applied optics

Expert knowledge for the recognition of leukemic cells

Rocio Ochoa-Montiel,[1,2] Gustavo Olague,[3,*] and Humberto Sossa[1,4]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación. Av. Juan de Dios Bátiz and M. Othón de Mendizabal, CDMX 07738, Mexico
[2] Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología. Calz. Apizaquito s/n, Apizaco, Tlaxcala, Mexico
[3] CICESE Research Center, EvoVision Laboratory, Ensenada 22860, Mexico
[4] Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias. Av. General Ramón Corona 2514, Zapopan, Jalisco, Mexico
*Corresponding author: olague@cicese.mx

Received 5 December 2019; revised 14 April 2020; accepted 15 April 2020; posted 15 April 2020 (Doc. ID 385208); published 8 May 2020

This work shows the advantage of expert knowledge for leukemic cell recognition. In the medical area, visual analysis of microscopic images has regularly used biological samples to recognize hematological disorders. Nowadays, techniques of image recognition are needed to achieve an adequate identification of blood tissues. This paper presents a procedure to acquire expert knowledge from blood cell images. We apply Gaussian mixtures, evolutionary computing, and standard techniques of image processing to extract knowledge. This information feeds a support vector machine or multilayer perceptron to classify healthy or leukemic cells. Additionally, convolutional neural networks are used as a benchmark to compare our proposed method with the state of the art. We use a public database of 260 healthy and leukemic cell images. Results show that our traditional pattern recognition methodology matches deep learning accuracy since the recognition of blood cells achieves 99.63%, whereas the convolutional neural networks reach 97.74% on average. Moreover, the computational effort of our approach is minimal, while meeting the requirement of being explainable.

https://doi.org/10.1364/AO.385208

1. INTRODUCTION

Image visual recognition is a fundamental task to solve problems in diverse fields. Branches of medicine such as pathology and hematology require the analysis of images to identify some abnormalities in tissues [1,2]. Nevertheless, lack of homogeneity in the images due to non-controlled conditions in the biological sample acquisition or the image capture, make the task of recognition difficult. Furthermore, inexperience or physical conditions such as fatigue or health conditions of people in charge of image acquisition and visual inspection cause differences in the interpretation of the images. Particularly, hematological diseases such as leukemia represent severe problems worldwide, and in undeveloped nations, the situation is worse, as cases in Mexico represent one of the leading causes of death in people under 15 years [3]. Some studies suggest that the incidence of a specific leukemia type named acute lymphoblastic (AL) is among the highest in the world [4]. Although advanced methods exist for the detection of this illness, these are very expensive and inaccessible for most of the affected people. Typically, the centers where these methods are available are located in big cities, which increases the cost and time of detection and treatment. For these reasons, it is common practice to carry out a visual analysis of blood smears to detect some abnormalities in blood cells. Sometimes, the first stage of treatment starts from the results in basic laboratory tests, with visual analysis being one of them.

Expert knowledge in computer science is generally achieved with rule-based systems that encapsulate synthetically what an expert knows (through extensive knowledge or ability based on research, experience, or occupation) in a particular field. The problem of blood cell recognition in leukemia has been addressed widely in the literature with diverse techniques of image processing and machine learning [5–14]. However, a review of the influence of expert knowledge in the task of recognition of leukemic cells has been seldom addressed. Considering expert knowledge as a method of information integration that allows the recognition of an object, in which specific information refers to particular knowledge about a distinct area or field, in Refs. [15–17], we found some advantages of using expert knowledge in the task of image recognition. In Ref. [15], expert knowledge is used as a pre-processing step to prepare input images for a classifier based on deep learning to study acute lymphocytic leukemia detection in single-cell blood smear images with excellent results. The authors of Ref. [16] describe data augmentation techniques to extract high-level features to distinguish between immature leukemic blasts and healthy cells by using a deep-learning-based method. They achieve more accurate recognition while incorporating data derived from expert knowledge. Also, [17] proves that fuzzy models built on expert knowledge, within a particular case, make the design more accurate while adapting the parameters and outputs to an application of data analysis.

It is worth mentioning that there are approaches that do not use expert knowledge to achieve object recognition [18,19]. Although these models achieve excellent results in many problems, several drawbacks are the lack of robustness, the high amount of overwhelming images, and that final results are in general unexplainable. Additionally, the models usually fail when the dataset used to test the model changes or the number of images in training is minimal concerning the number of classes and examples. For instance, [20] proposes to classify 40 classes of blood cells using typical network architectures, but the performance is low due to the extremely unbalanced number of images among different categories. Also, the limited number of images reported in Ref. [21] affects the accuracy in recognition of a type of blood cell called a lymphocyte, reaching no more than 74.7% of accuracy.

From previous revision, it is possible to note that the influence of expert knowledge for the task of recognition, and particularly for leukemic cells, is missing. In this work, we propose to determine the importance of the use of expert knowledge for such a task. In order to avoid the influence of different types of classifiers discriminating among classes [16], we selected two types of classifiers: those that naturally require expert knowledge and those that do not need it. Thus, representative classifiers used widely in literature are support vector machine (SVM) [22,23] and multilayer perceptron (MLP) [24,25] for the first types, and convolutional neural networks (CNNs) as the second type.

The main contribution of this paper is to show that expert knowledge is useful for the recognition of leukemia from blood cell nucleus images, which also leads to a better interpretation of the image, as opposed to approaches where this knowledge is unused. Moreover, the method for generating expert knowledge is entirely explainable since standard classical techniques of image processing such as the Gaussian mixture model (GMM) and evolutionary computing isolate the nucleus from the images in the dataset. A small sample of images with balanced classes is enough to make the comparison for recognition of leukemic cells by selected classifiers, using expert knowledge with derived features from the nucleus isolated in the previous stage. In the end, we show that the use of the proposed method reduces time consumption and computational resources in contrast to traditional techniques, which in their standard form, do not use expert knowledge.

The organization of this paper is as follows. Section 2 presents the materials and methods of the proposed work and the process to obtain expert knowledge of blood cells for their recognition. Section 3 details the experimental results. Finally, we give our conclusions in Section 4.

2. MATERIALS AND METHODS

An essential aspect of detecting hematological diseases is the morphology of cell components. In this regard, the nucleus of the cell brings relevant information for the blood cell recognition [26]. Notice the relevance of the nucleus for the recognition of a type of cell. Figure 1 shows images of some subtypes of healthy blood cells, in which the salient part is in the center of the image. It looks like a region of semi-circular shape more prominent than the remaining parts. Within this region, we can see other regions of intense color made of dark purple hues; this is the nucleus of the cell. As can be seen, the nucleus is a useful image region to discriminate between different types of cells. From the images, it is suggested that cells for each subtype shown have nuclei with similar shapes. Although there are other elements in the cell of interest, in this work, only the nucleus serves to recognize a leukemic cell from a healthy cell. Images of leukemic cells are shown in Fig. 2, where the same subtype of a cell is presented in its healthy and leukemic states. Within the most notable differences between a leukemic and healthy cell are the nucleus shape, its size, and the homogeneity.

In the first stage of our proposed expert system, we obtain the nucleus region from the input image. Later, some geometric properties (or descriptors) are extracted from this image region, following criteria proposed by medical experts. In a second stage, descriptors obtained through this expert knowledge are used to recognize the cell type as either a healthy or sick cell according to their qualities or characteristics.

Typical datasets used for the recognition of leukemic cells have drawbacks related to copyright issues since most images are not public [7,13], and the datasets are unbalanced [8,14] or in other cases were built from diverse sources and different acquisition conditions [12,18]; to overcome these disadvantages, in this work, we have selected a small dataset from Labati [27]. Although it has been reported that CNNs have a poor performance with small datasets, in the problem that we propose, the relationship among the number of features, number of classes, and dataset size does not influence significantly the performance of the CNNs used, as shown in Section 3. Additionally, there is evidence that in some tasks, CNNs achieve excellent results even with small datasets as in Ref. [28], where 260 images per class are used for discrimination between two classes in each experiment, whereas in Ref. [29], the problem of multi-instance multi-label

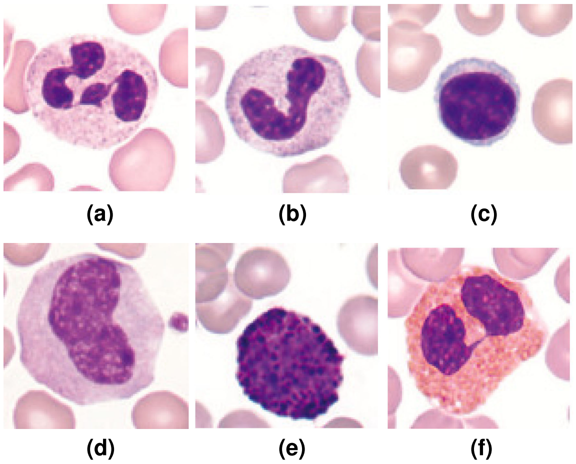


Fig. 1. Different subtypes of healthy blood cells [26]: (a) Neutrophil cell, (b) Band neutrophil cell, (c) Lymphocyte cell, (d) Monocyte cell, (e) Basophil cell, and (f) Eosinophil cell.

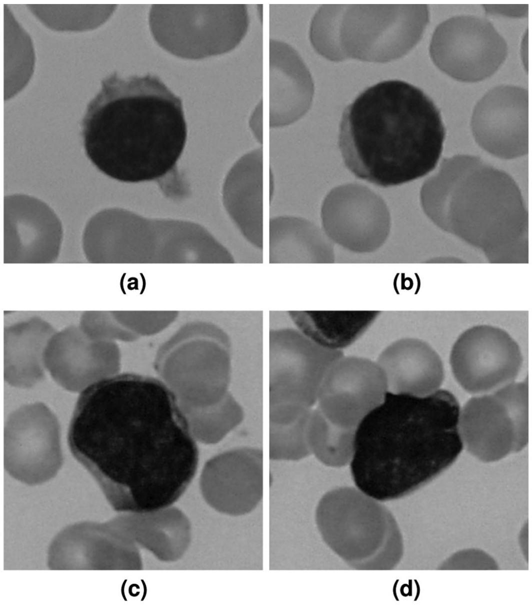


Fig. 2. Representative regions in images of healthy and leukemic cells: (a) Healthy cell, (b) Healthy cell, (c) Leukemic cell, and (d) Leukemic cell.

classification is addressed using 591 images for 23 labels, and 2000 images for five labels in separate experiments.

Labati acquired the images with an optical microscope coupled with a Canon PowerShot G5 camera [27]. All images are in .jpg format with 24-bit color depth, resolution 2592 × 1944. Images are divided into two datasets: ALL_IDB1 and ALL_IDB2, both including two types of cells: healthy cells and cells with AL leukemia (ALL) in which expert oncologists provide the classification. Images from ALL_IDB1 are taken with different magnifications of the microscope ranging from 300 to 500. Thus, each image contains one or more cells; in total, they are 108 (59 healthy cells and 49 leukemic cells) with a variable resolution between 1226 × 652 and 2592 × 1944.

In this work, we use the dataset ALL_IDB2; it is a collection of cropped area images of healthy and leukemic cells that belongs to ALL_IDB1, which includes 260 images of blood cells (130 for each cell type) in .tif format with resolution 257 × 257. Images were converted to grayscale using linear transformation in the range of [0–255] and resized to 600 × 600 pixels using bicubic interpolation. Each image contains only one cell of interest, as illustrated in Fig. 2.

A. Isolating the Nucleus of Blood Cells

Features such as the size and shape of cells, nucleus, and distribution of these cells in the sample are typically used for the detection and diagnosis of cancer from microscopic images [1]. Notice that in leukemia, the affected cells never were healthy, mainly because of the leukemic state given by the immaturity of the blood cell and its proliferation. For this reason, the cell appears in atypical shapes, as in Fig. 2, where images of a subtype of blood cell named lymphocyte appear with a nucleus bigger and less round in comparison with the healthy cells. Thus, experts regard the nucleus as an important region to discriminate a healthy cell from a leukemic cell. These morphological features of a nucleus, such as its roundness, area, and perimeter, to mention a few, are useful for the recognition of blood cells.

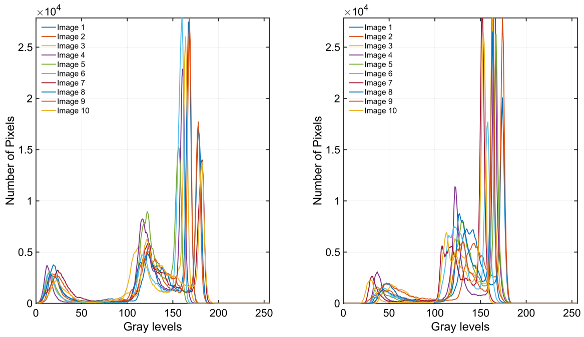


Fig. 3. Overlapping representative histograms of 10 images per class: healthy and leukemic cells.

In the images of blood cells, it is worth to note that these include at least three prominent regions: a dark region, which corresponds to the nucleus of a cell, a region with median intensity where most small cells arise (small circles), and finally a clearer region known as the background. In this regard, Fig. 3 shows the superposed histograms of gray scale images of healthy and leukemic cells. The histograms show three modes, which correspond to the most prominent regions in the images, the gray level values that occur most frequently in the histogram. According to the typical shape of histograms, we observe that for both types of cells, the mode referring to the nucleus of a cell (the mode on the left side) is separate from the remaining parts.

As mentioned before, the extraction of a nucleus region in the input image (darker region) through a normalized histogram is an estimation of the occurrence probability of intensity levels in the image, and thus it can be contained in a mixture of Gaussians. This is because an image is composed of gray intensity levels in the range $L = [0, L-1]$. These gray levels can be represented as a histogram $h(l_m) = s_m$, where l_m is the mth intensity value, and s_m is the number of pixels in the image with intensity l_m. Then, $h(l_m)$ can be normalized by dividing each of its components by the total number of pixels in the image denoted by T, so that the normalized histogram $h(l_m)$ is given by $v(l_m) = s_m/T$, for $m = 0, 1, 2, \ldots, L-1$, which is an estimate of the probability of occurrence of intensity level l_m in the image [30]. Thus, it is valid to consider the normalized histogram $v(l_m)$ described by a mixture of Gaussians through Eq. (1) as follows:

$$p(x) = \sum_{i=1}^{K} P_i \cdot p_i(x) = \sum_{i=1}^{K} \frac{P_i}{\sqrt{2\pi}\sigma_i} e^{\frac{(x-\mu_i)^2}{2\sigma_i^2}}. \quad (1)$$

A GMM is a parametric approach used to approximate probabilistic data models. Hence, such models can be used to compute the thresholds to separate regions of interest (ROIs) in an image [31]. In this sense, we use GMM because, unlike other nonparametric models such as Otsu [32], it considers the data of several components that can be modeled by a probability distribution, and they are susceptible to being analyzed separately. As

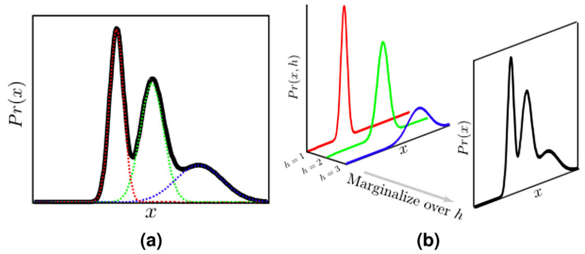


Fig. 4. Mixture of Gaussians: (a) Complex multimodal probability density function created by taking a *mixture* of several normal distributions, and (b) Mixture of Gaussians in terms of a jointed distribution $Pr(x, h)$ between the observed variable x and a discrete hidden variable h.

a consequence, a ROI can be obtained from the mixed model, whereas with non-parametric models, this does not occur.

Considering that in Eq. (1), P_i is *a priori* probability of class i, then $p_i(x)$ is assumed as the probability distribution function of intensity level random variable x in class i. In this way, the data in the mixture of Gaussians are described as a weighted sum of K normal distributions (number of classes within the image, i.e., the representative modes in Fig. 3), where μ_i and σ_i are the means and covariances of the normal distributions, respectively, and P_i are positive value weights that sum to one. Thus, according to Ref. [33], mixtures of Gaussians models describe complex multi-modal probability densities by combining simpler constituent distributions (see Fig. 4).

Hereunder, the value of K is adjusted to three because the histograms of images contain three modes that represent each class or region in these images; thus, $i = 1, \ldots 3$ in Eq. (1).

Later, the values for μ, σ, and P in Eq. (1) are computed using the squared minimum error between the sum of Gaussian mixtures $p(x)$ and the image histogram $v(l_m)$; this last is seen as a probability distribution function according to Eq. (2):

$$E = \frac{1}{L} \sum_{j=1}^{L} (p(x_j) - v(x_j))^2. \quad \textbf{(2)}$$

The problem of parameter estimation describes the error minimization of Eq. (2) as follows:

$$\text{Minimize } f(E) = \frac{1}{L} \sum_{j=1}^{L} (p(x_j) - v(x_j))^2, \quad \textbf{(3)}$$

with design variables

$$y = \{p_1, \sigma_1, \mu_1, p_2, \sigma_2, \mu_2, p_3, \sigma_3, \mu_3\} \quad \textbf{(4)}$$

subject to

$$z_1(E) = (p_1 + p_2 + p_3) - 1 = 0. \quad \textbf{(5)}$$

Note that for each mixture, we need to estimate three parameters: μ, σ, and p, which describe the mixture.

Since the estimation of parameters is a difficult problem, an analytical solution considering a set of simultaneous transcendental equations is not available due to the nonlinearity of equations. Additionally, the use of an iterative approach based on the gradient information is highly dependent on the initial values [30]. The differential evolution (DE) algorithm [34] has been selected in this work for the estimation of parameters because it has demonstrated excellent performance in problems of image processing and the estimation of parameters in diverse applications [35,36]. In this work, DE is an evolutionary algorithm in which the parameters of Eq. (4) define each individual of the population N_p. The constraint $z_1(E)$ in Eq. (5) is handled through feasibility rules proposed in Ref. [37], which considers the fitness of each individual during the selection process within the evolutionary algorithm.

Algorithm 1. Differential Evolution Algorithm

```
...
while $ConvergenceCriterion = unknown$ do
        ▷ $q_t$ defines a vector of the current vector population
            ▷ $nv_t$ defines a vector of the new vector population
    for $t = 0$; $t < N_p$; $t++$ do
        $r_1 \leftarrow rand(N_p)$
                    ▷ select a random index from $1, 2, \ldots, N_p$
        $r_2 \leftarrow rand(N_p)$
                    ▷ select a random index from $1, 2, \ldots, N_p$
        $r_3 \leftarrow rand(N_p)$
                    ▷ select a random index from $1, 2, \ldots, N_p$
        $u_t \leftarrow q_{r3} + F * (q_{r1} - q_{r2})$
        if $f(u_t) < f(q_t)$ then $nv_t \leftarrow u_t$;
        else $nv_t \leftarrow q_t$;
  ...
```

DE addresses the starting point problem by sampling the objective function at multiple, randomly chosen initial points. The preset parameter limits (design variables) define the domain from which the N_p vectors are chosen (initial population) in such a way that each vector is indexed with a number from zero to $N_p - 1$. Then, DE generates new points through perturbations of existing points; in other words, DE disturbs vectors with scaled differences of two randomly selected population vectors. Thus, to generate the trial vector, u_0, a scaled random vector difference is added to a third randomly selected population vector. In the selection stage, the trial vector competes against the population vector of the same index. In the step of select-and-save, the vector with the lower objective function value (considering minimization) is a member of the next generation. The procedure repeats until all N_p population vectors have competed against a randomly generated trial vector. Once testing the last trial vector, the survivors of the N_p pairwise competitions become parents for the next generation in the evolutionary cycle [38]. The pseudo-code (Algorithm 1) shows the general idea of DE.

In DE, D-dimensional parameter vectors are defined as the population for each generation G [see Eq. (6)]. The population size is defined with $N_p = 30$, whose individuals are defined as in Eq. (4):

$$q_{t,G}, \quad t = 1, 2, \ldots, N_p. \quad \textbf{(6)}$$

The DE algorithm's primordial functions are mutation, crossover, and selection. Thus, for each vector $\overrightarrow{q_{t,G}}$, $t = 1, 2, \ldots, N_p$, a mutant vector $\overrightarrow{ve}$ is generated as follows:

$$\overrightarrow{ve} = \overrightarrow{q_{r1,G}} + F \cdot (\overrightarrow{q_{r2,G}} - \overrightarrow{q_{r3,G}}). \quad \textbf{(7)}$$

Next, a set of indices randomly created r_1, r_2, r_3 in the population $[1, 2, 3, \ldots, N_p]$ need to be integers, mutually different, and with $F > 0$. Furthermore, r_1, r_2, and r_3 should be different from the running index t; thus, it is necessary that $N_p \geq 4$ to satisfy this condition. F controls the amplification of the differential variation $(\overrightarrow{q_{r2,G}} - \overrightarrow{q_{r3,G}})$. In general, it is a real and constant factor in the range [0,2]. Crossover is used in order to increase the diversity of the perturbed parameter vectors, and hence a trial vector u is used as follows:

$$\overrightarrow{u_{t,G+1}} = (u_{1t,G+1}, u_{2t,G+1}, \ldots, u_{Dt,G+1}), \tag{8}$$

where D is the vector dimension. To build the trial vector u, the following conditions are used:

$$u_{wt,G+1} = \begin{cases} ve_{wt,G+1} & \text{if}(randb(w) \leq CR) \text{ or } w = rnbr(t) \\ q_{wt,G} & \text{if}(randb(w) > CR) \text{ and } w \neq rnbr(t) \end{cases}, \tag{9}$$

where $w = 1, 2, \ldots, D$, and $randb(w)$ is the wth evaluation of a uniform random number generator in the range [0,1]. CR is the crossover constant defined by the user with a value between zero and one. $rnbr(t)$ is a randomly chosen index $w = 1, 2, \ldots, D$. This ensures that $u_{t,G+1}$ will obtain at least one parameter from $ve_{t,G+1}$. Selection of individuals for next-generation $G + 1$ is achieved while comparing the trial vector $u_{t,G+1}$ with the target vector $q_{t,G}$. Because the aim is to minimize the objective function (3), the vector with a lower cost will be retained. The constraint defined in (5) is handled with a tolerance $\epsilon = 1e^{-6}$. Remaining parameters are used as follows: mutation factor $F = 0.5$, crossover constant $CR = 0.8$, and number of generations $G = 200$. These values were obtained experimentally. The last process is repeated until $f(E) = 0.004$ or $G = 200$ is reached.

Once the parameters μ, σ, and P of Eq. (1) are computed through Algorithm 1, these can be used to select the region of the nucleus in the cell. First and foremost, it is worth noting that the ROI depicting darker pixels in the image refers to the mode located leftmost in the image histogram; in other words, the nucleus region (see Fig. 3). Thereby, a linear transformation can be used in this manner to contrast enhancement exclusively over the darker image region. The following transformation is used to achieve this:

$$Y = A\frac{c - a}{b - a}. \tag{10}$$

$A = 255$ in Eq. (10) refers to the maximum gray level value, whereas a and b provide bottom and upper limits (calculate with the mode), and c defines the gray value in the input image. This is illustrated graphically in Fig. 5.

Finally, we note that a and b values in Eq. (10) can be derived from the parameters previously obtained by the DE algorithm using each Gaussian model of Eq. (1). The calculation of these values is done taking into a count the three-sigma rule, which is about 99.7% of values from a normal distribution considering three standard deviations [39]. Hence, from μ_1 and σ_1, the values for a and b are proposed as follows:

$$\begin{aligned} a &= \mu_1 - 3 * \sigma, \\ b &= \mu_1 + 3 * \sigma \ . \end{aligned} \tag{11}$$

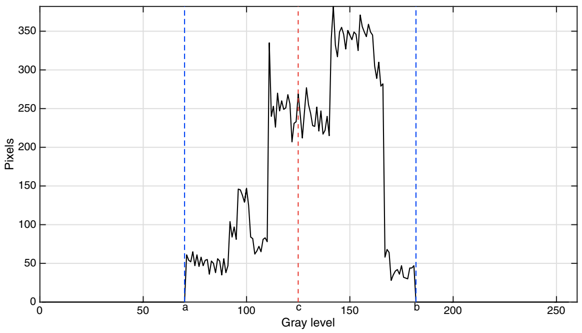
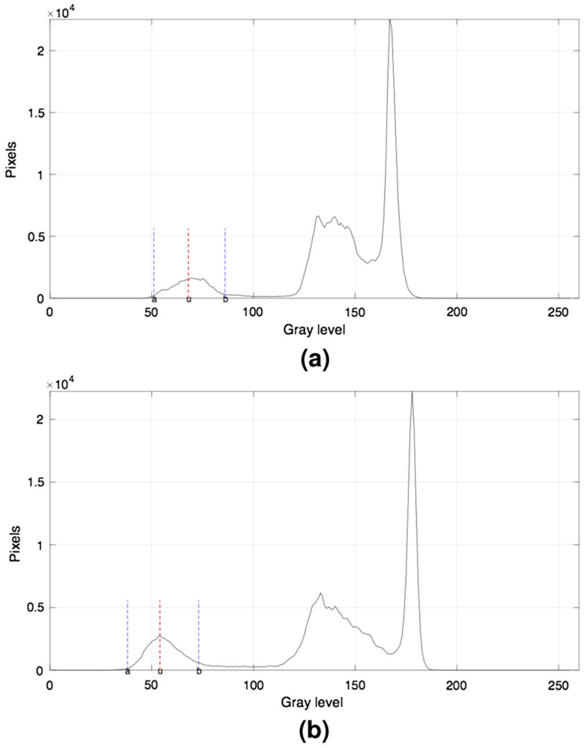


Fig. 5. Graphical representation of parameters used for a contrast enhancement through Eq. (10).



(a)

(b)

Fig. 6. Graphical representation of positions of a and b for images (b) and (c) in Fig. 2: (a) Histogram of healthy cell, and (b) Histogram of leukemic cell.

As a result, with the application of Eq. (10) in the input image, it is possible to outline the nucleus within the image. To illustrate the positions for a and b in the histogram of an image, two histograms showing the positions for these variables are presented in Fig. 6. The histograms belong to images (b) and (c) in Fig. 2.

After applying the process previously described to each image, we obtain the region of the cell nucleus. Post-processing applied to this image allows us to obtain a homogeneous region of the nucleus and, in some cases, to preserve and, in others, highlight the outstanding properties of the cell nucleus. We considered such a process as expert knowledge. Also, this step is a requirement for the extraction of features. Morphological

Table 1. Characteristics of Normal White Blood Cells [26]

Subtype of Blood Cell	Cell Size (µm)	Nucleus	Chromatin
Neutrophil	10–15	2–5 lobes connected by thin filaments without visible chromatin	Coarsely clumped
Band neutrophil	10–15	Constricted and chromatin visible within the thinnest part	Coarsely clumped
Lymphocyte	7–18	Round to oval; may be slightly indented; occasional nucleoli	Condensed to deeply condensed
Monocyte	12–20	Variable; may be round horseshoe or kidney shaped; often has folds producing	Moderately clumped; lacy brain-like convolutions
Eosinophil	12–17	2–3 lobes connected by thin filaments without visible chromatin	Coarsely clumped
Basophil	10–14	Usually two lobes connected by thin filaments without visible chromatin	Coarsely clumped

operators are allowed to join some regions in the image. Edges are extracted with Robert's algorithm [40]. Later, a dilatation operation is applied using a structuring element of a disk with radius five. The algorithm for edge extraction, and the disk size and its shape were selected experimentally through visual inspection. Methods for edge extraction additionally tested were Sobel [41], Canny [42], and Prewitt [43]. Disk geometries such as diamond, octagon, and sphere shapes with sizes between three and seven also were tested.

In the second stage, expert knowledge obtained from this region is later used to recognize the cell.

B. Feature Extraction

We considered features taken from medical literature, where various properties of the cells are depicted as meaningful to their visual identification [44]. In this work, some of these properties are part of the expert knowledge of blood cell recognition. In this regard, the World Health Organization's (WHO's) *Classification of Tumors and Lymphoid Tissues*, including leukemia, is based on morphology, immunophenotyping, genetic features, and clinical features [26]. In this work, we address the leukemia morphology since only white cells are affected. Table 1 shows the characteristics for normal cells as well as the appearance of their nuclei. Thus, in leukemia, diverse changes from these characteristics can be found. Images of cells described in Table 1 can be observed in Fig. 1.

Thus, we selected features to describe the nucleus, considering the properties listed in Table 1. For example, the constricted attribute refers to the narrow shape, typically well defined, while coarsely clumped means the presence of a rough agglomeration in some regions into the nucleus, which in some images can appear as darker subregions as well as with the presence of small holes. Thus, these properties have a relationship with measures of eccentricity and the Euler number, to mention a few.

Considering commonly used features in literature for the recognition of blood cells [7,9–12], we selected those that most closely relate to Table 1. Hence, we consider 17 features: area, diameter ratio, extent, eccentricity, orientation, solidity, Euler number, perimeter, convex area, rectangularity, and the seven Hu's invariants. We apply the MATLAB toolbox of image processing to obtain the first nine features, and rectangularity through the difference between the width and height of the BoundingBox. Hu's moment invariants give a method for objects' descriptions regarding area, position, and orientation, to mention a few. It is important to note that variables used in equations describe the seven Hu's invariants in a new context.

Since the moment of an object is defined by Eq. (12), the moment of order zero refers to the object area:

$$m = \int_{-\infty}^{\infty} x^p y^p f(x, y) \mathrm{d}x \mathrm{d}y, \tag{12}$$

where $p, q = 0, 1, 2, \ldots, p + q$ is the order of the moment, x and y are pixel coordinates relative to an arbitrary order, and $f(x, y)$ is the pixel luminosity. Furthermore, the coordinates of the centroid in terms of moments are defined as follows:

$$x' = \frac{m_{10}}{m_{00}}, \quad y' = \frac{m_{01}}{m_{00}}. \tag{13}$$

Equation (14) defines invariant moments derived by computing the central moment μ involving the centroid. Thus, central moments are invariant to image translations:

$$\mu = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x')^p (y - y')^q f(x, y) \mathrm{d}x \mathrm{d}y. \tag{14}$$

After we obtain scale invariance η by normalization of central moments through Eq. (15),

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \gamma = (p + q + 2)/2, \; p + q = 2, 3, \ldots. \tag{15}$$

Finally, we compute the seven-moment invariants Φ_1–Φ_7 introduced by Hu [45] through Eqs. (16)–(22). These values are the seven features of Hu's invariants, which remain unchanged to image scaling, translation, and rotation:

$$\Phi_1 = \eta_{20} + \eta_{02}, \tag{16}$$

$$\Phi_2 = \eta_{20} + \eta_{02}^2 + 4\eta_{11}^2, \tag{17}$$

$$\Phi_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2, \tag{18}$$

$$\Phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \tag{19}$$

$$\begin{aligned} \Phi_5 = {} & (\eta_{30} + 3\eta_{12}) + (\eta_{30} + \eta_{12}) \\ & \times [(\eta_{30} + \eta_{12})^2) - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} - \eta_{03}) \\ & \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \end{aligned} \tag{20}$$

$$\begin{aligned} \Phi_6 = {} & (\eta_{20} + \eta_{02}) + [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} - \eta_{03}), \end{aligned} \tag{21}$$

$$\Phi_7 = (3\eta_{21} + \eta_{03})(\eta_{30} + \eta_{12}) + [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2].$$
$$- (\eta_{30} - 3\eta_{12})(\eta_{21} - \eta_{03})[3(\eta_{30} - \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] \quad \textbf{(22)}$$

All features are scalar values, which are normalized in the range [0,1] using a normalization type min-max for each feature. We do not consider any process of reduction in the selected features, or geometric transformations from the images.

C. Classifiers

As mentioned in Section 1, we made the selection of classifiers based on the requirement of using or not using data derived from expert knowledge, which classically is managed by the classifier. For this purpose, we distinguish the following classifiers.

1. Multilayer Perceptron

MLPs are a specific type of artificial neural network. These nets are composed of basic computing units stacked in multiple layers to form a feedforward network (input, hidden, and output layers). Each neuron receives an input $Xm = \{Xm_1, \ldots, Xm_{nF}\}$, with $Xm \in R^{nF}$, and computes and stores a weighted linear sum of the signal as in Eq. (23):

$$Om = \sum_{cm=1}^{nF} we_{cm} Xm_{cm} + \text{bias}, \quad \textbf{(23)}$$

where we_{cm} biases are the weights associated with the neuron. Om from each neuron then passes through an activation function (AF), to supply the output transmitted by the neuron. In this work, the backpropagation algorithm helps to train the net. We consider two hidden layers: the first layer with 17 neurons (number of features) and two neurons in the output layer. The AF used is variable learning rate backpropagation, which is given by the function *traingdx* in the MATLAB toolbox of deep learning.

2. Support Vector Machine

A SVM is a kernel method that builds an optimal hyperplane in the form of a decision surface that maximizes the separation margin between the two classes in the data. Support vectors refer to a small subset of the training observations used to support the optimal location of the decision surface. In this work, a nonlinear SVM with the discriminate hyperplane defined in Eq. (24) is used:

$$fc(De) = \sum_{nd=1}^{nTa} \alpha_{nd} Ta_{nd} Ker(De_{nd}, De) + bi, \quad \textbf{(24)}$$

where fc associates descriptors De_{nd} to labels Ta_{nd} for the number of classes nTa. Training data are associated to (De_{nd}, Ta_{nd}), Ker is the kernel function (in this work a Gaussian kernel), and $Ta_{nd} \in \{-1, 1\}$. The sign of the output refers to the class membership of De.

In the first phase, the input data transform into a high-dimensional feature space where the kernel is specified. After, the optimization problem finds the optimal hyperplane to classify the features into two classes.

3. Convolutional Neural Network

A CNN is a type of machine learning in which a model learns to perform classification tasks directly from images, videos, texts, or sounds. This learning takes place through the location of patterns in images directly from the image data without considering any knowledge or meaning about the image. LeNet [46] and AlexNet [47] are CNNs used in this work because they represent, respectively, a standard CNN and a CNN with transfer learning.

The LeNet architecture used consists of four sets of convolutional and pooling layers, followed by a flattening convolutional layer, then two fully connected layers, and finally, a softmax classifier. The input for the net can be a grayscale or color image. The fully connected softmax output layer is two possible values that correspond to each class.

AlexNet is a pre-trained net that is eight layers deep and requires an image input size of 227 × 227 × 3, where 3 is the number of color channels. The last three layers of the net are configured for 1000 classes. Thus, these layers are fined-tuned for our problem of classification, replacing them with a fully connected layer, a softmax layer, and a classification output layer.

3. EXPERIMENTAL RESULTS AND DISCUSSION

All algorithms were executed on a standalone desktop computer. We used a CPU from Intel Core i9- 7900X CPU 3.31 Ghz,

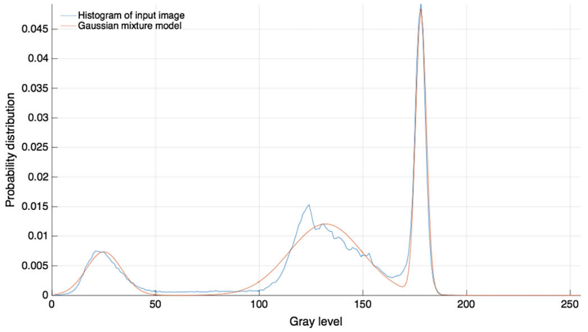


Fig. 7. Gaussian mixture distribution.

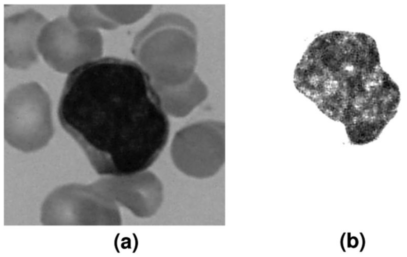


Fig. 8. Results of linear transformation to image (c) in Fig. 2: (a) Lymphocyte cell, and (b) Lymphocyte cell after linear transformation.

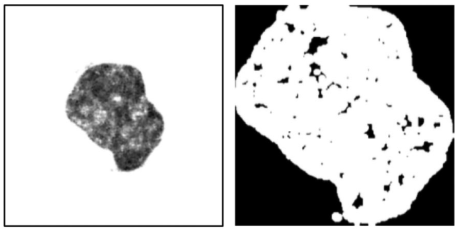


Fig. 9. Input images for tests with CNNs.

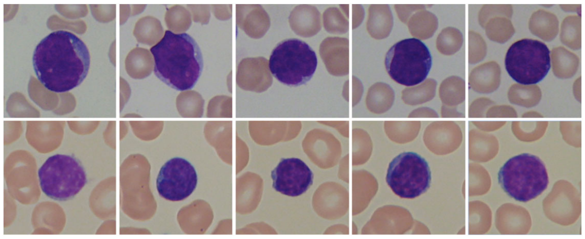


Fig. 10. Leukemic and healthy cells.

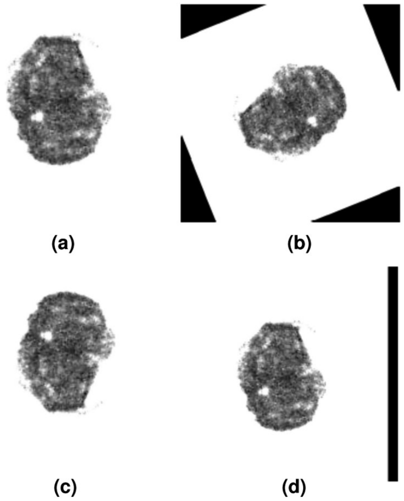


Fig. 11. Results from image data augmentation: (a) Input image, (b) Rotation, (c) Reflection, and (d) Translation.

Table 2. Results of Classification with CNNs Using Original Images from ALL_IDB2

Num. of Exec.	Without Image Data Augmentation				With Image Data Augmentation			
	LeNet		AlexNet		LeNet		AlexNet	
	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)
1	82.05	71.22	97.44	250.83	84.62	62.28	96.15	251.48
2	82.05	54.45	98.72	240.92	69.23	33.78	98.72	245.48
3	84.62	67.03	97.44	229.41	74.36	63.72	98.72	236.55
4	69.23	46.64	98.72	226.92	87.18	84.19	98.72	224.61
5	74.36	41.92	92.31	233.86	70.51	44.78	97.44	242.61
6	85.90	68.08	97.44	232.13	94.87	88.27	98.72	224.03
7	74.36	52.86	96.15	234.78	76.92	52.41	97.44	238.78
8	82.05	44.86	96.15	229.86	79.49	59.16	96.15	231.22
9	80.77	39.27	98.72	229.11	83.33	59.45	94.87	232.98
10	82.05	45.97	94.87	232.73	89.74	80.88	92.31	233.75
11	82.05	49.98	97.44	234.14	88.46	79.36	97.44	242.69
12	78.21	28.50	93.59	232.56	85.90	63.58	97.44	229.33
13	79.49	67.08	93.59	239.14	91.03	78.45	100	231.25
14	91.03	71.16	97.44	228.31	67.95	39.14	98.72	226.59
15	92.31	90.16	97.44	235.38	83.33	62.17	98.72	230.56
16	91.03	111.06	96.15	232.59	79.49	66.64	100.00	234.42
17	80.77	70.28	97.44	231.53	89.74	86.02	97.44	236.06
18	80.77	31.91	98.72	237.36	78.21	57.83	96.15	236.06
19	80.77	37.09	97.44	229.30	88.46	55.02	96.15	225.30
20	84.62	66.20	97.44	232.72	91.03	52.14	100.00	234.31
21	66.67	31.69	100.00	238.27	79.49	47.88	93.59	231.97
22	70.51	46.22	94.87	232.19	85.90	65.56	96.15	236.69
23	85.90	79.67	98.72	231.13	80.77	57.75	97.44	235.95
24	83.33	46.66	98.72	231.25	88.46	79.17	98.72	235.80
25	69.23	35.22	97.44	227.36	80.77	69.38	100.00	237.83
26	92.31	86.50	100.00	227.42	76.92	63.33	100.00	237.83
27	79.49	64.53	97.44	230.88	83.33	82.09	97.44	233.16
28	84.62	52.86	97.44	240.66	83.33	44.73	100.00	233.88
29	80.77	45.38	97.44	247.23	97.44	62.48	98.72	236.70
30	88.46	79.81	94.87	241.03	84.62	41.92	98.72	231.58
Average	81.32	57.48	97.05	234.03	83.16	62.78	**97.74**	234.70
σ^2	±44.85		±3.42		±51.32		±3.70	
σ	±6.69		±1.84		±7.16		±1.92	
Outlier detected?	No		No		No		No	
Critical value of Z:	2.90		2.90		2.90		2.90	

64GB RAM, 222Gb hard drive, 64-bit Windows10 Enterprise Edition operating system, graphics processing unit (GPU) GeForceGTX 1080, and MATLAB R2018a.

The first stage is used to isolate the region of the cell nucleus according to the method described in Section 2.A. Then, the aim is to generate a probability distribution closest to the histogram of each image. Estimation of the parameters in Eq. (1) was calculated through 30 executions of Algorithm 1. Figure 7 illustrates the correlation between the histogram of input image v, and GMM p obtained particularly for image (c) in Fig. 2.

As a last step, μ_i, σ_i, and P_i are obtained, and the values for μ_1 and σ_1 are used to compute a and b in Eq. (11). These parameters belong to the leftmost mode in the image histogram, i.e., the distribution of gray levels of the darker image region (the nucleus). Hence, an output image after applying the linear transformation defined in Eq. (10) is obtained [see Fig. 8(b)]. Then, as a result of the post-processing described in Section 2.B, we obtain a binary image for the extraction of features used later by the classifiers SVM and MLP. An image with post-processing is illustrated on the right side in Fig. 9.

Inputs for CNNs include grayscale and binary images from the isolated nucleus and color images (original images from ALL_IDB2), respectively, for each test. The first image set refers to images without post-processing, as shown in the image on the left in Fig. 9.

In the following stage, we assess the influence of using expert knowledge by using the classifiers described in Section 2.C.

Table 3. Results of Classification with CNNs Using Data Augmentation with ALL_IDB2

Num. of. Exec.	Images without Post-Processing				Images with Post-Processing			
	LeNet		AlexNet		LeNet		AlexNet	
	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)
1	66.67	56.45	92.31	271.63	58.97	63.44	85.90	251.23
2	75.64	57.02	92.31	271.83	53.85	49.47	92.31	249.13
3	73.08	56.08	89.74	259.75	57.69	34.86	89.74	238.31
4	74.36	71.03	87.18	256.72	65.38	69.50	93.59	253.14
5	76.92	62.38	89.74	266.78	44.87	52.23	88.46	246.19
6	66.67	42.20	87.18	263.55	52.56	43.50	85.90	242.05
7	70.51	54.50	85.90	258.45	52.56	34.67	92.31	245.14
8	71.79	51.98	91.03	261.92	47.44	41.27	88.46	237.70
9	62.82	39.47	88.46	261.86	48.72	52.50	88.46	246.30
10	80.77	63.23	84.62	258.38	51.28	34.09	85.90	243.38
11	56.41	54.03	88.46	262.64	52.56	52.14	85.90	241.31
12	67.95	47.59	91.03	249.52	60.26	51.64	85.90	240.88
13	71.79	60.56	92.31	258.97	56.41	33.14	89.74	240.78
14	67.95	38.72	88.46	264.14	62.82	48.81	92.31	244.55
15	64.10	52.88	87.18	263.48	51.28	58.64	84.62	232.00
16	78.21	61.39	88.46	259.63	57.69	36.58	91.03	237.41
17	76.92	66.16	91.03	263.55	61.54	70.06	87.18	230.63
18	73.08	46.66	92.31	264.45	53.85	42.50	87.18	229.81
19	66.67	50.94	85.90	260.53	53.85	53.28	83.33	237.08
20	60.26	45.02	88.46	259.67	60.26	46.28	94.87	233.48
21	73.08	44.39	92.31	264.17	51.28	63.89	85.90	236.11
22	61.54	33.14	85.90	259.95	67.95	51.52	89.74	240.80
23	76.92	47.44	94.87	258.59	57.69	30.47	89.74	234.20
24	67.95	64.97	91.03	263.64	56.41	47.91	92.31	229.22
25	65.38	66.66	89.74	262.73	60.26	66.95	92.31	234.64
26	67.95	43.03	89.74	261.48	50.00	37.63	87.18	232.81
27	71.79	35.44	91.03	260.66	55.13	32.92	84.62	237.98
28	70.51	53.78	91.03	258.55	69.23	55.98	91.03	235.16
29	70.51	45.36	89.74	254.39	55.13	55.38	89.74	236.28
30	50.00	44.34	83.33	268.34	57.69	55.73	91.03	237.22
Average	69.27	51.89	**89.36**	261.67	56.15	48.90	88.89	239.26
σ^2	±45.17		±7.05		±33.15		±9.11	
σ	±6.72		±2.65		±5.75		±3.01	
Outlier detected?	No		No		No		No	
Critical value of Z:	2.90		2.90		2.90		2.90	

Table 4. Results of Classification with CNNs without Data Augmentation with ALL_IDB2

Num. of. Exec.	Images without Post-Processing				Images with Post-Processing			
	LeNet		AlexNet		LeNet		AlexNet	
	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)	Acc.	Time(s)
1	70.51	99.67	89.74	260.53	57.69	54.52	74.36	262.61
2	65.38	36.50	92.31	253.77	50.00	60.56	69.23	264.02
3	83.33	50.58	91.03	246.89	50.00	46.48	84.62	253.14
4	75.64	39.00	91.03	250.27	51.28	35.09	75.64	249.59
5	78.21	40.94	88.46	260.77	50.00	48.08	79.49	257.95
6	76.92	84.27	87.18	247.25	56.41	42.38	74.36	246.11
7	69.23	37.03	92.31	251.38	50.00	68.38	79.49	249.98
8	76.92	36.13	87.18	247.53	51.28	57.28	73.08	247.19
9	52.56	35.16	89.74	250.42	47.44	84.69	82.05	251.78
10	66.67	30.63	92.31	246.02	51.28	50.27	75.64	251.22
11	82.05	72.44	93.59	244.22	53.85	43.91	71.79	248.02
12	76.92	75.81	85.90	245.48	48.72	41.33	84.62	246.05
13	73.08	46.16	94.87	246.22	51.28	37.55	83.33	252.30
14	66.67	47.91	91.03	251.28	50.00	63.38	73.08	252.39
15	71.79	32.06	82.05	247.91	53.85	54.73	82.05	247.78
16	71.79	43.02	91.03	247.30	50.00	35.05	79.49	245.61
17	85.90	49.06	93.59	246.58	60.26	35.23	78.21	248.75
18	73.08	47.78	94.87	244.78	50.00	44.64	76.92	254.14
19	83.33	39.28	92.31	248.88	57.69	39.16	79.49	254.92
20	74.36	44.34	91.03	248.75	57.69	65.95	80.77	253.56
21	60.26	46.66	94.87	247.80	50.00	45.23	76.92	252.02
22	51.28	35.66	87.18	249.44	50.00	68.52	75.64	249.56
23	53.85	37.27	91.03	251.20	51.28	56.91	76.92	249.72
24	76.92	41.98	83.33	250.63	53.85	57.36	79.49	259.58
25	65.38	62.38	85.90	252.25	50.00	39.84	74.36	255.80
26	73.08	42.28	93.59	247.48	50.00	49.33	80.77	252.41
27	75.64	44.61	80.77	247.44	50.00	38.50	79.49	253.28
28	71.79	57.80	92.31	246.94	69.23	54.91	88.46	255.14
29	71.79	36.08	91.03	244.22	48.72	29.06	79.49	258.34
30	79.49	49.17	84.62	245.19	50.00	33.97	74.36	251.02
Average	71.29	48.05	**89.87**	248.96	52.39	49.43	78.12	252.47
σ^2	±76.06		±14.66		±20.15		±18.13	
σ	±8.72		±3.82		±4.48		±4.25	
Outlier detected?	No		No		No		No	
Critical value of Z:	2.90		2.90		2.90		2.90	

Original images (in color) have distinctive background tonality for each class: leukemic and healthy cells are noticeably different. Color is an essential trait for the extraction of features with CNNs that influences the classification task, and it is not desirable because clearly the aim is to assess the importance of the use of expert knowledge to recognize the cell of interest and not other items in the image background. Figure 10 shows leukemic cells in the first row and healthy cells in the second row. In the second row, a more intense reddish color can be seen in the background than in the images in the first row.

Experiments with CNNs include tests with and without image data augmentation; this set of transformations for image augmentation contains rotation, translation, and reflection. Data augmentation prevents network overfitting, as it helps with memorizing traits of the training images. We considered rotation with an angle between 0° and 360°, and for translation, a distance within the interval of $[-30, 30]$ measured in pixels. Values for angle and distance in the transformations were picked randomly from a continuous uniform distribution within a specified interval. Nevertheless, only the horizontal and vertical reflections were used in the final test because images are meaningfully affected by rotation and translation, as can be seen in Fig. 11.

On the other hand, because the input image for the pretrained CNN AlexNet is an RGB color image, in the test with grayscale and binary images, these were duplicated over the three channels.

Table 5. Results of Classification with AlexNet Using DATA Augmentation with ALL_IDB1

No. Execution	Accuracy	Time (s)	No. Execution	Accuracy	Time (s)
1	93.94	351.16	16	100.00	341.14
2	93.94	343.88	17	96.67	339.11
3	100	341.27	18	100.00	335.78
4	100	340.23	19	100.00	341.44
5	100.00	338.34	20	100.00	343.56
6	100.00	341.84	21	100.00	339.95
7	96.67	341.36	22	100.00	339.95
8	100.00	337.31	23	100.00	339.39
9	96.97	342.84	24	100.00	339.28
10	100.00	344.05	25	100.00	338.02
11	100.00	340.83	26	100.00	339.05
12	100.00	340.33	27	100.00	339.92
13	100.00	342.47	28	96.67	341.59
14	100.00	338.30	29	100.00	335.81
15	100.00	340.41	30	100.00	341.27
Average:				**99.19**	340.85
σ^2				±3.12	
σ				±1.77	
Outlier detected?				Yes	
Critical value of Z:				2.90	

Table 6. Results for Classification with MLP and SVM Using the 17 Features Extracted from ALL_IDB2

	MLP		SVM	
No. Exec.	Accuracy	Time(s)	Accuracy	Time(s)
1	99.46	2.89	97.00	0.63
2	99.46	0.55	100.00	0.07
3	99.85	0.50	97.00	0.07
4	99.62	0.51	100.00	0.02
5	99.54	0.52	100.00	0.11
6	99.46	0.50	100.00	0.01
7	99.62	0.53	99.00	0.01
8	99.69	0.59	99.00	0.01
9	99.62	0.49	99.00	0.01
10	99.46	0.51	99.00	0.01
11	99.62	0.50	99.00	0.01
12	99.62	0.56	100.00	0.01
13	99.69	0.53	99.00	0.01
14	99.77	0.51	100.00	0.01
15	99.69	0.50	100.00	0.01
16	99.69	0.50	100.00	0.01
17	99.69	0.50	97.00	0.01
18	99.69	0.48	100.00	0.01
19	99.54	0.47	100.00	0.01
20	99.69	0.48	99.00	0.01
21	99.54	0.49	99.00	0.01
22	99.54	0.49	99.00	0.01
23	99.85	0.49	100.00	0.01
24	99.69	0.46	97.00	0.01
25	99.69	0.52	99.00	0.01
26	99.77	0.61	99.00	0.01
27	99.62	0.47	100.00	0.01
28	99.54	0.48	100.00	0.01
29	99.62	0.48	100.00	0.01
30	99.62	0.48	97.00	0.01
Average	**99.63**	0.59	99.00	0.04
σ^2	±0.016		±0.00	
σ	±0.11		±0.01	
Outlier detected?	No		No	
Critical value of Z:	2.90		2.90	

Because of the small dataset size, we realize 30 executions for each experiment, with different training and test sets for each implementation. Thus, to validate the method, the accuracy is measured with the average of these executions. In all experiments, we use 70% of data for training and the remainder for testing. This approach, known as train/split, was designed to avoid strongly biased yield estimates even with small sample sizes and to produce robust and unbiased performance estimates, regardless of sample size [48].

We consider the Grubbs statistical method for outlier detection [49]. We run the experiments using Grubbs' test from the results of every experiment, with a significance level of 0.05 (two-sided). Values of average, variance, standard deviation, outliers detected, and critical value Z are reported in Tables 2–6. According to the results, there are no outliers, except in the experiment using ALL_IDB1 and AlexNet with image data augmentation (Table 5). Nevertheless, this is not significant because only one outlier was detected. Moreover, in both experiments with the original images using ALL_IDB1 and ALL_IDB2, the results are similar: 99.19 and 97.74, respectively, meaning the results are inside the expected range even though ALL_IDB1 is unbalanced, with fewer images. All tables provide also average time in seconds for the experiments.

Results of experiments with CNNs for three image sets—original, without post-processing (grayscale), and with post-processing (binary)—using data augmentation and without data augmentation are presented, respectively, in Tables 2–4. Note that tests with CNNs–AlexNet using image data augmentation achieved the best performance.

Also, results for classifiers MLP and SVM are in Table 6, using as input the 17 extracted features from the post-processing applied to the nucleus images. In order to give a summary of outcomes from classifiers, Table 7 provides the best average results for each set of experiments. In contrast, expected results with color images are better than the other sets of images due to characteristics of color background. AlexNet reaches 97.74% in color images and 89.87% of accuracy using images of cell nuclei without post-processing, while LeNet obtains 71.79%. Moreover, the use of image data augmentation favored the results of both CNNs in tests with binary images, while in tests without post-processing, the changes are not meaningful, obtaining an absolute-accuracy difference of 2.52 and 0.51 for LeNet and AlexNet, respectively. On the other hand, in tests with the classifiers MLP and SVM, the best results are obtained with MLP reaching 99.63% accuracy, while for SVM, 99.00%, both slightly better than AlexNet.

Finally, Fig. 12 shows a graphical comparison between processing times for executions with the best classifiers. Notice that time consumed for the classifiers that use expert knowledge is meaningfully lower than for those that do not use it. Despite GPU deep learning, parallel computing is used to speed up

Table 7. Summary of Outcomes from Classification with CNNs, MLP, and SVM Applied to ALL_IDB2

	With Data Augmentation			Without Data Augmentation		
Classifier Used	Original images	Imgs. without post-proc.	Imgs. with post-proc.	Original images	Imgs. without post-proc.	Imgs. with post-proc.
LeNet	83.16	69.27	56.15	81.32	71.79	52.39
AlexNet	97.74	89.36	88.89	97.05	89.87	78.12
	Features extracted from images with post-processing					
MLP	99.63					
SVM	99.00					

Table 8. Outcomes of Time from Classification with CNNs, MLP, and SVM Applied to ALL_IDB2

	With Data Augmentation			Without Data Augmentation		
Classifier Used	Original images	Imgs. without post-proc.	Imgs. with post-proc.	Original images	Imgs. without post-proc.	Imgs. with post-proc.
LeNet	62.78	51.89	48.90	57.48	48.05	49.43
AlexNet	234.70	261.67	239.26	234.03	248.96	252.47
	Features extracted from images with post-processing					
MLP	0.59					
SVM	0.04					

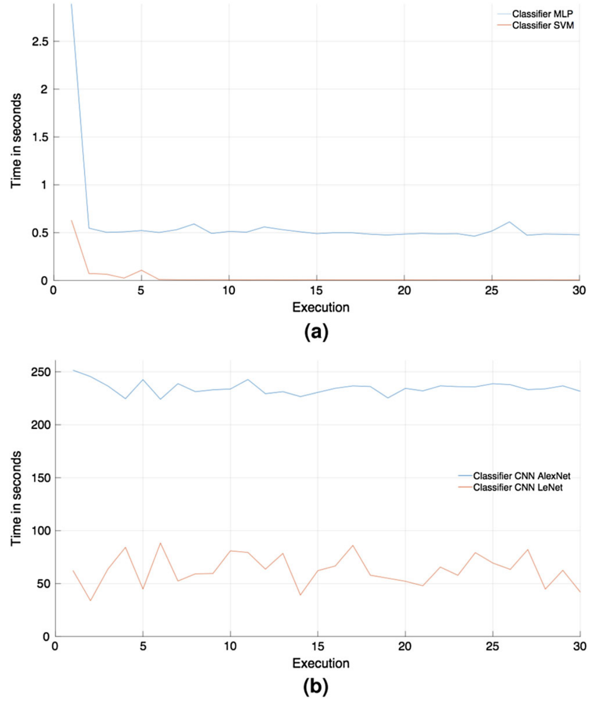


Fig. 12. Graphical representation of times best results for each classifier: (a) Time for classifiers that used expert knowledge as input data, and (b) Time for classifiers thad did not use expert knowledge as input data.

the computations during the training of CNNs. In Table 8, a summary of times (in seconds) registered for results of Table 7 is presented. Experiments with AlexNet depict that image data augmentation is time consuming when using images without post-processing, and the opposite occurs when binary images are used. In contrast, with original images, there are no significant changes with the use of image data augmentation. In experiments with LeNet, the use of image data augmentation and the type of image used as input do not provoke great changes in the time used since the difference is of 3.84 when using images without post-processing, and 0.53 for binary images. In summary, it is worth mentioning that classifiers that use expert knowledge have lower time consumption.

4. CONCLUSION

We present a technique to derive expert knowledge for leukemic cell recognition. The method consists of a GMM, an evolutionary algorithm, and techniques of image processing, which allow us to obtain the ROI within an image for *a posteriori* recognition, thus encapsulating appropriate expert knowledge. Experiments demonstrate that the use of this expert knowledge is useful to approach visual recognition of blood cells with hematological diseases such as leukemia. In this regard, the use of standard techniques of image processing provides a set of descriptors that in combination with classifiers such as SVMs and MLPs surpasses the accuracy and performance of popular (last generation) CNNs such as LeNet and AlexNet. Although results are similar for MLP and AlexNet (using transfer learning and image data augmentation), in the latter, a reduction in image size is required. Moreover, we obtain the results using images from ALL_IDB2, in which the characteristics of the image background influence the performance of the net, as mentioned in Section 3. Thus, from experiments with ALL_IDB1 and ALL_IDB2, we note the critical problem of CNNs of solving the problem without selecting the ROI, which is undesirable since experts need to identify the disease based on the features of the nucleus cell.

Moreover, the proposed method derives this from expert knowledge; as a result, it is fully explainable since the extraction of features follows a logical process using human reasoning for the recognition of leukemic cells, in contrast to CNNs, where the process to derive the extraction of features is unknown, which is one main drawback of deep learning. This work demonstrates through experimental results with datasets ALL_IDB1 and ALL_IDB2 that deep learning correctly classifies both problems, raising the question of knowing under which criteria the system applies for this recognition. It is evident that the traits used by a human expert focusing only on deep learning techniques cannot easily incorporate expert knowledge. In brief, the results of these models go beyond human perception, i.e., they are models incapable of providing explainable reasoning regarding the prediction made. Instead, by using expert knowledge, more accurate and robust explainable models can be obtained, which in the future will allow the creation of better learning and interpretation techniques for solutions to problems in various fields of knowledge.

Finally, in future work, we would like to expand the proposed approach to other fields while considering solutions using expert knowledge, low computational requirements, excellent performance, and symbolic learning.

Funding. Universidad Autónoma de Tlaxcala.

Acknowledgment. This research was supported in part by the Computer Research Center of National Polytechnical Institute (projects 20190007 and 20200630), CICESE Research Center (project 634-135), and by the Autonomous University of Tlaxcala, Mexico. R. Ochoa acknowledges CONACYT for the scholarship granted towards pursuing her PhD studies.

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. R. Kumar, R. Srivastava, and S. Srivastava, "Detection and classification of cancer from microscopic biopsy images using clinically significant and biologically interpretable features," J. Med. Eng. **2015**, 1 (2015).
2. M. Brereton, B. D. L. Salle, J. Ardern, K. Hyde, and J. Burthem, "Do we know why we make errors in morphological diagnosis? An analysis of approach and decision-making in haematological morphology," EBioMedicine **2**, 1224–1234 (2015).
3. D. Gómez-Almaguer, E.-R. Marcos-Ramírez, E.-H. Montaño-Figueroa, G.-J. Ruiz-Argüelles, C.-R. Best-Aguilera, M. del Carmen López-Sánchez, E. Barrera-Chairez, J.-L. López-Arrollo, C.-O. Ramos-Peñafiel, A. L.-P. Na, E.-E. González-López, P.-E. Rivas-García, C.-A. Tellez-Hinojosa, A. Gómez-De-León, and J.-C. Jaime-Pérez, "Acute leukemia characteristics are different around the world: the Mexican perspective," Clin. Lymphoma Myeloma Leuk. **17**, 46–51 (2017).
4. P. Colunga-Pedraza, G. Gomez-Cruz, J. Colunga-Pedraza, and G. Ruiz-Argüelles, "Geographic hematology: some observations in Mexico," Acta Haematol. **140**, 114–120 (2018).
5. S.-H. Rezatofighi and H. Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," Comput. Med. Imaging Graphics **35**, 333–343 (2011).
6. A.-S. Abdul-Nasir, M.-Y. Mashor, N.-H. Harun, and R. Hassan, "Colour image enhancement techniques for acute leukaemia blood cell morphological features," in *IEEE International Conference on Systems, Man and Cybernetics* (2010), pp. 3677–3682.
7. J. Laosai and K. Chamnongthai, "Classification of acute leukemia using medical-knowledge-based morphology and CD marker," Biomed. Signal Process. Control **44**, 127–137 (2018).
8. J. Rawat, A. Singh, H. S. Bhadauria, J. Virmani, and J.-S. Devgun, "Computer assisted classification framework for prediction of acute lymphoblastic and acute myeloblastic leukemia," Biocybern. Biomed. Eng. **37**, 637–654 (2017).
9. A.-S. Negm, O.-A. Hassan, and A.-H. Kandil, "A decision support system for acute leukaemia classification based on digital microscopic images," Alexandria Eng. J. **57**, 2319–2332 (2018).
10. G. Jothi, I. Hannah, T.-A. Ahmad, and D. K. Renuga, "Rough set theory with JAYA optimization for acute lymphoblastic leukemia classification," Neural Comput. Appl. **31**, 5175–5194 (2018).
11. A.-T. Sahlol, A.-M. Abdeldaim, and A.-E. Hassanien, "Automatic acute lymphoblastic leukemia classification model using social SPIDER optimization algorithm," Soft Comput. **23**, 6345–6360 (2018).
12. Z. Moshavash, H. Danyali, and M.-S. Helfroush, "An automatic and robust decision support system for accurate acute leukemia diagnosis from blood microscopic images," J. Digit. Imaging **31**, 702–717 (2018).
13. C. Fatichah, M.-L. Tangel, F. Yan, J.-P. Betancourt, R. Widyanto, F. Dong, and K. Hirota, "Fuzzy feature representation for white blood cell differential counting in acute leukemia diagnosis," Int. J. Control Autom. Syst. **13**, 742–752 (2015).
14. E.-A. Mohammed, M.-M.-A. Mohamed, C. Naugler, and B.-H. Far, "Toward leveraging big value from data: chronic lymphocytic leukemia cell classification," Netw. Model. Anal. Health Inf. Bioinf. **6**, 6–23 (2017).
15. K. K. Jha and H. S. Dutta, "Mutual information based hybrid model and deep learning for acute lymphocytic leukemia detection in single cell blood smear images," Comput. Methods Programs Biomed. **179**, 104987 (2019).
16. S. Hosseinzadeh-Kassani, P. Hosseinzadeh-Kassani, M. J. Wesolowski, K. Schneider, and R. Deters, "A hybrid deep learning architecture for leukemic B-lymphoblast classification," in *The 10th International Conference on ICT Convergence At: Jeju Island, South Korea* (2019), pp. 1–7.
17. R. Wieland and W. Mirschel, "Combining expert knowledge with machine learning on the basis of fuzzy training," Ecol. Inf. **38**, 26–30 (2017).
18. L.-E.-S. Vogado, R.-M.-S. Veras, F.-H.-D. Araujo, R.-R.-V. Silva, and K.-R.-T. Aires, "Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification," Eng. Appl. Artif. Intell. **72**, 415–422 (2018).
19. C. Matek, S. Schwarz, K. Spiekermann, and C. Marr, "Human-level recognition of blast cells in acute myeloid leukaemia with convolutional neural networks," Nat. Mach. Intell. **1**, 538–544 (2019).
20. F. Qin, N. Gao, Y. Peng, Z. Wu, S. Shen, and A. Grudtsin, "Fine-grained leukocyte classification with deep residual learning for microscopic images," Comput. Methods Prog. Biomed. **162**, 243–252 (2018).
21. J. Zhao, M. Zhang, Z. Zhou, J. Chu, and F. Cao, "Automatic detection and classification of leukocytes using convolutional neural networks," Med. Biol. Eng. Comput. **55**, 1287–1301 (2017).
22. P. Huang, S. Zhang, M. Li, J. Wang, C. Ma, B. Wang, and X. Lv, "Classification of cervical biopsy images based on lasso and EL-SVM," IEEE Access **8**, 24219–24228 (2020).
23. W. Ding, "SVM-based feature selection for differential space fusion and its application to diabetic fundus image classification," IEEE Access **7**, 149493 (2019).
24. D. Ray and J.-S. Hesthaven, "Detecting troubled-cells on two-dimensional unstructured grids using a neural network," J. Comput. Phys. **397**, 108845 (2019).
25. S. D. Thepade and D. Abin, "Face gender recognition using multi layer perceptron with Otsu segmentation," in *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* (2018), pp. 1–5.
26. B. Rodak and J. Carr, *Clinical Hematology Atlas*, 5th ed. (Elsevier, 2016).
27. R. Labati, V. Piuri, and F. Scotti, "ALL-IDB: The acute lymphoblastic leukemia image database for image processing," in *18th IEEE International Conference on Image Processing* (2011), pp. 2045–2048.
28. S. Xia, Y. Xia, H. Yu, Q. Liu, Y. Xia, Y. Luo, G. Wang, and Z. Chen, "Transferring ensemble representations using deep convolutional neural networks for small-scale image classification," IEEE Access **7**, 168175 (2019).
29. M. Dong, K. Pang, Y. Wu, J.-H. Xue, T. Hospedales, and T. Ogasawara, "Transferring CNNs to multi-instance multi-label classification on small datasets," in *IEEE International Conference on Image Processing (ICIP)* (2017), pp. 1332–1336.
30. R. Gonzalez and R. Woods, *Digital Image Processing*, 4th ed. (Pearson, 2017).
31. R. Ochoa-Montiel, O. Flores-Castillo, H. Sossa, and G. Olague, "Automatic contrast enhancement with differential evolution for leukemia cell identification," in *Pattern Recognition. MCPR*, LNCS (Springer Cham, 2019), Vol. **11524**, pp. 282–291.
32. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979).
33. S. Prince, *Computer Vision, Models, Learning and Inference*, 1st ed. (Cambridge University, 2012).

34. R. Storn and K. Price, "Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces," J. Global Optim. **11**, 341–359 (1997).
35. F. Chen, J. Shi, Y. Ma, Y. Lei, and M. Gong, "Differential evolution algorithm with learning selection strategy for SAR image change detection," in *IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2017), pp. 450–457.
36. S. Sehgal, S. Kumar, and M. Bindu, "Remotely sensed image thresholding using Otsu and differential evolution approach," in *7th International Conference on Cloud Computing, Data Science and Engineering—Confluence (Confluence)* (2017), pp. 138–142.
37. J. Branke, K. Deb, K. Miettinen, and R. Slowiński, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, 1st ed. (Springer, 2008), Vol. **5252**.
38. K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, 1st ed. (Springer, 2005).
39. E. Grafarend, *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Mixed Models*, 1st ed. (Walter de Gruyter GmbH & Co., 2006).
40. L. Roberts, *Machine Perception of Three-Dimensional Solids, Outstanding Dissertations in the Computer Sciences* (Garland Publishing, 1963).
41. I. Sobel, "An isotropic 3 × 3 image gradient operator," Presentation at Stanford A.I. Project 1968 (1968).
42. J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**, 679–698 (1986).
43. J. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics* (1970), pp. 75–149.
44. K. Foucar, D. Chabot-Richards, D. Cazuchlewski, K.-H. Karner, K. Reichard, M. Vasef, C. Wilson, Q.-Y. Zhang, and K. Culbreath, *Diagnostic Pathology: Blood and Bone Marrow*, 2nd ed. (Diagnostic Pathology, 2017).
45. M.-K. Hu, "Visual pattern recognition by moment invariants," IRE Trans. Inf. Theory **8**, 179–187 (1962).
46. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**, 2278–2324 (1998).
47. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012).
48. A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," PLoS One **14**, e0224365 (2019).
49. F. E. Grubbs, "Sample criteria for testing outlying observations," Ann. Math. Stat. **21**, 27–58 (1950).