	0.872	0.923	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 91  7  1  0  0  1 |  a = HL_60
  9 88  3  0  0  0 |  b = JURKAT
  0  0 85 15  0  0 |  c = PC3
  0  0 18 82  0  0 |  d = PCS
  0  0  0  0 90 10 |  e = RAMOS
  0  0  0  0 13 87 |  f = WBC
```

64,d=4, SVM EFCS feature selection

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	537	89.5 %
Incorrectly Classified Instances	63	10.5 %
Kappa statistic	0.874	
Mean absolute error	0.035	
Root mean squared error	0.1871	
Relative absolute error	12.6 %	
Root relative squared error	50.1996 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.93	0.02	0.903	0.93	0.916	0.955	HL_60
	0.9	0.012	0.938	0.9	0.918	0.944	JURKAT
	0.87	0.044	0.798	0.87	0.833	0.913	PC3
	0.8	0.026	0.86	0.8	0.829	0.887	PCS
	0.94	0.016	0.922	0.94	0.931	0.962	RAMOS
	0.93	0.008	0.959	0.93	0.944	0.961	WBC
Weighted Avg.	0.895	0.021	0.897	0.895	0.895	0.937	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
93	6	0	0	1	0	a = HL_60
8	90	2	0	0	0	b = JURKAT
0	0	87	13	0	0	c = PC3
0	0	20	80	0	0	d = PCS
2	0	0	0	94	4	e = RAMOS
0	0	0	0	7	93	f = WBC

64,d=4, LMT no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	514	85.6667 %
Incorrectly Classified Instances	86	14.3333 %
Kappa statistic	0.828	
Mean absolute error	0.0726	
Root mean squared error	0.186	
Relative absolute error	26.1242 %	
Root relative squared error	49.9016 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.88	0.028	0.863	0.88	0.871	0.975	HL_60
	0.84	0.02	0.894	0.84	0.866	0.978	JURKAT
	0.82	0.044	0.788	0.82	0.804	0.974	PC3
	0.82	0.034	0.828	0.82	0.824	0.983	PCS
	0.9	0.026	0.874	0.9	0.887	0.99	RAMOS
	0.88	0.02	0.898	0.88	0.889	0.99	WBC
Weighted Avg.	0.857	0.029	0.857	0.857	0.857	0.982	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
88	8	1	0	2	1	a = HL_60
12	84	3	1	0	0	b = JURKAT
0	2	82	16	0	0	c = PC3
0	0	18	82	0	0	d = PCS
1	0	0	0	90	9	e = RAMOS
1	0	0	0	11	88	f = WBC

64,d=4, LMT CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	499	83.1667 %
Incorrectly Classified Instances	101	16.8333 %
Kappa statistic	0.798	
Mean absolute error	0.0784	
Root mean squared error	0.2057	
Relative absolute error	28.2391 %	
Root relative squared error	55.193 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.88	0.03	0.854	0.88	0.867	0.964	HL_60
	0.84	0.026	0.866	0.84	0.853	0.966	JURKAT
	0.8	0.052	0.755	0.8	0.777	0.971	PC3
	0.76	0.036	0.809	0.76	0.784	0.979	PCS
	0.89	0.04	0.817	0.89	0.852	0.977	RAMOS
	0.82	0.018	0.901	0.82	0.859	0.983	WBC
Weighted Avg.	0.832	0.034	0.834	0.832	0.832	0.973	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
88	9	1	0	2	0	a = HL_60
12	84	2	1	1	0	b = JURKAT
0	3	80	17	0	0	c = PC3
0	1	23	76	0	0	d = PCS
2	0	0	0	89	9	e = RAMOS
1	0	0	0	17	82	f = WBC

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	513	85.5 %
Incorrectly Classified Instances	87	14.5 %
Kappa statistic	0.826	
Mean absolute error	0.0676	
Root mean squared error	0.1911	
Relative absolute error	24.3445 %	
Root relative squared error	51.2728 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.87	0.03	0.853	0.87	0.861	0.963	HL_60
	0.86	0.028	0.86	0.86	0.86	0.974	JURKAT
	0.78	0.038	0.804	0.78	0.792	0.966	PC3
	0.84	0.03	0.848	0.84	0.844	0.981	PCS
	0.88	0.026	0.871	0.88	0.876	0.974	RAMOS
	0.9	0.022	0.891	0.9	0.896	0.986	WBC
Weighted Avg.	0.855	0.029	0.855	0.855	0.855	0.974	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
87	8	1	0	3	1	a = HL_60
12	86	2	0	0	0	b = JURKAT
2	5	78	15	0	0	c = PC3
0	0	16	84	0	0	d = PCS
1	1	0	0	88	10	e = RAMOS
0	0	0	0	10	90	f = WBC

128 ,d=2, SVM no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	545	90.8333 %
Incorrectly Classified Instances	55	9.1667 %
Kappa statistic	0.89	
Mean absolute error	0.0306	
Root mean squared error	0.1748	
Relative absolute error	11 %	
Root relative squared error	46.9042 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.95	0.006	0.969	0.95	0.96	0.972	HL_60
	0.91	0.012	0.938	0.91	0.924	0.949	JURKAT
	0.87	0.036	0.829	0.87	0.849	0.917	PC3
	0.88	0.022	0.889	0.88	0.884	0.929	PCS
	0.92	0.018	0.911	0.92	0.915	0.951	RAMOS
	0.92	0.016	0.92	0.92	0.92	0.952	WBC
Weighted Avg.	0.908	0.018	0.909	0.908	0.909	0.945	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
95	3	1	0	1	0	a = HL_60
3	91	5	1	0	0	b = JURKAT
0	3	87	10	0	0	c = PC3
0	0	12	88	0	0	d = PCS
0	0	0	0	92	8	e = RAMOS
0	0	0	0	8	92	f = WBC

128 ,d=2, SVM CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	526	87.6667 %
Incorrectly Classified Instances	74	12.3333 %
Kappa statistic	0.852	
Mean absolute error	0.0411	
Root mean squared error	0.2028	
Relative absolute error	14.8 %	
Root relative squared error	54.4059 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.93	0.01	0.949	0.93	0.939	0.96	HL_60
	0.91	0.014	0.929	0.91	0.919	0.948	JURKAT
	0.82	0.048	0.774	0.82	0.796	0.886	PC3
	0.8	0.034	0.825	0.8	0.812	0.883	PCS
	0.93	0.024	0.886	0.93	0.907	0.953	RAMOS
	0.87	0.018	0.906	0.87	0.888	0.926	WBC
Weighted Avg.	0.877	0.025	0.878	0.877	0.877	0.926	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
93	4	1	0	0	2	a = HL_60
4	91	4	1	0	0	b = JURKAT
1	1	82	16	0	0	c = PC3
0	1	19	80	0	0	d = PCS
0	0	0	0	93	7	e = RAMOS
0	1	0	0	12	87	f = WBC

128 ,d=2, SVM EFCS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	546	91 %
Incorrectly Classified Instances	54	9 %
Kappa statistic	0.892	
Mean absolute error	0.03	
Root mean squared error	0.1732	
Relative absolute error	10.8 %	
Root relative squared error	46.4758 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.95	0.008	0.96	0.95	0.955	0.971	HL_60
	0.93	0.012	0.939	0.93	0.935	0.959	JURKAT
	0.87	0.044	0.798	0.87	0.833	0.913	PC3
	0.81	0.024	0.871	0.81	0.839	0.893	PCS
	0.98	0.016	0.925	0.98	0.951	0.982	RAMOS
	0.92	0.004	0.979	0.92	0.948	0.958	WBC
Weighted Avg.	0.91	0.018	0.912	0.91	0.91	0.946	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 95  3  1  0  1  0 |  a = HL_60
  3 93  3  1  0  0 |  b = JURKAT
  0  2 87 11  0  0 |  c = PC3
  0  1 18 81  0  0 |  d = PCS
  0  0  0  0 98  2 |  e = RAMOS
  1  0  0  0  7 92 |  f = WBC
```

128 ,d=2, LMT no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	514	85.6667 %
Incorrectly Classified Instances	86	14.3333 %
Kappa statistic	0.828	
Mean absolute error	0.0646	
Root mean squared error	0.1924	
Relative absolute error	23.2393 %	
Root relative squared error	51.6345 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.87	0.018	0.906	0.87	0.888	0.957	HL_60
	0.86	0.028	0.86	0.86	0.86	0.955	JURKAT
	0.83	0.052	0.761	0.83	0.794	0.959	PC3
	0.8	0.03	0.842	0.8	0.821	0.971	PCS
	0.93	0.036	0.838	0.93	0.882	0.977	RAMOS
	0.85	0.008	0.955	0.85	0.899	0.982	WBC
Weighted Avg.	0.857	0.029	0.86	0.857	0.857	0.967	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 87  8  2  0  3  0 |  a = HL_60
  6 86  5  3  0  0 |  b = JURKAT
  0  5 83 12  0  0 |  c = PC3
  0  1 19 80  0  0 |  d = PCS
  3  0  0  0 93  4 |  e = RAMOS
  0  0  0  0 15 85 |  f = WBC
```

128 ,d=2, LMT CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	504	84	%
Incorrectly Classified Instances	96	16	%
Kappa statistic	0.808		
Mean absolute error	0.0722		
Root mean squared error	0.2007		
Relative absolute error	25.9897 %		
Root relative squared error	53.8599 %		
Total Number of Instances	600		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.88	0.014	0.926	0.88	0.903	0.957	HL_60
	0.83	0.03	0.847	0.83	0.838	0.942	JURKAT
	0.79	0.056	0.738	0.79	0.763	0.954	PC3
	0.81	0.034	0.827	0.81	0.818	0.973	PCS
	0.9	0.036	0.833	0.9	0.865	0.979	RAMOS
	0.83	0.022	0.883	0.83	0.856	0.978	WBC
Weighted Avg.	0.84	0.032	0.842	0.84	0.841	0.964	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 88  7  2  0  2  1 |  a = HL_60
  7 83  8  2  0  0 |  b = JURKAT
  0  6 79 15  0  0 |  c = PC3
  0  1 18 81  0  0 |  d = PCS
  0  0  0  0 90 10 |  e = RAMOS
  0  1  0  0 16 83 |  f = WBC
```

128 ,d=2, LMT EFCS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	522	86.16	%
Incorrectly Classified Instances	78	13	%
Kappa statistic	0.844		
Mean absolute error	0.0613		
Root mean squared error	0.183		
Relative absolute error	22.0849 %		
Root relative squared error	49.0908 %		
Total Number of Instances	600		

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.89	0.018	0.908	0.89	0.899	0.982	HL_60
	0.88	0.018	0.907	0.88	0.893	0.978	JURKAT
	0.84	0.042	0.8	0.84	0.82	0.967	PC3
	0.85	0.03	0.85	0.85	0.85	0.975	PCS
	0.9	0.032	0.849	0.9	0.874	0.986	RAMOS
	0.86	0.016	0.915	0.86	0.887	0.978	WBC
Weighted Avg.	0.87	0.026	0.872	0.87	0.87	0.978	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 89  7  1  0  3  0 |  a = HL_60
  4 88  5  1  2  0 |  b = JURKAT
  0  2 84 14  0  0 |  c = PC3
  0  0 15 85  0  0 |  d = PCS
  2  0  0  0 90  8 |  e = RAMOS
  3  0  0  0 11 86 |  f = WBC
```

128 ,d=4, SVM no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	540	90 %
Incorrectly Classified Instances	60	10 %
Kappa statistic	0.88	
Mean absolute error	0.0333	
Root mean squared error	0.1826	
Relative absolute error	12 %	
Root relative squared error	48.9898 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.92	0.012	0.939	0.92	0.929	0.954	HL_60
	0.91	0.014	0.929	0.91	0.919	0.948	JURKAT
	0.85	0.034	0.833	0.85	0.842	0.908	PC3
	0.88	0.028	0.863	0.88	0.871	0.926	PCS
	0.92	0.018	0.911	0.92	0.915	0.951	RAMOS
	0.92	0.014	0.929	0.92	0.925	0.953	WBC
Weighted Avg.	0.9	0.02	0.901	0.9	0.9	0.94	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 92  5  2  0  1  0 |  a = HL_60
  5 91  3  1  0  0 |  b = JURKAT
  0  2 85 13  0  0 |  c = PC3
  0  0 12 88  0  0 |  d = PCS
  1  0  0  0 92  7 |  e = RAMOS
  0  0  0  0  8 92 |  f = WBC
```

128 ,d=4, SVM CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	529	88.1667 %
Incorrectly Classified Instances	71	11.8333 %
Kappa statistic	0.858	
Mean absolute error	0.0394	
Root mean squared error	0.1986	
Relative absolute error	14.2 %	
Root relative squared error	53.2917 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.89	0.012	0.937	0.89	0.913	0.939	HL_60
	0.91	0.024	0.883	0.91	0.897	0.943	JURKAT
	0.84	0.036	0.824	0.84	0.832	0.902	PC3
	0.84	0.032	0.84	0.84	0.84	0.904	PCS
	0.93	0.022	0.894	0.93	0.912	0.954	RAMOS
	0.88	0.016	0.917	0.88	0.898	0.932	WBC
Weighted Avg.	0.882	0.024	0.882	0.882	0.882	0.929	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 89  9  1  0  0  1 |  a = HL_60
  6 91  1  2  0  0 |  b = JURKAT
  0  2 84 14  0  0 |  c = PC3
  0  0 16 84  0  0 |  d = PCS
  0  0  0  0 93  7 |  e = RAMOS
  0  1  0  0 11 88 |  f = WBC
```

128 ,d=4, SVM EFCS feature selection.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         535               89.1667 %
Incorrectly Classified Instances        65               10.8333 %
Kappa statistic                          0.87
Mean absolute error                      0.0361
Root mean squared error                  0.19
Relative absolute error                 13      %
Root relative squared error             50.9902 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.93      0.018      0.912     0.93      0.921      0.956    HL_60
                 0.91      0.012      0.938     0.91      0.924      0.949    JURKAT
                 0.87      0.044      0.798     0.87      0.833      0.913    PC3
                 0.8       0.026      0.86      0.8       0.829      0.887    PCS
                 0.92      0.018      0.911     0.92      0.915      0.951    RAMOS
                 0.92      0.012      0.939     0.92      0.929      0.954    WBC
Weighted Avg.    0.892     0.022      0.893     0.892     0.892      0.935

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 93  6  0  0  1  0 |  a = HL_60
  7 91  2  0  0  0 |  b = JURKAT
  0  0 87 13  0  0 |  c = PC3
  0  0 20 80  0  0 |  d = PCS
  2  0  0  0 92  6 |  e = RAMOS
  0  0  0  0  8 92 |  f = WBC
```

128 ,d=4, LMT no feature selection.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         516               86      %
Incorrectly Classified Instances        84               14      %
Kappa statistic                          0.832
Mean absolute error                      0.0731
Root mean squared error                  0.1876
Relative absolute error                 26.3064 %
Root relative squared error             50.3264 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.87      0.02       0.897     0.87      0.883      0.964    HL_60
                 0.88      0.026      0.871     0.88      0.876      0.989    JURKAT
                 0.84      0.046      0.785     0.84      0.812      0.968    PC3
                 0.8       0.03       0.842     0.8       0.821      0.981    PCS
                 0.89      0.026      0.873     0.89      0.881      0.97     RAMOS
                 0.88      0.02       0.898     0.88      0.889      0.984    WBC
Weighted Avg.    0.86      0.028      0.861     0.86      0.86       0.976

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 87  9  1  0  2  1 |  a = HL_60
  8 88  2  2  0  0 |  b = JURKAT
  0  3 84 13  0  0 |  c = PC3
  0  0 20 80  0  0 |  d = PCS
  2  0  0  0 89  9 |  e = RAMOS
  0  1  0  0 11 88 |  f = WBC
```

128 ,d=4, LMT CFS feature selection.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         493               82.1667 %
Incorrectly Classified Instances       107               17.8333 %
Kappa statistic                          0.786
Mean absolute error                      0.0911
Root mean squared error                  0.2075
Relative absolute error                 32.7936 %
Root relative squared error             55.6901 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.85      0.016      0.914     0.85      0.881      0.977    HL_60
                 0.89      0.034      0.84      0.89      0.864      0.978    JURKAT
                 0.73      0.054      0.73      0.73      0.73       0.957    PC3
                 0.76      0.048      0.76      0.76      0.76       0.968    PCS
                 0.85      0.032      0.842     0.85      0.846      0.985    RAMOS
                 0.85      0.03       0.85      0.85      0.85       0.98     WBC
Weighted Avg.    0.822     0.036      0.823     0.822     0.822      0.974

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 85 10  2  0  2  1 |  a = HL_60
  7 89  3  1  0  0 |  b = JURKAT
  0  4 73 23  0  0 |  c = PC3
  0  2 22 76  0  0 |  d = PCS
  1  0  0  0 85 14 |  e = RAMOS
  0  1  0  0 14 85 |  f = WBC
```

128 ,d=4, LMT EFCS feature selection.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         523               87.1667 %
Incorrectly Classified Instances        77               12.8333 %
Kappa statistic                          0.846
Mean absolute error                      0.0635
Root mean squared error                  0.189
Relative absolute error                 22.8697 %
Root relative squared error             50.7029 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.87      0.034      0.837     0.87      0.853      0.963    HL_60
                 0.87      0.022      0.888     0.87      0.879      0.969    JURKAT
                 0.86      0.03       0.851     0.86      0.856      0.971    PC3
                 0.87      0.026      0.87      0.87      0.87       0.977    PCS
                 0.88      0.024      0.88      0.88      0.88       0.976    RAMOS
                 0.88      0.018      0.907     0.88      0.893      0.962    WBC
Weighted Avg.    0.872     0.026      0.872     0.872     0.872      0.97

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 87  7  2  0  3  1 |  a = HL_60
 12 87  0  1  0  0 |  b = JURKAT
  0  2 86 12  0  0 |  c = PC3
  0  0 13 87  0  0 |  d = PCS
  2  2  0  0 88  8 |  e = RAMOS
  3  0  0  0  9 88 |  f = WBC
```

256 ,d=2, SVM no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	541	90.1667 %
Incorrectly Classified Instances	59	9.8333 %
Kappa statistic	0.882	
Mean absolute error	0.0328	
Root mean squared error	0.181	
Relative absolute error	11.8 %	
Root relative squared error	48.5798 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.94	0.006	0.969	0.94	0.954	0.967	HL_60
	0.92	0.016	0.92	0.92	0.92	0.952	JURKAT
	0.82	0.04	0.804	0.82	0.812	0.89	PC3
	0.87	0.026	0.87	0.87	0.87	0.922	PCS
	0.93	0.016	0.921	0.93	0.925	0.957	RAMOS
	0.93	0.014	0.93	0.93	0.93	0.958	WBC
Weighted Avg.	0.902	0.02	0.902	0.902	0.902	0.941	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
94	3	2	0	1	0	a = HL_60
3	92	5	0	0	0	b = JURKAT
0	5	82	13	0	0	c = PC3
0	0	13	87	0	0	d = PCS
0	0	0	0	93	7	e = RAMOS
0	0	0	0	7	93	f = WBC

256 ,d=2, SVM CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	525	87.5 %
Incorrectly Classified Instances	75	12.5 %
Kappa statistic	0.85	
Mean absolute error	0.0417	
Root mean squared error	0.2041	
Relative absolute error	15 %	
Root relative squared error	54.7723 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.91	0.012	0.938	0.91	0.924	0.949	HL_60
	0.93	0.026	0.877	0.93	0.903	0.952	JURKAT
	0.79	0.042	0.79	0.79	0.79	0.874	PC3
	0.84	0.03	0.848	0.84	0.844	0.905	PCS
	0.92	0.024	0.885	0.92	0.902	0.948	RAMOS
	0.86	0.016	0.915	0.86	0.887	0.922	WBC
Weighted Avg.	0.875	0.025	0.876	0.875	0.875	0.925	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
91	5	3	0	0	1	a = HL_60
3	93	3	1	0	0	b = JURKAT
1	6	79	14	0	0	c = PC3
0	1	15	84	0	0	d = PCS
1	0	0	0	92	7	e = RAMOS
1	1	0	0	12	86	f = WBC

256 ,d=2, SVM EFCS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	545	90.8333 %
Incorrectly Classified Instances	55	9.1667 %
Kappa statistic	0.89	
Mean absolute error	0.0306	
Root mean squared error	0.1748	
Relative absolute error	11 %	
Root relative squared error	46.9042 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.97	0.008	0.96	0.97	0.965	0.981	HL_60
	0.91	0.004	0.978	0.91	0.943	0.953	JURKAT
	0.88	0.052	0.772	0.88	0.822	0.914	PC3
	0.81	0.024	0.871	0.81	0.839	0.893	PCS
	0.95	0.012	0.941	0.95	0.945	0.969	RAMOS
	0.93	0.01	0.949	0.93	0.939	0.96	WBC
Weighted Avg.	0.908	0.018	0.912	0.908	0.909	0.945	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
97	1	2	0	0	0	a = HL_60
3	91	5	1	0	0	b = JURKAT
0	1	88	11	0	0	c = PC3
0	0	19	81	0	0	d = PCS
0	0	0	0	95	5	e = RAMOS
1	0	0	0	6	93	f = WBC

256 ,d=2, LMT no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	516	86 %
Incorrectly Classified Instances	84	14 %
Kappa statistic	0.832	
Mean absolute error	0.0618	
Root mean squared error	0.1929	
Relative absolute error	22.2558 %	
Root relative squared error	51.7713 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.87	0.018	0.906	0.87	0.888	0.962	HL_60
	0.86	0.032	0.843	0.86	0.851	0.956	JURKAT
	0.78	0.05	0.757	0.78	0.768	0.955	PC3
	0.81	0.03	0.844	0.81	0.827	0.967	PCS
	0.94	0.026	0.879	0.94	0.908	0.985	RAMOS
	0.9	0.012	0.938	0.9	0.918	0.983	WBC
Weighted Avg.	0.86	0.028	0.861	0.86	0.86	0.968	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
87	7	2	0	3	1	a = HL_60
8	86	5	1	0	0	b = JURKAT
0	8	78	14	0	0	c = PC3
0	1	18	81	0	0	d = PCS
1	0	0	0	94	5	e = RAMOS
0	0	0	0	10	90	f = WBC

256 ,d=2, LMT CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	498	83 %
Incorrectly Classified Instances	102	17 %
Kappa statistic	0.796	
Mean absolute error	0.0753	
Root mean squared error	0.2092	
Relative absolute error	27.1218 %	
Root relative squared error	56.1249 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.89	0.024	0.881	0.89	0.886	0.957	HL_60
	0.84	0.034	0.832	0.84	0.836	0.948	JURKAT
	0.74	0.054	0.733	0.74	0.736	0.952	PC3
	0.77	0.036	0.811	0.77	0.79	0.962	PCS
	0.9	0.034	0.841	0.9	0.87	0.977	RAMOS
	0.84	0.022	0.884	0.84	0.862	0.977	WBC
Weighted Avg.	0.83	0.034	0.83	0.83	0.83	0.962	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 89  7  1  0  1  2 |  a = HL_60
 10 84  5  1  0  0 |  b = JURKAT
  1  8 74 17  0  0 |  c = PC3
  0  2 21 77  0  0 |  d = PCS
  1  0  0  0 90  9 |  e = RAMOS
  0  0  0  0 16 84 |  f = WBC
```

256 ,d=2, LMT EFCS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	517	86.1667 %
Incorrectly Classified Instances	83	13.8333 %
Kappa statistic	0.834	
Mean absolute error	0.0615	
Root mean squared error	0.1906	
Relative absolute error	22.1285 %	
Root relative squared error	51.1313 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.91	0.022	0.892	0.91	0.901	0.969	HL_60
	0.86	0.03	0.851	0.86	0.856	0.962	JURKAT
	0.82	0.052	0.759	0.82	0.788	0.956	PC3
	0.81	0.026	0.862	0.81	0.835	0.971	PCS
	0.89	0.018	0.908	0.89	0.899	0.988	RAMOS
	0.88	0.018	0.907	0.88	0.893	0.978	WBC
Weighted Avg.	0.862	0.028	0.863	0.862	0.862	0.971	

=== Confusion Matrix ===

```
  a  b  c  d  e  f   <-- classified as
 91  5  2  0  0  2 |  a = HL_60
  6 86  5  1  2  0 |  b = JURKAT
  0  6 82 12  0  0 |  c = PC3
  0  1 18 81  0  0 |  d = PCS
  3  1  0  0 89  7 |  e = RAMOS
  2  2  1  0  7 88 |  f = WBC
```

256, d=4, SVM no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	544	90.6667 %
Incorrectly Classified Instances	56	9.3333 %
Kappa statistic	0.888	
Mean absolute error	0.0311	
Root mean squared error	0.1764	
Relative absolute error	11.2 %	
Root relative squared error	47.3286 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.93	0.008	0.959	0.93	0.944	0.961	HL_60
	0.93	0.012	0.939	0.93	0.935	0.959	JURKAT
	0.83	0.036	0.822	0.83	0.826	0.897	PC3
	0.87	0.032	0.845	0.87	0.857	0.919	PCS
	0.95	0.016	0.922	0.95	0.936	0.967	RAMOS
	0.93	0.008	0.959	0.93	0.944	0.961	WBC
Weighted Avg.	0.907	0.019	0.908	0.907	0.907	0.944	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
93	4	2	0	1	0	a = HL_60
3	93	3	1	0	0	b = JURKAT
0	2	83	15	0	0	c = PC3
0	0	13	87	0	0	d = PCS
1	0	0	0	95	4	e = RAMOS
0	0	0	0	7	93	f = WBC

256, d=4, SVM CFS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	518	86.3333 %
Incorrectly Classified Instances	82	13.6667 %
Kappa statistic	0.836	
Mean absolute error	0.0456	
Root mean squared error	0.2134	
Relative absolute error	16.4 %	
Root relative squared error	57.2713 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.86	0.028	0.86	0.86	0.86	0.916	HL_60
	0.86	0.028	0.86	0.86	0.86	0.916	JURKAT
	0.84	0.036	0.824	0.84	0.832	0.902	PC3
	0.84	0.032	0.84	0.84	0.84	0.904	PCS
	0.91	0.024	0.883	0.91	0.897	0.943	RAMOS
	0.87	0.016	0.916	0.87	0.892	0.927	WBC
Weighted Avg.	0.863	0.027	0.864	0.863	0.863	0.918	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
86	13	1	0	0	0	a = HL_60
11	86	2	1	0	0	b = JURKAT
0	1	84	15	0	0	c = PC3
1	0	15	84	0	0	d = PCS
1	0	0	0	91	8	e = RAMOS
1	0	0	0	12	87	f = WBC

256, d=4, SVM EFCS feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	537	89.5 %
Incorrectly Classified Instances	63	10.5 %
Kappa statistic	0.874	
Mean absolute error	0.035	
Root mean squared error	0.1871	
Relative absolute error	12.6 %	
Root relative squared error	50.1996 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.93	0.02	0.903	0.93	0.916	0.955	HL_60
	0.91	0.008	0.958	0.91	0.933	0.951	JURKAT
	0.88	0.044	0.8	0.88	0.838	0.918	PC3
	0.82	0.024	0.872	0.82	0.845	0.898	PCS
	0.9	0.016	0.918	0.9	0.909	0.942	RAMOS
	0.93	0.014	0.93	0.93	0.93	0.958	WBC
Weighted Avg.	0.895	0.021	0.897	0.895	0.895	0.937	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
93	4	2	0	1	0	a = HL_60
7	91	2	0	0	0	b = JURKAT
0	0	88	12	0	0	c = PC3
0	0	18	82	0	0	d = PCS
3	0	0	0	90	7	e = RAMOS
0	0	0	0	7	93	f = WBC

256, d=4, LMT no feature selection.

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	521	86.8333 %
Incorrectly Classified Instances	79	13.1667 %
Kappa statistic	0.842	
Mean absolute error	0.064	
Root mean squared error	0.1814	
Relative absolute error	23.0395 %	
Root relative squared error	48.6868 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.87	0.016	0.916	0.87	0.892	0.972	HL_60
	0.93	0.022	0.894	0.93	0.912	0.992	JURKAT
	0.82	0.044	0.788	0.82	0.804	0.975	PC3
	0.8	0.032	0.833	0.8	0.816	0.98	PCS
	0.92	0.028	0.868	0.92	0.893	0.977	RAMOS
	0.87	0.016	0.916	0.87	0.892	0.986	WBC
Weighted Avg.	0.868	0.026	0.869	0.868	0.868	0.98	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
87	8	2	0	2	1	a = HL_60
6	93	0	1	0	0	b = JURKAT
0	3	82	15	0	0	c = PC3
0	0	20	80	0	0	d = PCS
1	0	0	0	92	7	e = RAMOS
1	0	0	0	12	87	f = WBC

256, d=4, LMT CFS feature selection

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	488	81.3333 %
Incorrectly Classified Instances	112	18.6667 %
Kappa statistic	0.776	
Mean absolute error	0.0923	
Root mean squared error	0.2181	
Relative absolute error	33.239 %	
Root relative squared error	58.5143 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.82	0.02	0.891	0.82	0.854	0.975	HL_60
	0.85	0.038	0.817	0.85	0.833	0.964	JURKAT
	0.75	0.052	0.743	0.75	0.746	0.95	PC3
	0.77	0.044	0.778	0.77	0.774	0.953	PCS
	0.88	0.044	0.8	0.88	0.838	0.978	RAMOS
	0.81	0.026	0.862	0.81	0.835	0.963	WBC
Weighted Avg.	0.813	0.037	0.815	0.813	0.813	0.964	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
82	14	1	0	2	1	a = HL_60
9	85	3	1	2	0	b = JURKAT
0	4	75	21	0	0	c = PC3
0	1	22	77	0	0	d = PCS
0	0	0	0	88	12	e = RAMOS
1	0	0	0	18	81	f = WBC

256, d=4, LMT EFCS feature selection

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	495	82.5 %
Incorrectly Classified Instances	105	17.5 %
Kappa statistic	0.79	
Mean absolute error	0.0772	
Root mean squared error	0.2141	
Relative absolute error	27.7895 %	
Root relative squared error	57.4525 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.87	0.034	0.837	0.87	0.853	0.964	HL_60
	0.84	0.028	0.857	0.84	0.848	0.964	JURKAT
	0.73	0.06	0.709	0.73	0.719	0.944	PC3
	0.74	0.044	0.771	0.74	0.755	0.965	PCS
	0.91	0.026	0.875	0.91	0.892	0.981	RAMOS
	0.86	0.018	0.905	0.86	0.882	0.969	WBC
Weighted Avg.	0.825	0.035	0.826	0.825	0.825	0.965	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
87	8	1	0	2	2	a = HL_60
12	84	3	0	1	0	b = JURKAT
0	4	73	22	0	1	c = PC3
0	1	25	74	0	0	d = PCS
3	0	0	0	91	6	e = RAMOS
2	1	1	0	10	86	f = WBC