AUTOMATION OF DIFFERENTIAL BLOOD COUNT

Neelam Sinha and A.G.Ramakrishnan

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India.
email addresses: {neelam, ramkiag}@ee.iisc.ernet.in

ABSTRACT

A technique for automating the differential count of blood is presented. The proposed system takes as input, color images of stained peripheral blood smears and identifies the class of each of the White Blood Cells (WBC), in order to determine the count of cells in each class. The process involves segmentation, feature extraction and classification. WBC segmentation is a two-step process carried out on the HSV-equivalent of the image, using K-Means clustering followed by EM-algorithm. Features extracted from the segmented cytoplasm and nucleus, are motivated by the visual cues of shape, color and texture. Various classifiers have been explored on different combinations of feature sets. The results presented here are based on trials conducted with normal cells. For training the classifiers, a library set of 50 patterns, with about 10 samples from each class, is used. The test data, disjoint form the training set, consists of 34 patterns, fairly represented by every class. The best classification accuracy of 97% is obtained using Neural networks, followed by 94% using SVM.

Keywords: Differential blood count, Cell segmentation, EM algorithm

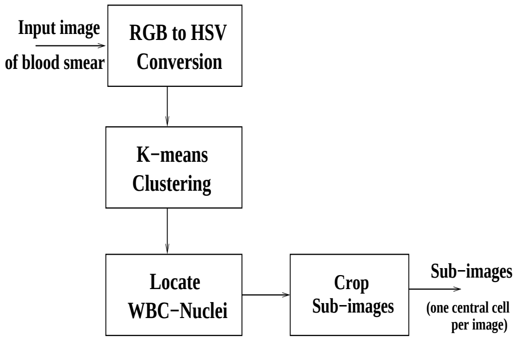
1. INTRODUCTION

Human peripheral blood consists of 5 types of White Blood Cells (WBC) called lymphocytes, monocytes, eosinophils, basophils and neutrophils [1]. Differential blood count (DBC) is carried out to calculate the relative percentage of each type of WBC, since it helps in diagnosing many ailments. High neutrophil count could suggest cancer, while high lymphocyte count suggests AIDS. High monocyte and eosinophil count usually point at bacterial infection. Thus DBC forms an important statistic of one's health status. A typical blood smear consists of WBC's, red blood cells, platelets, plasma and cell fragments. The automation of DBC involves segmentation, feature extraction and classification, followed by a counter to keep track of the number of cells counted in each class. Some of the techniques proposed to accomplish this task, are explained in the following section.

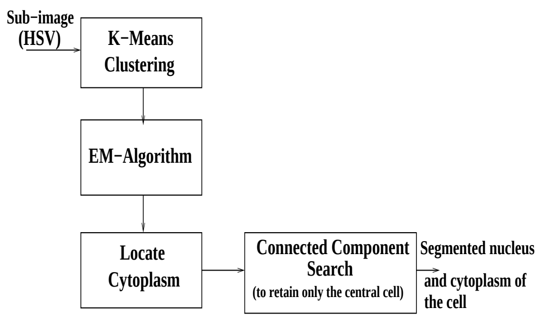
2. SURVEY OF EXISTING METHODS

Bikhet *et al.* [2] have reported segmentation and classification of the 5 types of WBC's in peripheral blood using gray images of blood smears. Hierarchical thresholding is used to localize the WBC's. The features extracted are the areas of nucleus and cytoplasm, average gray level, circularity measure and the ratio of nucleus to cell area. Classification accuracy of 90% is reported on 71 cells. The nature of the cells as being healthy or diseased isn't mentioned. Besides, the classifier used has not been disclosed. Ongun *et al.* [3] have worked on color smear images, containing both peripheral and immature cells. WBC's are segmented by morphological preprocessing combined with fuzzy patch labeling. Shape features are areas of cell and nucleus, ratio of nucleus to overall cell area, cell perimeter, compactness and boundary energy of the nucleus. Texture features include contrast, homogeneity and entropy derived from the gray-level co-occurrence matrix. Color histogram, mean and standard deviation of the color components in CIE-Lab domain, form the color features. This 57-dimensional feature set is used for classification using various classifiers, with a peak performance of 91% using SVM. The condition of the cells as being healthy or diseased is not mentioned. Park [4] has reported segmentation and classification carried out on low resolution gray images of immature cells. The feature set consists of shape, texture and statistical color features. Neural nets have resulted in an accuracy of 70%. The performance of the DBC-system depends most on the quality of segmentation, since the subsequent steps in the analysis depend on it. Color information could be utilized for more reliable segmentation. Besides, to make the system automatic, it is necessary to make the segmentation free of parameter-tuning. The system we propose performs color-based segmentation, free of the assumptions of circularity of cells and also the need for parameter-tuning. Segmentation is followed by extraction of simple features, motivated by visual cues, for classification.

The proposed system aims at distinguishing between the five classes of mature WBC's, namely lymphocyte, monocyte, eosinophil, basophil and neutrophil, in order to automate the process of DBC. The input to the system is a digital



(a)



(b)

Fig. 1. Overview of the segmentation scheme (a) Generation of sub-images containing single cells (b) Segmentation of nucleus and cytoplasm

image of blood smears of healthy subjects and the output is the count of cells in each of the classes. The main stages of the proposed system are: (i) Segmentation (ii) Feature Extraction and (iii) Classification.

3. SEGMENTATION SCHEME

The objective of segmentation is to extract the WBC's from the background cells as well as to distinguish between the cytoplasm and the nucleus. We first locate the nuclei of the cells using K-Means clustering following which a rectangular region that encompasses the entire cell is cropped. Subsequent processing is carried out on these sub-images each of which is assumed to contain only one WBC. K-Means clustering followed by EM-algorithm are used to get the final segmentation. Protrusion of neighboring cells is removed using connected component analysis. The image is converted to its HSV equivalent using,

$$H = cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\right]$$
$$S = 1 - \frac{3}{R+G+B}min(R,G,B)$$
$$V = \frac{1}{3}(R+G+B)$$

Each pixel in the image is represented by a vector of 3 components namely, H, S and V. The S-component, as it plays a more conspicuous part, is given more weightage as compared to the other two. K-Means clustering is performed on this collection of vectors. In our trials, we have used 5 clusters. Centroids are initialized by finding the mean vector and looking for those K-vectors that are farthest from the mean. Euclidean distance in the feature space is used as the measure of dissimilarity. The convergence criteria is that the difference in the centroids in successive iterations is less than a pre-defined threshold. At the end of this run, we get a class label for each of the pixels, and the centroids for each of the classes. A priori knowledge tells us that the centroid with maximum value of saturation, corresponds to the nucleus. We then crop a rectangular region, surrounding the nucleus, of sufficient area so as to enclose the entire cell. Thus sub-images, each containing only one WBC are obtained. Further processing involves two steps: (i) Initial estimation of parameters using K-Means (ii) Refinement of parameters using EM.

3.1. Initial Estimation using K-Means

Each sub-image is separately processed. K-Means clustering is repeated on the smaller dataset to obtain tighter clusters. At the end of this, we obtain a class-label for each pixel and the centroids for each class. Each cluster is modeled by a Gaussian distribution. The parameters are initialized using the clustering obtained by the K-Means algorithm. For the kth cluster, the mean is given by,

$$\boldsymbol{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{x}_i \tag{1}$$

where, $\mathbf{x}_i$ is every 3-D vector that belongs to the kth cluster, $\boldsymbol{\mu}_k$ is the mean vector and n_k is the number of vectors in the kth cluster.
Since the three features H, S and V are independent, the off-diagonal elements of the covariance matrix are taken as zero. Hence only the variance of each of the dimensions need to be computed. For the kth cluster, the dth diagonal element of the covariance matrix is given by:

$$C^k_{dd} = \frac{1}{n_k}\sum_{i=1}^{n_k}(x_{id} - \mu_{kd})^2 \tag{2}$$

where, n_k is the number of vectors in the kth cluster, x_{id} is the d-th dimension of the ith vector and μ_{kd} is the dth dimension of the mean vector of cluster k. These parameter values are refined in the subsequent step. The EM-algorithm is employed as follows.

4. PARAMETER REFINEMENT USING EM

The EM algorithm [5] consists of two steps: an Expectation step, followed by a Maximization step. The Expectation is with respect to the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observations. The Maximization step then provides a new estimate of the parameters. These two steps are iterated until convergence.

4.1. The E Step:

The E step computes the probability S_{ik} associated with the labeling the ith pixel, x_i as belonging to the kth cluster,

$$S_{ik} = \frac{1}{2\pi|\mathbf{C}^k|^{\frac{3}{2}}} e^{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{C}^k)^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)} \tag{3}$$

where C^k is the covariance matrix associated with cluster k, μ_k is the mean vector of cluster k, i and k take values $1, 2 \ldots N$ and $1, 2 \ldots K$, respectively. Here $N = width \times height$ and $K =$Number of clusters.

4.2. The M step:

The M-step refines the model parameters given the clustering arrived at E-step. The weighted mean of the kth cluster is updated as:

$$\hat{\mu}_k = \frac{\sum_1^{n_k} S_{ik} x_i}{\sum_1^{n_k} S_{ik}} \tag{4}$$

The weighted variance of the dth feature in the kth cluster is updated as:

$$\widehat{C}_{dd}^k = \frac{\sum_{i=1}^{n_k} S_{ik}(x_{id} - \hat{\mu}_{kd})^2}{\sum_{i=1}^{n_k} S_{ik}} \tag{5}$$

where x_{id} is the dth dimension of the ith vector and $\hat{\mu}_{kd}$ is the dth dimension of the mean vector of cluster k.
Both E and M-steps are carried out iteratively. The convergence criteria is taken as,

$$|\hat{\mu}_k^{(n+1)} - \hat{\mu}_k^{(n)}| < Threshold \tag{6}$$

Thresholding each of the distributions results in one region being captured in one distribution. A priori knowledge helps us associate nucleus with Gaussian distribution with the highest level of saturation, while cytoplasm is identified as the distribution with maximum number of pixels in immediate contact with the nucleus.

5. FEATURE EXTRACTION

Features for discriminating between different cell classes are devised based on domain knowledge of human experts. The features considered are based on (i) Shape (ii) Color (iii) Texture.

5.1. Shape features

Shape descriptors [6] are a set of numbers that describe a given shape. We use the binary masks of the nucleus and cytoplasm to compute these features. The features considered are eccentricity of the nucleus and cytoplasm contours, compactness of the nucleus, area-ratio and the number of nucleus lobes. Eccentricity is defined as the ratio between the major and minor axes, while compactness is defined as the ratio of area to square of the perimeter. Area ratio is taken as the ratio of the number of pixels that make up the cytoplasm to the ones that make up the nucleus. The number and structure of nucleus lobes is one of the prominent features used to identify the class of the WBC. A declustering technique involving the computation of the negative distance transform followed by watershed algorithm is used [7]. The advantage of using this method is that the contour information is not lost as would have been if one used morphological operations of open-close, for declustering.

5.2. Color features

The color features are extracted from the segmented nucleus and cytoplasm. The average value of each of the color components (R, G and B) of the nucleus and those of the cytoplasm are computed.
$Mean_C = \frac{1}{N}\sum_1^N C_i$ where N is the total number of pixels in the region of interest (either nucleus or cytoplasm) and C_i is the corresponding color (either R or G or B) component of the ith pixel.

5.3. Texture features

Texture features [8, 9] are computed only for the cytoplasm. The texture of the cytoplasm is visually distinct across the various classes, but it is not so for nucleus. The texture features used are based on computations of Gray-Level co-occurrence matrix (GLCM) and Autocorrelation matrix. The features based on GLCM are energy, entropy and correlation. The features based on autocorrelation matrix are coarseness and busyness. Gray-scale co-occurance matrix $\mathbf{P}_d$ is obtained by
$\mathbf{P}_d = |((r, s), (t, v) : I(r, s) = i, I(t, v) = j|$

The features computed are :

- Energy= $\sum_i \sum_j \mathbf{P}_d{}^2(i, j)$
- Entropy= $-\sum_i \sum_j \mathbf{P}_d(i, j) \log \mathbf{P}_d(i, j)$
- Correlation= $\sum_i \sum_j \frac{(i-\mu)(j-\mu)\mathbf{P}_d(i,j)}{\sigma_x \sigma_y}$
  where μ is the mean of $\mathbf{P}_d$ and σ_x and σ_y are the standard deviations of $\mathbf{P}_d(x)$ and $\mathbf{P}_d(y)$ respectively.

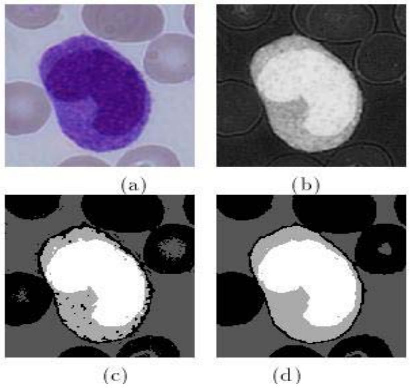


Fig. 2. Sequence of processing (a) Original image (b) Saturation image (c) K-Means output (d) Final output

Autocorrelation matrix $\rho(x,y)$ is computed as
$\rho(x,y) = \frac{\sum_i \sum_j I(i,j)I(i+x,j+y))}{\sum_i \sum_j I^2(i,j)}$
The features computed are :

- Coarseness $C_s = \frac{2}{\frac{\sum_i \sum_j Max(i,j)}{n} + \frac{\sum_i \sum_j Max(i,j)}{m}}$ where $Max(i,j) = 1$ if point (i,j) is either a row maxima or column maxima, else $Max(i,j) = 0$.
- Busyness $B_s = 1 - C_s^{\frac{1}{\alpha}}$, where $\frac{1}{\alpha}$ is a power to make C_s significant against 1.

6. CLASSIFICATION

According to the classical rule of thumb, the number of training patterns for each class must be 5 to 10 times the dimensionality of the feature vector. But due to unavailability of sufficient data, we have had to make do with a far smaller training set. The training data consists of 50 samples and test data consists of 34 samples, with fair representation from each class, but for the exception of the class basophil, of whose we have very few samples. The test instances are different from the ones used for training. Table 1 shows the performance of the classifiers used on various combinations of feature sets.

Table 1. *Comparison of Classifier performance on the feature sets (in %)*

Classifier	Texture	Shape-Color	Combined
NN	50.0	64.7	73.5
KNN	41.2	67.6	70.6
W-KNN	50.0	70.6	82.3
Bayes	44.1	79.4	82.3
SVM	50.0	91.1	**94.1**
NNet	**73.5**	**97.1**	**94.1**

7. RESULTS

The novelty of the approach for automation of DBC lies in the segmentation scheme used. The proposed segmentation scheme has been applied on 115 peripheral blood smear slides, stained using May-Grunwald-Giesma (MGG) stain. Each image consists of one or more cells of the same or different types, and is of size 1000×1300. Segmentation accuracy of 80% is obtained. This scheme is automatic since it needs no parameter-tuning. It would work on any dataset whose magnification is known. For testing the performance of the system nearly equal samples are chosen from all classes so as to give equal weightage to each class. The classification accuracy obtained using very simple features, is comparable with that obtained with the best of the existing techniques that use more sophisticated features.

8. REFERENCES

[1] Dacie and Lewis, *Practical Haematology*, Churchill Livingstone, Inc., 2001.

[2] Bikhet S.F., Darwish A.M., Tolba H.A., and Shaheen S.I., "Segentation and classification of white blood cells," in *IEEE Intl. Conf. On Acoustics, Speech and Signal Processing, ICASSP*, 1999, pp. 2259–2261.

[3] Guclu Ongun, Ugur Halici, Kemal Leblebicioglu, and Volkan Atalay, "Feature extraction and classification of blood cells for an automated differential blood count system," in *IEEE IJCNN*, 2001, pp. 2461–2466.

[4] J. Park and J. Keller, "Fuzzy patch label relaxation in bone marrow cell segmentation," in *A Computational Intelligence System for Cell Classification*, 1997, pp. 1133–1138.

[5] Yair Weiss, "Motion segmentation using EM- a short tutorial," Tech. Rep., MIT, MA 02139, USA, Nov. 1996, E10-120.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley Publishing Co., 1993.

[7] Luc Vincent and Pierre Soille, "Watersheds in digital space and efficient algorithm based on immersion simulations," *IEEE Transactions on PAMI*, vol. 13, no. 6, pp. 583598, 1991.

[8] N. Abbadeni, "Computational measures corresponding to perceptual textural features," in *Proceeding of the IEEE International Conference on Image Processing*, 2000, pp. 897–900.

[9] M. Tuceryan and A. K. Jain, *Texture Analysis, Handbook of Pattern Recognition and computer Vision (2nd Edition)*, World Scientific Publishing Co., 1998.