Peripheral blood smear image analysis: A comprehensive review

Emad A. Mohammed, Mostafa M. A. Mohamed, Behrouz H. Far, Christopher Naugler

Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada; Department of Pathology and Laboratory Medicine, Calgary Laboratory Services, University of Calgary, Calgary, Alberta, Canada; Department of Biomedical Engineering, Faculty of Engineering, Helwan University, Cairo, Egypt

E-mail: *Christopher Naugler - christopher.naugler@cls.ab.ca
*Corresponding author

Received: 07 November 2013 Accepted: 06 February 2014 Published: 28 March 2014

Abstract

Peripheral blood smear image examination is a part of the routine work of every laboratory. The manual examination of these images is tedious, time-consuming and suffers from interobserver variation. This has motivated researchers to develop different algorithms and methods to automate peripheral blood smear image analysis. Image analysis itself consists of a sequence of steps consisting of image segmentation, features extraction and selection and pattern classification. The image segmentation step addresses the problem of extraction of the object or region of interest from the complicated peripheral blood smear image. Support vector machine (SVM) and artificial neural networks (ANNs) are two common approaches to image segmentation. Features extraction and selection aims to derive descriptive characteristics of the extracted object, which are similar within the same object class and different between different objects. This will facilitate the last step of the image analysis process: pattern classification. The goal of pattern classification is to assign a class to the selected features from a group of known classes. There are two types of classifier learning algorithms: supervised and unsupervised. Supervised learning algorithms predict the class of the object under test using training data of known classes. The training data have a predefined label for every class and the learning algorithm can utilize this data to predict the class of a test object. Unsupervised learning algorithms use unlabeled training data and divide them into groups using similarity measurements. Unsupervised learning algorithms predict the group to which a new test object belong to, based on the training data without giving an explicit class to that object. ANN, SVM, decision tree and K-nearest neighbor are possible approaches to classification algorithms. Increased discrimination may be obtained by combining several classifiers together.

Key words: Feature extraction, feature selection, microscopic image analysis, peripheral blood smear, segmentation

INTRODUCTION

The screening of prepared blood films is tedious, time-consuming and subject to inter- and intra-observer variation. This has implications for both laboratory resources as well as diagnostic accuracy. As a result, many researchers have addressed problems related to microscopic image analysis of peripheral blood smears and indeed developing image analysis systems for computer-assisted interpretations of peripheral blood...
smears is of a great importance. The goal of this paper is to provide a comprehensive review of the algorithms and methodologies used to analyze peripheral blood smear images. Figure 1 illustrates the basic steps of the workflow in a peripheral blood smear image analysis system. It starts with a segmentation step, which targets the isolation of the desired cell from the complicated blood smear image into a separate mask (cell mask) and may further divide the cell mask into semantic regions (nucleus mask and cytoplasm mask). The segmentation step provides the input data for the rest of the algorithm and thus a high accuracy segmentation algorithm must be used to guarantee the success of the subsequent image analysis algorithms. The next step is to process the resulting masks by computational tools that can measure, extract and select quantitative features useful for more objective and accurate detection, diagnosis and prognosis of the image under test. The feature selection step is crucial for the success of the cell classification process. A good feature selection algorithm (FSA) manages to select the most distinctive features for every cell class. These features must be very similar within cells of the same class and different for different classes. A significant problem exists for any classifier if the selected features for the different classes are greatly overlapped (i.e. have almost the same feature values). The last step is to classify the extracted features, which aims to assign a class to the features of the object of interest. The classification process starts with a training phase for a classifier model with known cell classes. The training process aims to find the optimal model parameter to minimize the classification error. The final step is to test the trained classifier model with test cells (not used in the training process) to validate the model parameter (i.e. sensitivity and specificity of the classifier model). Finally, the trained classifier model can be used to classify unknown class cells if the model shows high sensitivity and specificity. More than one classifier may be trained on the same dataset and the results of all the classifiers are aggregated to form a better classification.

**PERIPHERAL BLOOD SMEAR MICROSCOPIC IMAGE SEGMENTATION**

Segmentation is the process of correctly and accurately extracting different parts of an image. Leukocytes exhibit wide variations of cell morphology and size that make them difficult to be segmented accurately. Many studies have addressed this problem.

There are many techniques that can be used to address the problem of blood smear image segmentation. In this section a brief introduction to these techniques is introduced and then followed by the algorithms that utilize these techniques.

Multispectral imaging is an imaging technology that can record spectral and spatial information of a specimen. It consists of gray images acquired at a specific narrow band of wavelengths. Watershed algorithm is a clustering algorithm used extensively in image processing algorithms. It is used to group the pixels of an object in a separate mask to further isolate it from the image for the purpose of analysis.

Support vector machine (SVM) is a supervised learning algorithm that can analyze data and recognize patterns. Artificial neural network (ANN) is another type of machine learning algorithms. It can be used as a supervised or unsupervised learning method.

Multispectral white blood cell (WBC) segmentation using a SVM showed results insensitive to blood smear staining and illumination condition, but with low nucleus segmentation accuracy due to variation of the nucleus color. Other work using SVM obtained a
98.9% maximum cell accuracy using feature scale-space filtering and watershed clustering. However, the results show that the method based on an RGB color space did not give accurate results. An online learning system for accurate cell segmentation by simulating the visual attention of the human eyes has also been proposed. The results are promising however, it requires a great deal of processing and it may not be suitable for limited resource systems.

Over and under segmentation are problems arising when a segmented object contains parts of other objects or there is some missing part of the segmented object. This is usually the case when using clustering algorithms such as the watershed algorithm for segmentation. Research has proposed an automatic WBC segmentation using stepwise merging rules and a gradient vector flow snake that reduces the over-segmentation problem by 10.31% and the under-segmentation by 1.32%, but the algorithm is iterative and consumes a lot of system resources.

The SVM is widely used in many areas including pattern recognition, image processing and bioinformatics. The majority of studies utilizing SVM in hematopathology are dedicated to normal WBCs and acute lymphocytic leukemia detection. For example, previous work has applied an automated approach to clinical image segmentation using pathological modeling, a principal component analysis (PCA) and an SVM. Remarkable results have been achieved by applying the PCA to the extracted features and the results are used to train an SVM. However, the algorithm is iterative and as with other iterative processes is resource intensive. Many studies have addressed the problem of reducing the SVM training dataset (to reduce the number of support vectors used in the classification process) without sacrificing performance. Most of these studies are based on the k-means clustering algorithm. Liu and Feng present a new algorithm named kernel bisecting k-means and sample removal as a sampling preprocessing step for SVM training to improve the scalability. Other approaches such as a chunking algorithm, a decomposition algorithm, a sequential minimal optimization, and an “SVM light” algorithm have been used for SVM training. These algorithms tend to reduce the substantial training task into a series of smaller sub-tasks in order to decrease the SVM training time. However, the computational time still needs further improvement in practice.

Global image arithmetic (e.g. image subtraction and image addition) is used for localization and segmentation of lymphoblast cells from peripheral blood smear images. It gives accuracy of 90-95% in restoring the lymphoblast pixels from the original image. This is due to the color inconsistency of the lymphoblast cells. Color image segmentation using SVM and fuzzy C-means has been proposed and has the advantage of segmenting any type of image accurately and quickly. However, it suffers the problems of over and under-segmentation.

The ANN has been used extensively during the past few decades. It has many applications in pattern classification in addition to image segmentation. Jaffar et al. used a self-organizing map (SOM) neural network along with wavelets to segment WBCs; however again, this was computationally expensive. The results show that if the SOM training is performed on the wavelet-transformed image, it reduces the SOM training time and makes more compact segments. This method has the following advantages: It yields more homogeneous regions than those of other methods for color images, it reduces spurious blobs and it removes noisy spots. However, the method is still computationally expensive. Many modified approaches to the classical ANN application have been developed in the literature to speed up the classification process and may be used in both pattern classification and image segmentation. An example is a proposal for a pulse-coupled neural network (PCNN) with multichannel PCNN linking and feeding fields for color image segmentation. Pulse-based radial basis function units are introduced into the model neurons of PCNN to determine the fast links among neurons with respect to their spectral feature vectors and spatial proximity. However, the performance of the proposed method still needs enhancements comparable to those of other popular image segmentation algorithms for the segmentation of noisy images.

In our previous work, we proposed a segmentation method based on the watershed algorithm and optimal threshold using Otsu’s method for segmentation of chronic lymphocytic leukemia (CCL) cell segmentation. In this research, we tested 140 microscopic lymphocyte images (normal and CLL) and the algorithm obtained 99.92% maximum accuracy for nucleus segmentation and 99.85% maximum accuracy for cell segmentation. The cytoplasm can be extracted with a 99.63% maximum accuracy with a simple mask subtraction. We reported that a 1% reduction of the local minima of the watershed transform controlled the over and under segmentation problems. However, the algorithm suffers from the “occlusion” problem when lymphocytes are tightly bound to the surrounding red blood cells and the percentage of local minima reduction is dependent on the image quality.

In other previous work, we proposed a method based on an SVM classifier model and k-mean clustering algorithm. The proposed method overcomes the occlusion problem and over and under-segmentation problems are significantly reduced. In this research, we used 440 lymphocyte images (normal and CLL), in which 140 images are used for segmentation accuracy measurement and 12 images for SVM training. The algorithm obtained...
98.43% maximum accuracy for nucleus segmentation and 98.69% for cell segmentation. The cytoplasm region could be extracted by 99.85% maximum accuracy with a simple mask subtraction. This proposed method has some limitations when two or more lymphocytes are touching as these are identified as one entity. More inference rules must be utilized to overcome this problem. However, the overall performance of the algorithm is quite promising for accurate lymphocyte color cell segmentation.

Table 1 shows a time comparison of the segmentation algorithms used in [2,14,18]. The execution time is recorded in seconds and it represents the time required by the algorithm to extract the cell from the complicated background and further divide the cell into the nucleus and cytoplasm masks. The execution time recorded for the image arithmetic represents only time to segment the nucleus. The execution time is measured using Intel® quad core CPU i5 2.53 GHz, 4GB DDR RAM PC Windows® 7 64-bit using MATLAB® 2011b. The results show a superior performance of SVM based method.

| Execution time | SVM based segmentation method [20] | Watershed based segmentation method [2] | Global image arithmetic segmentation based method for the nucleus only [14] |
|----------------|-----------------------------------|----------------------------------------|----------------------------------------------------------------------|
| Total segmentation time | 16.044 seconds | 63.78 seconds | 4.89 seconds |
| Minimum          | 0.098 seconds   | 0.326 seconds  | 0.022 seconds |
| Average          | 0.115 seconds   | 0.455 seconds  | 0.035 seconds |
| Maximum          | 0.275 seconds   | 0.802 seconds  | 0.103 seconds |
| Standard deviation| ±0.02           | ±0.061         | ±0.011          |

SVM: Support vector machine

Table 1: Segmentation time in seconds for 140 blood images. Comparisons between the segmentation methods used in [2,14,18]. The time recorded is for all the segmentation processes (cell, nucleus and cytoplasm); except for image arithmetic, which includes time for nucleus segmentation only.

FEATURE EXTRACTION AND FEATURE SELECTION

The potential for automating accurate diagnostic decisions in pathology is limited by the lack of objective, definitive and measureable features for detecting and characterizing diseases. Peripheral blood smears are routinely investigated for abnormalities; however, the delicate visible differences exhibited by some disorders can lead to a significant number of false negatives during microscopic examination of the peripheral blood smears. Figure 2 shows the workflow of a typical n classes supervised classification that involves training and testing phases to classify a test image.

Measuring digital object properties has been a subject of study since the early 1970s and is has benefited from the culmination of considerable development [9,21]. It can also be used to discriminate between objects by measuring and comparing their properties. Feature extraction is the process of converting a given mask (cell, nucleus and cytoplasm masks) into a set of measurements. There are many features that can be measured for a given object in an image, [9] but these fall into three primary categories: Geometric features, histogram based features and intensity based features. Table 2 shows all the possible features that can be measured for a segmented cell.

Feature selection is the process of selecting a subset of these extracted features that have minimum redundancy and maximum relevance to the object of interest. The goal of the FSA is to reduce the dimensionality of the classifier input data by selecting the most distinctive features, which maximize the correct classification rate (CCR). The resulting classifier is thereby improved either in terms of learning speed, generalization capacity or simplicity of the representation.

There are two basic types of FSA. The first is the filter type, in which statistical analysis is used to rank the features according to the information represented by the features. The performance of a single feature classifier can be used to select features according to their individual predictive power. The predictive power of the feature can be measured in terms of error rate. Ranking criteria based on the CCR cannot distinguish between the top ranking variables where there are a large number of features that separate the data perfectly. A FSA based on filter type requires the use of the features probability density functions that are not easily computed. However, the probabilities can be estimated from the frequency counts in the case of discrete features. Noise in data can affect

Figure 2: Procedures for supervised learning technique, showing the overall training and classification procedures for (n) classes.
In a feed-forward network, the neurons are generally grouped into layers. Signals flow from the input layer through the output layer through unidirectional connections. The neurons are connected from one layer to the next, but not within the same layer. In recurrent networks, the output of some neurons is fed back to the network topology, which includes the number of units and their connectivity is typically determined first, often by trial and error, prior to the start of training. Each network, however, requires a methodology to search the space of all possible feature subsets, which is computationally expensive. There are two types of the search algorithm for the wrapper type: Sequential forward selection (SFS) and sequential backward selection (SBS). In SFS, features are progressively incorporated into larger subsets, whereas SBS starts with the set of all features and progressively eliminates the least promising ones.[23]

### CLASSIFICATION AND MULTIPLE CLASSIFIER SYSTEMS (MCSs)

**ANN**

ANNs[9] are powerful tools that can be trained to solve problems in a way similar to the human brain. They gather knowledge by detecting patterns and relationships in data and learn through experience. An ANN might consist of several thousand artificial neurons and the output of one neuron becomes an input to another neuron. There are several types of ANNs according to their structure and learning algorithms. The simplest neural network, called a perceptron, takes as input a real-valued vector of feature values, obtains a linear combination of them and outputs a (1) if the combination is greater than a threshold and (−1) otherwise. This corresponds to the following linear discriminant function:

\[
g(x) = w^T x - w_0
\]

where \(x\) is a feature vector, \(w\) is the vector of weights and \(w_0\) is the threshold.

Thus, \(g(x) = 0\) is the surface which separates items in class C1: \(g(x) > 0\) from items in class C2: \(g(x) < 0\), enabling the perceptron to act as a linear classifier for a two-class problem when the two classes are linearly separable. The perceptron must be trained, i.e. the weight vector \(w\) is obtained, before it can be applied to classify an item without a class label. A simple approach to obtain the weight vector is to start with random weights, apply the perceptron to classify a data item in the training set and modify the weights whenever the item is misclassified. A gradient descent approach is used to find the weights, which best separate the two classes in the training data[20] if the classes are not linearly separable. According to their structure the ANNs can be classified as feed-forward networks and recurrent networks.[24] In a feed-forward network, the neurons are generally grouped into layers. Signals flow from the input layer through the output layer through unidirectional connections. The neurons are connected from one layer to the next, but not within the same layer. In recurrent networks, the output of some neurons is fed back to the same neurons or to neurons in a preceding layer. The network topology, which includes the number of units and their connectivity is typically determined first, often by trial and error, prior to the start of training. Each

| Feature       | Interpretation                                      |
|---------------|-----------------------------------------------------|
| Area          | Geometric features: Scalar specifies the actual number of pixels in the region |
| Perimeter     | Geometric features: Scalar defines the distance around the boundary of the region of interest |
| Eccentricity  | Geometric features: Scalar that specifies the eccentricity of the ellipse that has the same second-moments as the region of interest |
| Equivalent diameter | Geometric features: Scalar specifies the diameter of a circle with the same area as the region |
| Extent        | Geometric features: Scalar that specifies the ratio of pixels in the region to pixels in the total bounding box |
| Bounding box  | Geometric features: The smallest rectangle containing the region of interest |
| Convex area   | Geometric features: Scalar that specifies the number of pixels in convex image |
| Filled area   | Geometric features: Scalar specifying the number of on pixels in filled image |
| Major axis length | Geometric features: Scalar specifying the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the region of interest |
| Minor axis length | Geometric features: Scalar defines the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the region of interest |
| Solidity      | Scalar specifying the proportion of the pixels in the convex hull that are also in the region |
| Compactness   | Geometric features: Scalar measures the efficiency of a contour to contain a given area |
| Mean          | Intensity based features: Scalar represents the average gray level value of the a given mask |
| Variance      | Intensity based features: Scalar represents a measure of inhomogeneity (second order moment) |
| Energy        | Histogram based features: Scalar represents the amounts of variation within a given mask |
| Entropy       | Histogram based features: Scalar represents the measure of non-uniformity in the image |
| Skewness      | Histogram based features: Scalar represents the symmetry of the PDF distribution of the gray level in a given mask |
| Kurtosis      | Histogram based features: Scalar represents the uniformity of the PDF distribution of the gray level in a given mask |

PDF: Probability density function
unit is a sigmoidal unit and is a smoothly differentiable function:

\[ f(x) = \frac{1}{1+e^{-x}} \]  

(2)

The commonly used learning algorithm for a multilayer network using sigmoidal units is the conjugate gradient decent back-propagation algorithm.[24] As in the case of a single perceptron, the error function is defined as the sum of the errors over all outputs units. The updates of the weight vectors are now more complex as there are multiple units as well as a layer of hidden units. The derivation of the weights, as well as various practical issues related to the implementation and application of the back-propagation algorithm, can be found in.[24] If the network has too few neurons, it may not be able to learn complex patterns; if it has too many neurons, it is likely to over-fit the data.

ANN learning algorithms can be classified to supervised learning, unsupervised learning and reinforcement learning. In the supervised model, the ANN requires the output in order to adjust its weight. In the unsupervised model, the ANN does not require the output, the ANN adapts purely in response to its input. The reinforcement learning algorithm employs a critic to evaluate the goodness of the neural network output corresponding to a given input.[24] The multi-layer perceptron (MLP) is a well-known type of ANN, which is usually used in classification problems. In MLP, the neurons are grouped in many layers as shown in Figure 3. In this approach, during the training process of the network, the network compares its actual results with the desired output and then computes the error by the function:

\[ \varepsilon(t) = \frac{1}{2} \sum (d_i(t) - y_i(t))^2 \]  

(3)

which represents the mean square difference between the network output and the desired output. Through the back propagation algorithm the error will be presented many times to the input of the forward activation place and the process will continue until the actual outputs get closer to the desired output.

SVM

The SVM is a powerful tool for classification that can be considered as an alternative to the MLP. The SVM was first introduced in 1992.[8] The basic idea of the SVM is to find the linear classifier known as the hyper-plane that separates two classes. Figure 4 shows the linear SVM classifier and the support vectors used to define the separation margin of the classifier. As it has shown in Figure 4, there is an ideal separating classifier – the hyper-plane – which increases the space between it and the nearest dataset points of different classes as much as possible. For separable classes as shown in Figure 4, an SVM classifier computes a decision function having a maximal margin “M” with respect to the two classes. There are two planes touching the boundary of dataset, \(w^Tx + b = +1\) and \(w^Tx + b = -1\); \(w\) is a vector perpendicular on the plane \(w^Tx + b = +1\). The maximum margin of the best classifier can expressed as \(M = \frac{1}{\|w\|}\). The decision boundaries can be found by solving the following constrained optimizing problem: Minimize \(\frac{1}{2}\|w\|^2\) subject to:

\[ y_i(w^Tx_i+b) \geq 1, \forall i \]  

(4)

where \(y_i\) is the hyper-plane equation and \(x_i\) is the data point set or basically the feature space. The Lagrange function formulation for this optimization problem is given by:

\[ L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum \alpha_i (y_i(w^Tx_i+b)−1), \alpha_i \geq 0 \forall i \]  

(5)

By setting the derivative of the Lagrange function to zero:

\[ \frac{\partial}{\partial b} L(w,b,\alpha) = 0, \frac{\partial}{\partial w} L(w,b,\alpha) = 0 \]  

(6)

Figure 3: Construction of a multi-layer perceptron artificial neural network with one input layer, two hidden layers and one output layer.
The optimization problem can be expressed by:

$$\max_{\alpha} w = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$  \hfill (8)$$

$x_i$ with the non-zero value of $\alpha_i$ are called support vectors. In case of a non-linearly separable dataset as shown in Figure 5, the positive slack variable $\xi_i$, which controls the constrained condition in the hyper-plane equation, is introduced leading to a soft margin classifier:

$$\min \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{n} \xi_i$$  \hfill (9)$$

$$\max_{\alpha} w = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$  \hfill (10)$$

where $C \geq 0$

The parameter $C$ describes the trade-off between the maximal margin and the correct classification.\[25\] The parameter $C$ is selected such that a larger $C$ corresponds to assigning a higher penalty to errors. To solve non-linear classification problems, the linear SVM are applied to high dimensional spaces as shown in Figure 5, which transform the data into a high dimensional space. This means transforming from $(x_i, x_j)$ to $(\phi(x_i), \phi(x_j))$ which is known as the kernel trick. This will lead to the following optimization problem:

$$\max_{\alpha} w = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$  \hfill (11)$$

The common kernel functions are polynomial with degree ($d$), the radial base function with width sigma ($\sigma$) and the sigmoid with parameter ($k$).

$$\phi(x_i, x_j) = ((x_i \cdot x_j) + C)^d$$  \hfill (12)$$

$$\phi(x_i, x_j) = e \left( \frac{-|x_i - x_j|^2}{2\sigma^2} \right)$$  \hfill (13)$$

Decision tree

Decision tree\[26\] is a popular technique used in classification problems, as they are accurate, relatively simple to implement, produce a model that is easy to interpret and understand and have built-in dimension reduction. A decision tree is a structure that is either a leaf, indicating a class, or a decision node that specifies some test to be carried out on a feature (or a combination of features), with a branch and sub-tree for each possible outcome of the test. The traditional version of a decision tree algorithm creates tests at each node that involve a single feature. As the tests at each node are very simple, it is easy for the domain expert to interpret the tree. There are several variants of oblique decision tree, which differ in how the linear combination is obtained.

K-nearest neighbor (K-NN)

The K-nearest neighbor (K-NN) algorithm is considered to be the simplest classifier model.\[27\] This algorithm belongs to the category of instance-based learning. In such techniques, the learning occurs only when the data items are to be classified. The classification algorithm typically classifies the data items as belonging to the nearest class that is represented by a set of measured features. The K-NN algorithm assigns to an unlabeled item the most frequently occurring class label among the K most similar data items. The similar data items are obtained using different distance metrics between the feature vectors such as Euclidean distance and city-block distance measurement metrics. The K-NN can also be applied using weights, where the neighbors who are closer to the query item have larger weights.

Adaptive boosting (AdaBoost)

AdaBoost is founded on the notion of using a set of weak classifier models and pooling the classification results to produce a stronger composite classifier model. In the sequence of weak models used, each classifier focuses its discriminatory power on the training samples misclassified by the previous weak classifier. The main reference for the AdaBoost algorithm is the original
paper by Freund and Schapire.28 AdaBoost maintains a probability distribution over all the training samples. This distribution is modified iteratively with each selection of a weak classifier. Initially, the probability distribution is uniform over the training samples. The weak classifier, which is chosen at iteration \( t \) of the AdaBoost algorithm is denoted \( h_t \) and the class label predicted by this weak classifier for the training data element \( x \) is denoted \( h_t(x) \).

By comparing \( h_t(x) \) with \( y \) for \( i = 1, 2, \ldots, m \), the error rate of the classifier \( h_t \) can be assessed. The classification error rate of the weak classifier \( h_t \) is denoted \( \epsilon_t \). The weak classifier \( h_t \) is associated with \( \alpha_t \), which denotes how much trust can be attained by this classifier. Obviously, the larger the value of \( \alpha_t \) of a classifier model, the lower the trust level. The final classifier model is denoted as \( H \). This classifier carries out a weighted aggregation of the classifications produced by the individual weak classifiers to predict the class label for a new data sample. The weak classifier can be a perceptron or a simple threshold.

The AdaBoost algorithm can utilize up to \( T \) weak classifiers, which can be as simple as individual attributes or, individual features that provide some discrimination between the objects of interest.

In the following steps the Adaboost algorithm is described for \( t = 1, 2, \ldots, T \) classifiers:

1. For the probability distribution \( D_t(i) \), use a weak classifier for the training data.
2. Apply the weak classifier \( h_t \) as selected in the previous step to all training data.

\[
h_t : x \rightarrow \{-1, 1\}
\]  

(15)

3. Estimate the classification error rate \( \text{Prob} \{ h_t(x) \neq y \} \) for the \( h_t \) classifier by

\[
\epsilon_t = \frac{1}{2} \sum_{i=1}^{m} D_t(x_i)[h_t(x_i) - y_i]
\]

(16)

4. Calculate the trust factor for \( h_t \) by

\[
\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)
\]

(17)

5. Update the probability distribution over the training data for the next iteration:

\[
D_{t+1}(x) = \frac{D_t(x)e^{-\alpha_t h_t(x)}}{Z_t}
\]

(18)

where the role of \( Z_t \) is to serve as a normalizer. This set a value for \( Z_t \) so that

\[
\sum_{i=1}^{m} D_{t+1}(x_i) = 1
\]

(19)

6. Repeat for \( T \) classifiers.

7. At the end of \( T \) iterations, construct the final classifier \( H \) as follows:

\[
H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))
\]

(20)

where \( x \) is the new data element whose class label need to be predicted on the strength of the information in the training data. If for a new data sample \( x \), \( H(x) \) turns out to be positive, the predicted class label for \( x \) is \( 1 \). Otherwise, it is \( -1 \). Figure 6 shows the aggregation of the weak classifiers used by the AdaBoost algorithms to form a stronger classifier model, which classifies non-linearly separable data.

**MCS**

An approach in classification, which has gained much acceptance in the community of data mining and data fusion is the concept of ensembles or committees of classifiers, which involves combining multiple models of classifiers to form a composite, more stronger one. The idea behind this is very simple, in which the training dataset is used to train several different models, each of which is used to assign a class label to a previously unseen instance. These class labels are then combined suitably to generate a single class label for the instance. This has been found to improve the accuracy of the resulting model,29 however the process is computationally expensive and it is hard to understand how the decision was obtained as this is depends on the fusion technique used which can be one of the following techniques: Majority-voting, maximum, minimum, average, sum, decision templates and Dempster-Shafer theory of evidence.30 Ensembles have been used extensively in the context of decision tree classification algorithms.31

There are several ways to build classifier models from the same training dataset.32 However, the approaches vary in how they introduce randomization into the process of model building so that different models are generated. One approach used in creating ensembles is to change the instances, which form the training set for each classifier in the ensemble.

The most popular methods for this include:26

- **Bagging**: In this approach, a new sample of the training set is obtained through bootstrapping with each instance weighted equally.
• Boosting: In this case, a new sample of the training set is obtained using a distribution based on previous results.
• Pasting: In this approach, the ensemble of classifier models is grown using a subsample of the entire training set.

CONCLUSIONS

The goal of image segmentation is the isolation of the regions of interest in the image either by partitioning the image into connected semantic regions or by extracting one or many specific objects from the image. The success of the segmentation process is the key factor in the success of the overall detection process of WBCs from the microscopic images of the peripheral blood smear. A number of approaches to segmentation and image analysis have been developed and were reviewed in this paper. Future work on image analysis of peripheral blood smears should emphasize building a software tool utilizing the best performance image analysis algorithms along with data mining techniques,[9] which has a strong influence in automating the process of the peripheral blood smear image analysis.

Table 3 shows a list of the commercially available software packages that can help in the development of the analysis techniques mentioned above. MATLAB®[31] is a high level interactive interpreted language. It has a variety of machine learning and statistical pattern recognition libraries that help the developer to design their own algorithms. Octave[34] is another scientific computing environment; it supports numerical solution for linear and non-linear problems. Moreover it contains a number of image analysis and machine learning tools such as SVM and ANN libraries. R[35] is a free software package that is designed to help the developer to solve mathematical problems. It supports a wide range of machine learning algorithms. ImageJ[36] is an open source Java implementation if some image enhancement and measurement tools. However, it does not support machine learning algorithms.

Table 4 shows a comparison of some of the commercially available blood image analysis systems. HemaCAM[37] is a computer-based red cell analysis system. It can automate the assessment of blood cell counts. Another available system is the CellaVision™[38] automated cell count system. There are two version of the CellaVision system: DM96 and DM1200. They differ mainly in blood film processing capacity.

Table 3: Available image analysis and machine learning software packages

| Software package | Features | Licensed |
|------------------|----------|----------|
| MATLAB®          | Statistical libraries, Image analysis and visualization, Machine learning algorithms | Yes |
| Octave           | Statistical libraries, Image analysis and visualization, Machine learning algorithms | GNU license |
| R                | Statistical libraries, Image analysis and visualization, Machine learning algorithms | Free |
| Image J          | Image enhancement, measurement and visualization | Open source |

GNU: GNU free public license

Table 4: Available blood smears image analysis systems

| System              | Features                                      |
|---------------------|-----------------------------------------------|
| HemaCAM             | Blood smear counting, Live view of cells, Measurement of cells, Easy integration into existing LIS |
| CellaVision™ DM96   | Analysis blood smear and body fluids, Digital slides functionality, Automated workflow |
| CellaVision™ DM1200 | Analysis blood smear and body fluids, Digital slides functionality, Automated workflow, Loading capacity of 12 slides |

LIS: Laboratory information system

REFERENCES

1. de Vet HC, Koudstaal J, Kwee W, Willembrand D, Arends JW. Efforts to improve interobserver agreement in histopathological grading. J Clin Epidemiol 1995;48:869-73.
2. Mohammed EA, Mohamed MM, Naugler C, Far BH. Chronic lymphocytic leukemia cell segmentation from microscopic blood images using watershed algorithm and optimal thresholding. Electrical and Computer Engineering (CCECE), 2013. 26th Annual IEEE Canadian Conference on, IEEE; 2013.
3. Guo N, Zeng L, Wu Q. A method based on multispectral imaging technique for white blood cell segmentation. Comput Biol Med 2007;37:70-6.
4. Jiang K, Xiao Q, Xiong Y. A novel white blood cell segmentation scheme based on feature space clustering. Soft Comput 2006;10:12-9.
5. Pan C, Park DS, Yoon S, Yang JC. Leukocyte image segmentation using simulated visual attention. Expert Syst Appl 2012;39:7479-94.
6. Levenson RM, Hoyt CC. Spectral imaging and microscopy. American Laboratory; 2000. Available from: http://www.spectralcameras.com/files/American_Laboratory_2000.pdf. [Last updated on 2000 Nov 14; Last cited on 2014 Jan 20].
7. Meyer F. Topographic distance and watershed lines. Signal Processing 1994;38:113-25.
8. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational
Learning Theory; ACM; 1992.

9. Fukunaga K. Introduction to Statistical Pattern Recognition. Waltham, Massachusetts: Academic press; Elsevier; 1990. Access online via Elsevier.

10. Ko BC, Gim JW, Nam YJ. Automatic white blood cell segmentation using stepwise merging rules and gradient vector flow snake. Micron 2011;42:695-705.

11. Li S, Fevens T, Kryžák A, Li S. Automatic clinical image segmentation using pathological modeling, PCA and SVM. Eng Appl Artif Intell 2006;19:403-10.

12. Liu X, Feng G. Kernel biclustering k-means clustering for SVM training sample reduction. Pattern Recognition, 2008. ICPR 2008, 19th International Conference on; IEEE; 2008.

13. Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. Neural Networks for Signal Processing [VII]. Proceedings of the 1997, IEEE Workshop; IEEE; 1997.

14. Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical report msr-tr-98-14, Microsoft Research; 1998.

15. Joachims T. Making large-scale SVM learning practical. In: Schölkopf B, Burges C, Smola A, editors. Advances in Kernel Methods-Support Vector Learning. Cambridge, Massachusetts: MIT Press; 1999.

16. Madhloom HT, Kareem SA, Ariffin H. An image processing application for the localization and segmentation of lymphoblast cell using peripheral blood images. J Med Syst 2012;36:2149-58.

17. Wang XY, Zhang XJ, Yang HY, Bu J. A pixel-based color image segmentation using support vector machine and fuzzy $C$-means. Neural Netw 2012;33:148-59.

18. Jaffar MA, Ishiaq M, Ahmed B. Fuzzy wavelet-based color image segmentation using self-organizing neural network. International Journal of Innovative 2010;6:4813-4824.

19. Zhuang H, Low K, You W. Multichannel pulse-coupled-neural-network-based color image segmentation for object detection. Industr Electron IEEE Trans 2012;59:329-308.

20. Mohammed EA, Mohamed MM, Naugler C, Far BH. Application of support vector machine and k-means clustering algorithms for robust chronic lymphocytic leukemia color cell segmentation. Proceedings of the 15th IEEE International Conference on e-Health Networking, Application and Services HEALTHCOM 2013; IEEE; 2013.

21. Bowie JE, Young JT. An analysis technique for biological shape-III. Acta Cytol 1977;21:739-46.

22. Kohavi R, John GH. Wrappers for feature subset selection. Artif Intell 1997;97:273-324.

23. Ververidis D, Kotropoulos C. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. Signal Processing 2008;88:2956-70.

24. Yegnanarayana B. Artificial Neural Networks. Patparganj Industrial Estate, Delhi, India: PHI Learning Pvt. Ltd.; 2004.

25. Kecman V. Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models. Cambridge, Massachusetts: MIT Press; 2001.

26. Olshen L, Breiman JH, Friedman RA, Stone CJ. Classification and Regression Trees. Stamford, Connecticut: Wadsworth International Group; 1984.

27. Mitchell TM. Machine learning and data mining. Commun ACM 1999;42:30-6.

28. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory. New York: Springer; 1995.

29. Kuncheva Li. Combining pattern classifiers: Methods and algorithms. Hoboken, New Jersey: John Wiley and Sons; 2004.

30. Beynon M, Cosker D, Marshall D. An expert system for multi-criteria decision making using Dempster Shafer theory. Expert Syst Appl 2001;20:357-67.

31. Polikar R. Ensemble based systems in decision making. Circuits Syst Mag IEEE 2006;6:21-45.

32. Eom J, Kim S, Zhang B. AptaCDSS-E: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction. Expert Syst Appl 2008;34:2465-79.

33. MathWorks.com. MATLAB® software. Available from: http://www.mathworks.com/products/matlab/.[Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].

34. GNU Octave. Octave software. Available from: https://www.gnu.org/software/octave/. [Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].

35. The R project for statistical computing. R software. Available from: http://www.r-project.org/.[Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].

36. Image processing and analysis in java. Image J software. Available from: http://rsbweb.nih.gov/ij/.[Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].

37. Computer-assisted microscopy for hematology. HemaCAM. Available from: http://www.hemacam.com/.[Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].

38. CellaVision.com. CellaVision. Available from: http://www.cellavision.com/.[Last updated on 2014 Jan 20; Last cited on 2014 Jan 20].