Segmentation and Classification of Leukocytes Using Neural Networks. A Generalization Direction

Pedro Rodrigues[1], Manuel Ferreira[2], and João Monteiro[3]

[1] Instituto Politécnico de Bragança; Campus de Santa Apolónia; Apartado 134; 5301-857 Bragança; Portugal; `pjsr@ipb.pt`
[2] Escola de Engenharia; Universidade do Minho; Campus de Azurém; 4800-058 Guimarães; Portugal; `mjf@dei.uminho.pt`
[3] Escola de Engenharia; Universidade do Minho; Campus de Azurém; 4800-058 Guimarães; Portugal; `Joao.Monteiro@dei.uminho.pt`

Summary. In image digital processing, as in other fields, it is commonly difficult to simultaneously achieve a generalizing system and a specialistic system. The segmentation and classification of leukocytes is an application where this fact is evident. First an exclusively supervised approach to segmentation and classification of blood white cells images is shown. As this method produces some drawbacks related to the specialistic/generalized problems, another process formed by two neural networks is proposed. One is an unsupervised network and the other one is a supervised neural network. The goal is to achieve a better generalizing system while still doing well the role of a specialistic system. We will compare the performance of the two approaches.

1 Introduction

The automatic identification and the automatic counting of human leukocytes, from blood, represent a factor of combativity and a factor of quality for the modern laboratories of clinical analyses. However, the leukocyte types that the current equipment is able to manage are restricted to few classes. Moreover, that equipment needs a set of expensive chemicals to reach its goals. Thus, nowadays, there are situations where the identification and differential counting operations have to be made by the human eye using a microscope. Those situations can be correlated with disease cases. Some abnormal cells may be observed only in specific diseases, while other may be seen occasionally in normal peripheral blood. The number of different white cells that can come out is relatively huge, more than twenty, and this increases the difficulty of obtaining a large functional process. Even restricting the set of white cells to a normal group, Fig. 1, the task, when seen by artificial vision, is not straightforward. The cell images might be in a very random aspect. The intensity,

the color, the natural artifacts, the relative statistical position among cells, the adjacency among cells, all these features can be very different from one sample to another sample. The cytoplasm geometry and the nucleus geometry are very diverse, even if the cells belong to the same class, Fig. 2. The texture of the nucleus and the texture of the cytoplasm are present in some images but in other images are blurred. A few issues of this problem are overcome by manipulating the physical and visual acquisition process and having care on the staining slides process. The majority of these issues are inherent in the blood itself. In fact, all these issues involve enormous tasks in order to be run in machine vision. They take, mainly, two directions. First, it is difficult to figure out all the mechanism that does the visual job as correctly as the humans do. And second, to obtain a little improvement it is necessary to widely increase the computational effort. We will show, mainly, a solution to try to minimize the first problem. The base for that is prosecuted through neural networks. First we will analyze an approach that uses, in neural terms, exclusively, feedforward neural networks trained by a backpropagation algorithm [1] (BPNN). We will see the positive and the negative aspects that this process discloses when applied on the white cells images. Then, we will show a hybrid model that tries to reduce some problems that were found in the previous solution. Support Vector Machine (SVM) [2], [3], [4] and Pulse-Coupled Neural Networks (PCNN) [5], [6] are integrated in such way that some negative aspects that arise with the first model are minimized.

This visual process is split into two different mechanisms. One mechanism must detect the pixels that are forming the region cell -segmentation. The second part is served by the first part. It has the task of extracting useful information from the segmented region. Finally a classifier is fed with the extracted information and it will decide in which cell class the region must be considered -classification. Both approaches have these two parts. The main difference lies in the form the data are delivered to the supervised neural networks. In the first case, the segmentation and the classification are exclusively supported by a supervised mechanism. The other method has an unsupervised part that cooperates with the supervised part. The junction of an unsupervised mechanism with a supervised mechanism is important to overcome the problems that are related with the strong variability of these images. The unsupervised mechanism will aggregate pixels that are forming features in multiple directions. The same high level feature can appear on the cells even if the cells have different colors. An exclusive supervisor method tends to learn this color instead of learning the real feature behind the color. Thus, if there appears a cell with a different color, when compared with the training cells colors, the exclusive supervised methods can fail. The addition of an unsupervised method improves the generality capacity of the system. This chapter begins with a presentation of a couple of techniques that exist to solve the problem. The presentation of our solutions is done in two parts. The first one is about a method that uses a supervised neural network to produce the segmentation and the classification of leukocytes. In the second part there is

shown a method that uses an unsupervised neural network and a supervised neural network to reach the same goal. In the following section the results of each approach are presented. At the end, a discussion and conclusions are expressed.
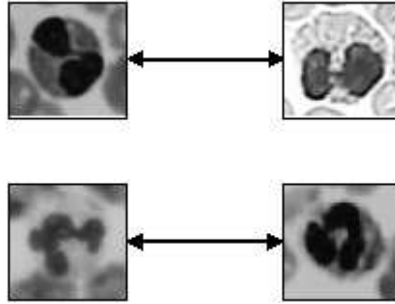

Fig. 1. Different types of leukocytes.

Fig. 2. Each row shows cells of the same type. The geometrical differences among cells of the same class can be higher.

2 Current Techniques

The companies that currently make equipment in this area generally solve the problem without image processing. Those methods are very accurate, but, with the exception of an extra of one or two classes, only the five common cells classes can be classified. They can be based in methods such as radiofrequency, direct current resistance, forward and side scatter of laser light, and fluorescence [7]. Normally these techniques require the addition of reagents. The usual form that makes possible the classification of any type of cells is still human vision and a microscope. It allows a diagnostic approach to blood disorders using blood film examination. For instance, an immature cell in the granulocytic series, occurring normally in bone marrow but not in the blood. During blood film examination, the individual types of white blood cells are enumerated. This is referred to as the differential count. Trying to put this ability into a machine has been very hard. Some researches [8] have shown

very good results using an unsupervised methods in the segmentation process. They assume some color saturation relations with the leukocytes and the rest of the elements. They assume also that the nuclei of the leukocytes have a similar hue on their cytoplasms. If for some fact these assumptions drop, the mechanisms can fail, as they have exemplified. A similar problem can be deduced from our first approach: the supervised neural networks assume, during the training process, arbitrary relations to store differential information among different regions. Thus, if the images to be processed are agreeing with training images, the process works at a high level of performance, but if the images to be processed and the training images mismatch, the process fails too. Our second approach tries to avoid creating these primary assumptions, but as we will see the goal is not reached in some directions. There are other researchers that are using image processing on fluorescence images by exploring other visual features [9]. The recent works that were studied still suffer in multiple ways from the issues of the variability present in natural cell images. To improve the generalization capacity, in [10] is presented a segmentation model that does not use the color information.

3 Approach - I (BPNN)

3.1 Backpropagation Neural Network

Feedforward Backpropagation is, probably, the artificial neural system most widely used in pattern recognition applications. In the majority of the supervision applications this neural network performs well and shows good computational performance. These were the initial presuppositions that led us to choose this network. The backpropagation algorithm used for training feedforward networks is based in the gradient descent. This method allows us to set the weights of the neurons automatically in a way to minimize the error between the target pattern set and the output pattern set. The neural network after the end of the training must map the input vector in an output vector without error. This neural network can consist in a several layers, generally, at least one hidden layer and one output layer. Each neuron has an activation function, in most cases a sigmoid function. The output value of each neuron can be represented by,

$$Y_{(l,n)} = f\left(\sum_{m=1}^{M_{l-1}} W_{(l,n,m)}\, X_{(l-1,m)} + b_{(l,n)}\right) \tag{1}$$

where n identifies the neuron within the l layer. W is the weight placed between the neuron n of the layer l and the output neuron m of the previous layer, or in the case where this layer is the input layer, m will indicate an input node of this neural net. X are the values of the neurons in the previous layer. b represents the biases of the neuron n at the layer l. M is the number of

neurons for each layer. $f(.)$ is the activation function that can be the sigmoid function,

$$f(v) = \frac{1}{1 + e^{-v}}.$$ (2)

Learning in a backpropagation network is in two steps. First each input pattern is presented to the network and propagated forward to the output. Second, the gradient descent is used to minimize the total error on the patterns in the training set. In gradient descent, weights are changed in proportion to the negative of an error derivative with respect to each weight,

$$\Delta W_{(l,\,n,\,m)} \propto -\frac{\partial E}{\partial W_{(l,\,n,\,m)}}$$ (3)

where E is,

$$E = \frac{1}{2} \sum_i \sum_n (T_{(i,n)} - Y_{(i,n)})^2$$ (4)

where Y is the activation of output unit n in response to pattern i and T is the target output value for unit n and pattern i.

3.2 The Segmentation Stage (BPNN)

A square scanning window is placed, which is centered over a pixel in the image, Fig. 3. In this way the window acquires the central pixel value and all the values of the neighbouring pixels. At the training step, this supplies the input pattern to the learning algorithm. To extract the rest of the patterns, the sampling window scans the entire image. To reduce the dimension of the neural net, only a part of the pixels that are inside the sampling window is extracted. The feedforward network used was trained by the backpropagation algorithm. The goal is to recognize the pattern where the central pixel is inside of one white cell region. The output patterns are indirectly formed manually. This enforces that a cell expert paints a set of images with a singular color that indicates the pixel is in the region cell. After the learning process ends, the system must be able to indicate if any pixel in an image belongs to a white cell region. Thus, all the pixels that make that statement true will be marked in the result image.

Architecture Used at Segmentation Stage (BPNN)

The modes and parameters that best performed for the segmentation were:

- Dynamic range to initialize the weights = [-0.02 , 0.02];
- Initializing mode: Nguyen-Widrow;
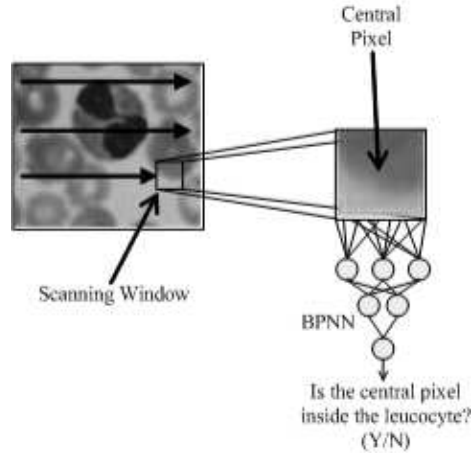- Activation function: Bipolar Sigmoid (for all neurons);

Fig. 3. In the segmentation part the goal is to recognize patterns where the central pixel is inside of one white cell region.

- Weights update mode: Delta-Bar-Delta (Batch);
- $\alpha = 0.002$ (Delta-Bar-Delta parameter);
- $\beta = 0.7$ (Delta-Bar-Delta parameter);
- $\kappa = 0.0001$ (Delta-Bar-Delta parameter);
- $\gamma = 0.4$ (Delta-Bar-Delta parameter);
- Hidden Layers: 1;
- Neurons in the input layer: 243;
- Neurons in the hidden layer: 4;
- Training patterns: 95354 (each pixel produces a pattern);
- Sampling window size: 25x25;
- Sampling pattern inside the sampling windows: on-off-off-on-off-off...
- Pixel format: HSL;

3.3 The Classification Step (BPNN)

An isolation process, based in the chain code, extracts each set of those pixels, previously segmented, that are forming one region cell. They are now numerically described by the Hu descriptors [11], geometrical primary descriptors and texture descriptors. These numbers are inputs to a new set of feedforward neural network. Each of them is training to recognize a white cell class, Fig. 4. Improvement of the generalization was tried by using a cross validation schema.

As the majority of the leukocytes may show two significant separable parts, the nucleus and the cytoplasm, it is important to split them to improve and simplify the classification process. A set of heuristics is used to analyze the histogram outline. Typically the cell histogram consists of two large modes,
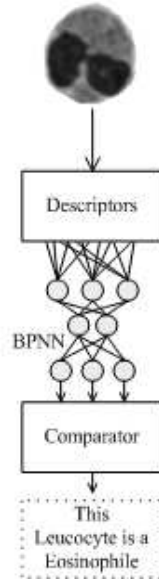
Fig. 4. The classification part. The classifier, after being trained, should be able to "announce" the class of white cell that is present at its inputs.

one will be the pixel occurrence of the cytoplasm and the other one will be the pixels occurrence of the nucleus. The threshold value must be between the peaks of these modes, Fig. 5. The heuristics conveys to a threshold value that separates the pixels of the nucleus from the pixels of the cytoplasm. The histogram outline is first smoothed and then it is scanned, with the gradient information being observed by the heuristics to try not to select a minimum that is not between the two modes. In most cases this method is enough to produce satisfactory results. But if the cytoplasm or the nucleus have a notorious granularity, some pixels in both regions can have the same values. Moreover, some times the cytoplasm and the nucleus have a low contrast between them. If these issues happen, the simple use of global threshold produces some mistakes. Thus, in future works this part needs to be improved.

The main heuristics:

- Is the negative gradient low enough to be the new threshold value considered? (if the negative gradient is high, the new threshold value can be in the wrong valley.)
- Has the relative long slope of one large mode already been overcome?
- After first mode descended, is the slope positive again and for many iterations? If yes, we are now at the second mode and the process must finish with the minimum found.
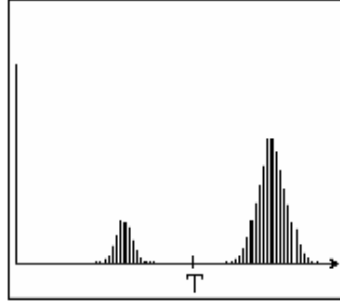
Fig. 5. Typical histogram of a leukocyte region.

After separating the nucleus from the cytoplasm, it is necessary to extract features from each one. The features are numerically coded by the Hu descriptors, geometrical primary descriptors and texture descriptors. They are invariant to translation, rotation and some of them are invariant to scale change. These features are extracted from the gray nucleus regions, from the gray cytoplasm regions, from the binarization of the nucleus and from the binarization of the cytoplasm. The set of features that were used and given to the neural network inputs are present in Table 1.

Shape Descriptors - Spatial Moments With Invariance to Translation and Rotation

- Row moment of inertia;
- Column moment of inertia;
- Aspect;
- Spread;
- Hu descriptor 1 normalized (nh1);
- Hu descriptor 2 normalized (nh2);
- Hu descriptor 3 normalized (nh3);
- Hu descriptor 4 normalized (nh4);
- Hu descriptor 5 normalized (nh5);
- Hu descriptor 6 normalized (nh6).

The Hu descriptors tend to be very small and to have an enormous dynamic range. This could be negative to the neural process. Thus, the Hu descriptors are normalized as shown in [12].

Texture Descriptors

To represent numerally the texture were used:

- Mean deviation;
- Standard deviation;

- Skewness;
- Kurtosis;
- FNC - First neighbour Contrast [12];
- SNC - Second neighbour Contrast [12].

Primary Descriptors

Six geometrical primary descriptors were used :

- The average intensity of the nucleus (N_uI);
- The average intensity of the cytoplasm (C_yI);
- The area of the nucleus (N_uA);
- The area of the cytoplasm (C_yA);
- The perimeter of the nucleus (N_uP);
- The perimeter of the cytoplasm (C_yP).
- Circularity.

These descriptors were related through the followings ratios:

$$IR = \frac{N_uI}{C_yI} \tag{5}$$

$$AR = \frac{C_yA}{N_uA} \tag{6}$$

$$PR = \frac{N_uP}{C_yP} \tag{7}$$

Architecture Used at Classification Stage (BPNN)

The modes and parameters that performed best for the classification were:

- Dynamic range to initialize the weights = [-0.05 , 0.05];
- Initialization mode: Nguyen-Widrow;
- Activation function: bipolar sigmoid (for all neurons);
- Weights update mode: Delta-Bar-Delta (Batch);
- $\alpha = 0.0035$ (Delta-Bar-Delta parameter);
- $\beta = 0.7$ (Delta-Bar-Delta parameter);
- $\kappa = 0.0009$ (Delta-Bar-Delta parameter);
- $\gamma = 0.5$ (Delta-Bar-Delta parameter);
- Hidden Layers: 1;
- Neurons in the input layer: 46;
- Neurons in the hidden layer: 12;
- Training patterns: 276;
- Patterns to cross validation: 48;
- Rejection thresholds: $K = 0.15$ and $Q = -0.3$.

Table 1. Relation of the used descriptors

	Nucleos (Binary)	Cytoplasm (Binary)	Nucleos (Gray)	Cytoplasm (Gray)	Whole cell (Gray)
Mq(2,0)	**X**	**X**			**X**
Mq(0,2)	**X**	**X**			**X**
Aspect	**X**	**X**			**X**
Spread	**X**	**X**			**X**
nh1	**X**	**X**			
nh2	**X**	**X**			
nh3	**X**	**X**			
nh4	**X**	**X**			
nh5	**X**	**X**			
nh6	**X**	**X**			
Mean deviation			**X**	**X**	
Standard deviation			**X**	**X**	
Skewness			**X**	**X**	
Kurtosis			**X**	**X**	
FNC			**X**	**X**	
SNC			**X**	**X**	
$N_u I$			**X**		
$C_y I$				**X**	
$N_u A$	**X**				
$C_y A$		**X**			
$N_u P$	**X**				
$C_y P$		**X**			
IR	**X**	**X**			
AR	**X**	**X**			
PR	**X**	**X**			
Circularity	**X**	**X**			

Comparator

To minimize mistakes in the classification, a rejection factor was added. The neural net output domain is $[-1; 1]$. There are two values to control the rejection. If one of the set of classification neural nets has its output in 1, this neural net is indicating the class for which it was trained. On other hand, if the neural net gives the value -1 it means this cell does not belong to that class. As the number of classification neural nets is equal to the number of cell classes to classify, it is important to observe the output value of each neural net. This is done in the comparator block. Thus, if none of the output values is higher than the Q threshold, the solution will be rejected with an 'Unknown'. If the two higher outputs have values where their difference is lower than the K threshold, then the rejection message will be 'Uncertain'. These values are figured out analyzing the ROC [1] curves to the neural net outputs.

---

[1] Receiver Operating Characteristic

The main conclusion after using this approach was that the performance on this method is very dependent, especially, on the training set. Obviously, if the segmentation fails the classification fails too. So, considering the segmentation step, we can find a set of problems. If the images, which we have submitted to the evaluation process after the training, are similar, in brightness, in color, in cell position statistics, then the process reaches a relatively good succeeded level. Roughly, 81% of the cells are classified correctly. But those are natural images. Thus, it is very difficult to guarantee the stability of the future images in relation to the training pattern set. If we try to increase the size of the training set, the problems with the training itself start increasing. The common issues related to "the curse of the dimensionality" of the neural network arise intensely. Staying stuck at local minimums and the exponential growth of the operations number make this solution impractical. Another problem about the use of the neural networks in this scenario and in this way was the great difficulty in enforcing the learning of the neural network over a particular feature. In other words, it is extremely difficult to select a training set to give essential features to the neural network. Normally to presume that the neural network will extract the essential features is an improbable presupposition.

4 Approach - II (PCNN-SVM)

The proposal of the second approach, the hybrid approach, is to try to overcome several problems of the first approach.

4.1 PCNN

One characteristic of the Pulsed-Coupled Neural Network is to be able to extract essential visual information from a given image. Regions composed by different patterns at different scales, as textures and sets of subregions made of many intensities, are identified at different interactions. The PCNN process is unsupervised. Therefor, we don't need to look out for the essential features of a region, neither does a neural network supervisor have to do that. This was the main reason to choose the PCNN as the unsupervision method. The pulsed coupled neural network is based on the Eckhorn model [5]. A PCNN neuron has the Feeding and the Linking compartment. Its synaptic weights, M and W, respectively control the communication with the neighbouring neurons. Only the Feeding compartment receives the input stimulus (pixel value)S. The neuron output is represented by Y. V_F and V_L adjust the global influences of the external neurons in each compartment respectively. The memory effect magnitude is controlled by the α_F , α_L and δ_n. The value of the Feeding compartment at the iteration n is determined by

$$F_{ij}\left[n\right] = e^{\alpha_F \delta_n} F_{ij}\left[n-1\right] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}\left[n-1\right] \tag{8}$$

The value of the Linking compartment at the iteration n is determined by

$$L_{ij}\left[n\right] = e^{\alpha_L \delta_n} L_{ij}\left[n-1\right] + V_L \sum_{kl} W_{ijkl} Y_{kl}\left[n-1\right] \tag{9}$$

The state of these compartments is combined by

$$U_{ij}\left[n\right] = F_{ij}\left[n\right]\left\{1 + \beta L_{ij}\left[n\right]\right\} \tag{10}$$

to obtain the internal state of the neuron and where β adjusts the interference between both compartments.

The internal state is compared with a dynamic value to produce the output Y,

$$Y_{ij}\left[n\right] = \begin{cases} 1 & se\ U_{ij}\left[n\right] > \Theta_{ij}\left[n-1\right] \\ 0 & otherwise \end{cases} \tag{11}$$

where Θ is the dynamic threshold described by,

$$\Theta_{ij}\left[n\right] = e^{\alpha_\Theta \delta_n} \Theta_{ij}\left[n-1\right] + V_\Theta Y_{ij}\left[n\right] \tag{12}$$

The memory effect of Θ is controlled by α_Θ and δ_n. The participation of the fired value Y in Θ is adjusted by V_Θ.

The state of each PCNN neuron in a neural network is dependent on the external stimuli that are represented by the input pixel values. It is also dependent on the output state of the neighbouring neurons and on its own internal state. i.e., a neuron that has fired recently reduces its sensibility to fire immediately again. These features enable the neuron fire patterns to show, at different times, important regions of the image. Those regions are features of the image in diverse domains, such as different textures, gray levels or aggregate subregions.

4.2 SVM

Given a training set of instance-label pairs (x_i, y_i), $i = 1, \ldots, l$ where $x_i \in \Re^n$ and $y \in \{1, -1\}^l$, the support vector machine requires the solution of the following optimization problem:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{13}$$

$$subject\ to \quad y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0$$

A SVM network maps the training vectors x_i in a higher dimensional space by the function ϕ. Thus, the SVM just needs to find the linear separated hyperplane with the maximal margin in this higher dimensional space. C is the penalty parameter of the error term ξ, $C > 0$.

$K\left(x_i, x_j\right) \equiv \phi(x_i)^T \phi\left(x_j\right)$ is the kernel function. There are innumerous kernel function types; some can fit better a specific application than others. We used the Adatron kernel [13], [14].

4.3 Zernike Descriptors

The Zernike moments perform better than the Cartesian moments in terms of noise resilience and information redundancy and furthermore they allow reconstruction capability (Note that this subsection is derived largely from [15]). The Zernike orthogonal moments [16] are constructed using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disc $x^2 + y^2 \leq 1$. They are represented as:

$$A_{mn} = \frac{m+1}{\pi} \iint_{x^2+y^2 \leq 1} f(x,y) \left[V_{mn}(x,y)\right]^* dx\, dy \qquad (14)$$

where $m = 0, 1, 2, \ldots, \infty$, $f(x,y)$ is the function being described, * denotes the complex conjugate and n is an integer, depicting the angular dependence and subject to the conditions:

$$m - |n| = even, \quad |n| \leq m \qquad (15)$$

The Zernike polynomials $V_{mn}(x,y)$ expressed in polar coordinates are:

$$V_{mn}(r,\theta) = R_{mn}(r)\exp(jn\theta) \qquad (16)$$

where (r,θ) are defined over the unit disc and $R_{mn}(r)$ is the orthogonal radial polynomial, defined as:

$$R_{mn}(r) = \sum_{s=0}^{\frac{m-|n|}{2}} (-1)^s F(m,n,s,r) \qquad (17)$$

where:

$$F(m,n,s,r) = \frac{(m-s)!}{s!\left(\frac{m+|n|}{2}-s\right)!\left(\frac{m-|n|}{2}-s\right)!} r^{m-2s} \qquad (18)$$

Translation and scale invariance can be achieved by normalising the image using the Cartesian moments prior to calculation of the Zernike moments [17] and the absolute value of a Zernike moment is rotation invariant.

4.4 Segmentation Stage (PCNN-SVM)

The whole process is quite similar to the previous approach. A sampling window scans the image pixel by pixel, but now the pattern that is extracted from it is supplied to the inputs of the PCNN. After some iterations, which

could reach the 20th or more, the PCNN has segmented different subregions of the image. Each iteration represents a new segmentation, Fig. 6. They are images with the size of the sampling window. All the iterations form a segmentation set and this could be seen as a signature [18]. The whole set is a huge dimension. To reduce it we have tried to use Zernike descriptors [19], [17] since normally they have a better behavior over the Hu descriptors. As the SVMs do not have the same problems as the Feedforward/Backpropagation neural networks (may become stuck in a local minimum) and show a high generalization ability in many real situations, the descriptors results are now the inputs for a SVM.

The SVM is trained as in the segmentation first approach. The difference is that now the PCNN and the Zernike are preceding transforms. We can see this relation in the Fig. 7.

The steps of the algorithm for the segmentation training part are explained next:

1. Scan the training image to be segmented, pixel by pixel, in a defined sequence. It can be from the top to the bottom and from the right to the left.
2. For each pixel in the image, consider a set of neighbouring pixels. This set can be extracted from a sampling window centrally placed over the pixel in decision. The size of this window, for this problem, was 51x51. In other problems the size must be adapted.
3. Each captured subregion in the previous steps is now processed by the Pulsed Coupled Neural Network. The PCNN will give a signature about each subregion. In this case, the PCNN will produce signatures of these subregions in 20 iterations.
4. The PCNN outputs construct, in each iteration, a new binary subregion. These binary subregions are numerically described by Zernike descriptors.
5. Until this step each image pixel to be segmented will have a numerical sequence with a length that is given by the number of PCNN iterations times the number of descriptors used. In the leukocytes problems six Zernike descriptors were used. Each one of those numerical sequences will be a training pattern.
6. The numerical sequences are normalized, one being the standard deviation and the mean value being zero.
7. The training patterns are presented at the inputs of the Support Vector Machine. The output patterns are formed by a unique variable. The value for this variable could be one of the two possible values. One value hints that the central pixel belongs to the white cell. On the other hand the complementary value refers that the central pixel is outside of a white cell region.
8. The last step will be repeated until all the training patterns are run through and until the SVM reaches convergence.

The steps of the algorithm for the segmentation pos-training part are explained next:

1. Scan the image to be segmented, pixel by pixel, in a defined sequence. It can be from the top to the bottom and from the right to the left.
2. For each pixel in the image, consider a set of neighbouring pixels. This set can be extracted from a sampling window centrally placed over the pixel in decision. The morphology and the size of the sampling windows have to be the same that were chosen for the training process.
3. Each captured subregion in the previous steps is now processed by the Pulsed Coupled Neural Network. The PCNN will give a signature about each subregion. The PCNN configuration has to be the same as the training part.
4. The PCNN outputs construct, in each iteration, a new binary subregion. These binary subregions are numerically described by Zernike descriptors. They are the Zernine descriptors that were used in the training stage.
5. Until this step each pixel of image to be segmented will have associated a numerical sequence with a length that is given by the number of PCNN iterations times the number of descriptors used. In the leukocytes problem six Zernike descriptors were used. Each one of those numerical sequences will be a pattern to be analyzed by the supervised method.
6. The numerical sequences are normalized, the standard deviation being the value one and the mean value being zero.
7. The patterns are presented at the inputs of the Support Vector Machine. The output patterns are formed by a unique variable. The value for this variable can be one of the two possible values. One value hints that the central pixel belongs to the white cell. On the other hand the complementary value refers that central pixel is outside of a white cell region.
8. The whole process will be finished when all the pixels of the image to be segmented are classified.

The Table 2 contains the parameterizations that were used in the PCNN (segmentation) and the Table 3 contains the general parameterizations that were used in the segmentation stage.

4.5 The Classification Stage (PCNN-SVM)

At the classification problem the changes are identical to the changes made in the segmentation (PCNN-SVM) problem relatively to approach I. Before applying the descriptors over the segmented regions we apply the PCNN. The signature result is numerically coded by the Zernike descriptors. Several SVMs are now used as classifiers of the type of cell, one for each cell type, Fig. 8. The PCNN has the task of splitting the cell in subregions in agreement with the essential features. Thus, the specific descriptors that were used in approach I are no longer needed, nor is the process to split the nucleus from the cytoplasm.
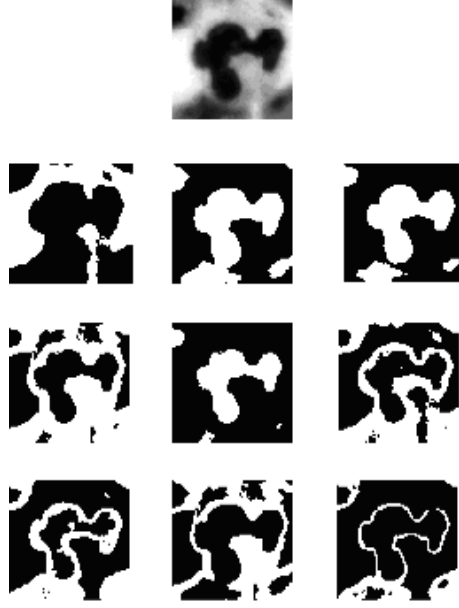
Fig. 6. The first image is an example of one image sampled by the sampling windows over a blood cells image. The rest of the images are the PCNN outputs along several iterations.
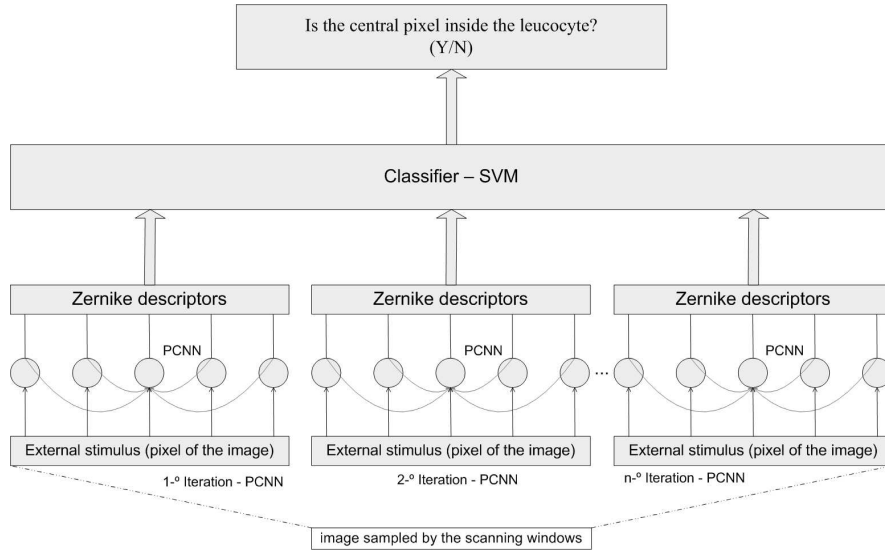

Fig. 7. The main components of the segmentation mechanism (PCNN-SVM).

Table 2. PCNN configuration for the segmentation stage.

PCNN Components	Settings
kernel M	5x5; Type $1/r$
kernel W	5x5; Type $1/r$
V_f	0.1
V_l	0.1
V_t	33
ß	0.1
A_f	10
A_l	6
A_t	18

Table 3. General configuration for the segmentation stage (PCNN-SVM).

Components	Settings
Pixel format	gray - 8 bits
Sampling window	square - 51x51
PCNN iterations	20
Descriptors	The first six Zernike descriptors: (0,0); (1,1); (2,0); (2,2); (3,1); (3,3).
Supervised Net	SVM
The number of SVM inputs	240 - 20 iterations x 6 complex values descriptors
SVM Kernel	Adatron
Number of SVMs	1
Training patterns	95354

The algorithm steps for the classification in training part are explained next:

1. Scan the pixels of the total image, that were segmented in the segmentation task, in a defined sequence. It can be from the top to the bottom and from the right to the left.
2. When the first pixel of a leukocyte that was previously marked by the segmentation process is found, a chain code method is used to mark the cell contour pixels.
3. Mark again all the pixels that are inside of the cell contour. This allows us to mark pixels that are inside the cell but were not detected by the segmentation process.
4. All the cell pixels are extracted from the marked region. Each region is processed by each PCNN in several iterations. At this stage, the number of iteration was set to 40. This number allows us to obtain a good information set of cell partition in order for it to be possible to produce the classification of the cell.
5. The PCNN outputs set is, in each iteration, a new binary subregion. These binary subregions are now numerically described by the Zernike descriptors.

6. For each cell the length of the numerical sequence that will describe the region is given by the number of PCNN iterations times the number of descriptors used. In this problem Zernike descriptors were used until their sixth order. Each one of those numerical sequences will be a training pattern to be learned by the supervised method.

7. The numerical sequences are normalized, the standard deviation being the value one and the mean value being zero.

8. Each training pattern is given to the SVM inputs. Each input pattern has a paired output pattern. The number of SVM is equal to the number of leukocytes to classify. Each SVM is trained in a way to signalize in its output when the input pattern belongs to the cell class that was assigned to it.

9. The last step will be repeated until all the training patterns are run through and until the SVM reaches the convergence.

10. To optimize the decision mechanism of the comparator block, the ROC curves are analyzed to figure out the threshold values to the SVM outputs, as was done in approach I.

The algorithm steps for the classification in the post-training part are explained next:

1. Scan the pixels of the total image, which were segmented in the segmentation task, in a defined sequence.

2. When the first pixel of a leukocyte that was previously marked by the segmentation process is found, a chain code method is used to mark the cell contour pixels.

3. Mark again all the pixels that are inside of the cell contour. This allows us to mark pixels that are inside the cell but were not detected by the segmentation process.

4. All the cell pixels are extracted from the marked region. Each region is processed by a PCNN in several iterations.

5. The PCNN outputs set are, in each iteration, a new binary subregion. These binary subregions are now numerically described by the Zernike descriptors.

6. For each cell the length of the numerical sequence that will describe the region is given by the number of PCNN iterations times the number of descriptors used. Each one of those numerical sequences will be a pattern to be analyzed by the supervised method.

7. The numerical sequences are normalized, the standard deviation being the value one and the mean value being zero.

8. Each pattern is given to the SVM inputs. This SVM has to be already trained. Each SVM will signalize in its output if the input pattern belongs to the cell class that was assigned to it.

9. Now, in the comparator block, if no SVM output is higher than the Q threshold, the solution will be rejected with an 'Unknown'. If the two

higher outputs have values where their difference is lower than the K threshold, then the rejection message will be 'Uncertain'.

The Table 4 contains the parameterizations that were used in the PCNNs (classification) and the Table 5 contains the general parameterizations that were used in the classification stage.
The comparator block works like the first approach comparator.

Table 4. PCNN configuration for the classification stage.

PCNN components	Settings
kernel M	7x7; type $1/r$
kernel W	7x7; type $1/r$
V_f	0.5
V_l	0.8
V_t	20
ß	0.2
A_f	10
A_l	1
A_t	5

Table 5. General configuration for the classification process (PCNN-SVM).

Components	Settings
Pixel format	gray - 8 bits
Input pixels	All the pixels belonging to a segmented cell
PCNN iterations	40
Descriptors	The first six Zernike descriptors: (0,0); (1,1); (2,0); (2,2); (3,1); (3,3).
Supervised Net	SVM
The number of SVM inputs	480 - 40 iterations x 6 complex values descriptors
SVM Kernel	Adatron
Number of SVMs	5 (the number of classes of cells that were considered)
Training patterns	276
Rejection thresholds:	$K = 0.10$ and $Q = $ -0.4.

5 Results

The main purpose of the second approach when compared to the first approach is to increase the generalization capacity, which is the weak point in the first approach. This is evident in the segmentation process. We tried to
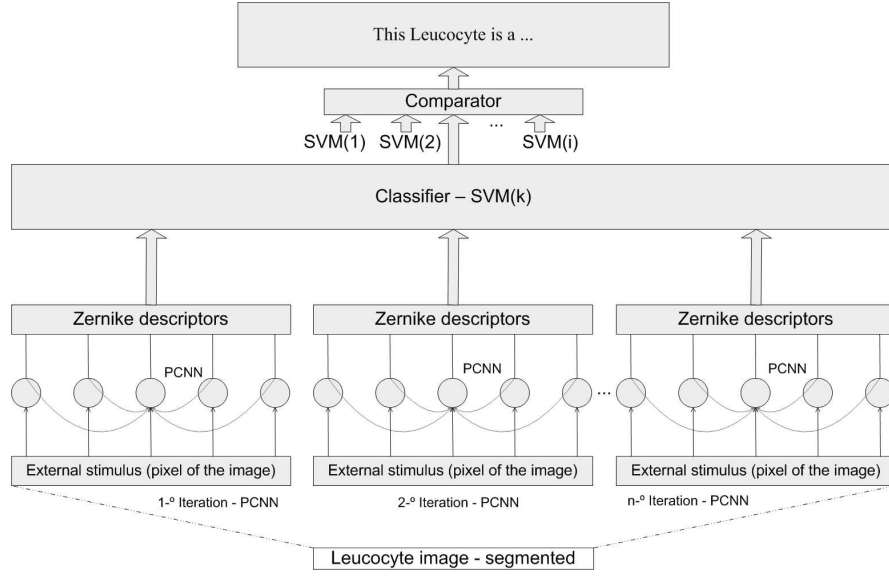
Fig. 8. The main components of the classification mechanism.

use gray pixels with the segmentation, in the first approach, but the results were very weak, even during the training step. Then, we decided to make the comparison using colored pixels in the first solution. The second solution demands a very high computational effort. In fact, one image with 129x110 pixels was processed in 1 second at the first approach and spent 217 seconds in the second approach. This was obtained in a 3Ghz Pentium 4 PC. This is due mainly to the PCNN process, as a PCNN task is executed by each pixel that appears in the image to be segmented. Then to see the performance of this second method and to minimize the computation time, which is the major weak point of this method, we have performed the tests with gray pixels.

5.1 Segmentation

In Fig. 9, Fig. 10 and Fig. 11 the first column has the original images. The second column has images that are the result of the first approach. The last column has the images from the second approach. All of them were out of the training set and their aspects were very random. In the two first cases the cytoplasm is completely segmented by the PCNN-SVM methods but the BPNN method fails, segmenting just partially the cytoplasm. The number of test images was 227. 77% of the test images were correctly segmented using the first approach. 86% of the test images were correctly segmented using the second approach. It is difficult to compare these results with other studies because the images databases are not the same. In any case, the segmentation results that are announced in [8] have a 95% success rate.
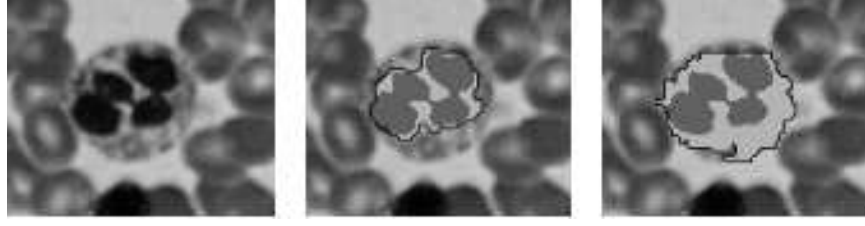
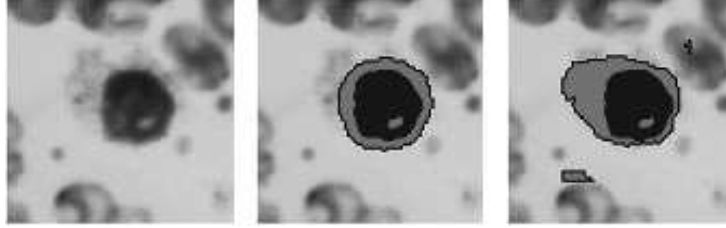**Fig. 9.** Comparison between the segmentation results of approach I and approach II for a Neutrophile.

Fig. 10. Comparison between the segmentation results of approach I and approach II for a Lymphocyte.
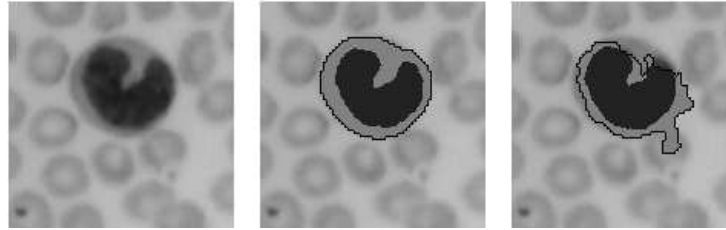

Fig. 11. Comparison between the segmentation results of approach I and approach II for a Monocyte.

5.2 Classification

To produce the classification results we used the white cell regions classified by a haematologist. Those results are presented in a confusion matrix. Table 6 is concerned with the first approach and Table 7 is concerned with the second approach. In the first column we have the cell classes announced by the haematologist and in the first row there are the classes announced by the system. Only the five normal white cells classes were considered. The Unc.(Uncertain) label means that the two higher matching classes were considered similar to the analyzed region. The Unk. (Unknown) means that the analyzed region does not seem to correspond to any class. Table 8 shows the fraction of correct answers for the BPNN method and for the PCNN-SVM method, as well as the number of cells classified. To obtain the classification results we selected

satisfactory segmented cells for both approaches. The rates were figured out from test data set. The approach I and the approach II rates were calculated using the same images. In [8] is announced 75% to 99% correct classification using 13 cell types.

Table 6. Confusion matrix for the classification process with the BPNN.

	Basophile	Eosinophile	Lymphocyte	Monocyte	Neutrophile	Unc.	Unk.
Basophile	4			1			
Eosinophile	1	38		4	4	2	2
Lymphocyte			55	6	1	3	
Monocyte	1	1	3	19	2	2	3
Neutrophile	2	10		4	61	8	4
Uncertain	1			3		12	2
Unknown		4	1	3	2	4	22

Table 7. Confusion matrix for the classification process with the PCNN-SVM.

	Basophile	Eosinophile	Lymphocyte	Monocyte	Neutrophile	Unc.	Unk.
Basophile	4			1			
Eosinophile	1	41		2	4	1	2
Lymphocyte			57	5		3	
Monocyte	1		1	22	2	2	3
Neutrophile	1	7		4	70	6	1
Uncertain	1			2		11	2
Unknown		2	1	3	2	3	20

Table 8. The fraction of correct answers.

	BPNN	PCNN-SVM	number of images
Basophile	80.0%	80.0%	5
Eosinophile	80.8%	85.4%	51
Lymphocyte	88.7%	91.9%	65
Monocyte	73.0%	84.6%	31
Neutrophile	79.2%	85.3%	89

6 Conclusions and Discussion

Approach II showed its greater ability to segment cell images with gray pixels over the first approach. Since color is a significant help to make the correct

detection (but it does not help the generalization), the first method has problems even during the training step if it is given gray pixel levels. This fact exposes the difficulty of the supervised neural net extracting the general features that are necessary to have success in the segmentation of images that are not in the training set. The color information only demonstrates itself to be a good feature if the future images do not change in relation to the training images and when the segmentation method is not supervised. The PCNN is able to correctly segment many subregions even in gray level pixels. The set of these subregion forms a better way to allow the generalistic capacity. Then, the SVM only needs to learn the generalistic features that are supplied to it by the PCNN. For these reasons, the second method has shown a better performance when images that have different aspects from the images of the training set were processed. We think that the results could be improved by giving to the PCNN color pixels too. On the other hand, it is difficult to control the previous segmentation made by the PCNN. The question is: is the subregion set figured out by the PCNN always the best set that defines the generalistic features? The answer is no. Although the subregion set can be extended to increase the PCNN iterations, hoping to achieve new generalistic features, there is not any certainty about it. We believe that a feedback mechanism applied to the PCNN bringing information from the supervised method could give clues to the PCNN to take the good direction. Another problem with the PCNN is the sequence order, where the same segmented subregion could appear in a different position in the sequence when the images to be segmented are different. This disturbs the signature that was learned by the SVM. A system to recognize each subregion extracted by the PCNN, in a isolated way, would be necessary. Fig. 11 shows a failed case using method II. Replacing the PCNN by other unsupervised mechanisms could be a good solution. Self organizing neural networks and multiresolution segmentation are just a few of a large set of possible choices. Certainly, it was possible to select a type of image where both methods always fail or to select types of images where both methods will perform always correctly. In a real situation we never know: images will appear that will fit the two cases (success or failure), and in this situation the second approach is in the better position to minimize the mistakes.

The average rate of correctly classified cells increases until 86.9%. The average rate of the first method was 81.31%. For these calculations the "Uncertain" and the "Unknown" cases were not considered. As this count is differential, it is possible to reject the doubtful cases and we shall do it. The second process is not a perfect method but it is better than the first one when related to the generalization problem. Thus we can say that the second method has a generalist potential over the first method.

When these results are compared with other studies they appear to be weak, but as we have said it is difficult to make the comparison because the images are not from the same set.

The first approach needs a very completed training set to avoid failures. And that is not easy to reach. The second approach is not so dependent on the training set but it obliges a very large computational effort. The first method has a much better time performance than the PCNN based method. In fact, the main responsibility for this lies with the PCNN and consequently the number of PCNN iterations. This kind of neural network is easily paralleled on distributed computers, on FPGAs and on other similar techniques [20],[21]. This makes practicable the implementation of this method for a useful system. We believe that the fusion of unsupervised with supervised methods must be pursued to try to overcome the problems of the variability in natural cell images.

References

1. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323** (1986) 533–536
2. Boser, B., Guyon, I., Vapnik, V.N.: A training algorithm for optimal margin classifiers. (1992)
3. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20** (1995) 273–297
4. Schölkopf, B., Burges, C.J.C., Smola, A.: Introduction to support vector learning. In: Advances in kernel methods. MIT press (1999) 1–15
5. Eckhorn, R., Reitboeck, H.J., Arndt, M., Dicke, P.: Feature linking via synchronization among distributed assemblies: Simulations of results from cat visual cortex. Neural Comp. (1990) 293–307
6. Johnson, J.L.: Pulse-coupled neural nets - translation, rotation, scale, distortion, and intensity signal invariance for images. Applied Optics **33** (1994) 6239–6253
7. de Jonge, R., Brouwer, R., van Rijn, M., van Acker, B.A.C., Otten, H.J.A.M., Lindemans, J.: Automated analysis of pleural fluid total and differential leukocyte counts with the sysmex xe-2100. Clin Chem Lab Med **44** (2006) 1367–1371
8. Ramoser, H., Laurain, V., Bischof, H., Ecker, R.: Leukocyte segmentation and classification in blood-smear images. In: Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the. (2005) 3371–3374
9. Hirono, T., Yabusaki, K., Yamada, Y.: The development of leukocyte counter using fluorescence imaging analysis. In: Biophotonics, 2004. APBP 2004. The Second Asian and Pacific Rim Symposium on. (2004) 213–214
10. Nilsson, B., Heyden, A.: Model-based segmentation of leukocytes clusters. In: Pattern Recognition, 2002. Proceedings. 16th International Conference on. Volume 1. (2002) 727–730vol.1
11. Hu, M.K.: Visual pattern recognition by moment invariants. Information Theory, IEEE Transactions on **8** (1962) 179–187
12. Masters, T.: Signal and Image Processing with Neural Networks. John Willey & Sons (1994)
13. Anlauf, J.K., Biehl, M.: The adatron - an adaptive perceptron algorithm. In: Europhysics Letters. (1989)

14. Cristianini, N., Campbell, C.: The kernel-adatron algorithm: a fast and simple learning procedure for support vector machines. (1998)
15. Shutler, J.D., Nixon, M.S.: Zernike velocity moments for description and recognition of moving shapes. (2002)
16. Teague, M.R.: Image analysis via the general theory of moments. Journal of the Optical Society of America (1917-1983) **70** (1980) 920–930
17. Khotanzad, A., Hong, Y.: Invariant image recognition by zernike moments. Pattern Analysis and Machine Intelligence, IEEE Transactions on **12** (1990) 489–497
18. Johnson, J.L.: The signature of images. (1994)
19. Zernike, F.: Beugungstheorie de schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode. In: Physica. (1934)
20. Kinser, J., Lindblad, T.: Implementation of pulse-coupled neural networks in a cnaps environment. IEEE Transactions on Neural Networks **10** (1999) 584–590
21. Lindblad, T., Kinser, J.M.: Image processing using pulse-coupled neural networks. Springer (2005)