

## Article

# Interpretable Lightweight Ensemble Classification of Normal versus Leukemic Cells

Yúri Faro Dantas de Sant'Anna , José Elwyslan Maurício de Oliveira  and Daniel Oliveira Dantas \* 

Departamento de Computação, Universidade Federal de Sergipe, São Cristóvão 49100-000, SE, Brazil

\* Correspondence: ddantas@ufs.br

**Abstract:** The lymphocyte classification problem is usually solved by deep learning approaches based on convolutional neural networks with multiple layers. However, these techniques require specific hardware and long training times. This work proposes a lightweight image classification system capable of discriminating between healthy and cancerous lymphocytes of leukemia patients using image processing and feature-based machine learning techniques that require less training time and can run on a standard CPU. The features are composed of statistical, morphological, textural, frequency, and contour features extracted from each image and used to train a set of lightweight algorithms that classify the lymphocytes into malignant or healthy. After the training, these classifiers were combined into an ensemble classifier to improve the results. The proposed method has a lower computational cost than most deep learning approaches in learning time and neural network size. Our results contribute to the leukemia classification system, showing that high performance can be achieved by classifiers trained with a rich set of features. This study extends a previous work by combining simple classifiers into a single ensemble solution. With principal component analysis, it is possible to reduce the number of features used while maintaining a high accuracy.



**Citation:** Sant'Anna, Y.F.D.d.; Oliveira, J.E.M.d.; Dantas, D.O. Interpretable Lightweight Ensemble Classification of Normal versus Leukemic Cells. *Computers* **2022**, *11*, 125. <https://doi.org/10.3390/computers11080125>

Academic Editors: Antonio Celesti, Ivanoe De Falco, Antonino Galletta and Giovanna Sannino

Received: 30 June 2022

Accepted: 15 August 2022

Published: 19 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** leukemia classification; ensemble learning; machine learning; feature analysis

## 1. Introduction

Leukocytes are some of the types of cells that compose the human blood. Leukemia is a disease that affects the function and shape of leukocytes and can occur in a chronic or acute form. Acute leukemia is more aggressive, has more intense symptoms, and evolves quickly. Lymphocytes, a type of leukocyte, are called lymphoblasts in their immature form. Acute lymphoblastic leukemia (ALL) is a type of cancer characterized by the accumulation of lymphoblasts within the bone marrow. ALL is the most common childhood leukemia, mainly affecting children between 3 and 7 years old, and 75% of diagnoses occur before the age of 6 [1]. According to the Brazilian Institute of Cancer (INCA), in the year 2020, leukemia killed about 6738 people. An early and more accessible diagnosis could save many of these lives [2].

Usually, leukemia diagnoses are done by microscopic analyses of blood smears. The diagnosis depends on the hematologist's expertise in distinguishing malignant from healthy lymphocytes. Pattern recognition, combined with image processing techniques, has been used in blood analyses to produce computer-aided diagnosis (CADx) systems that aim to improve the lymphocyte classification performance [3,4].

The Acute Lymphoblastic Leukemia Image Database (ALL-IDB) [5,6] for image processing provides a set of annotated images that can be used in the evaluation of classifiers of ALL cells. This initiative provides two different datasets: ALL-IDB1, consisting of 108 blood smear pictures collected from healthy and leukemic patients containing 510 single lymphocytes; ALL-IDB2, a collection of cropped areas of interest of normal; and malignant lymphocytes that belong to the ALL-IDB1 dataset.

Many studies have assisted hematologists in analyzing blood smear images for ALL recognition. Some of these attempts have considered aspects of lymphocytes such as

color, and textural and morphological features. The ALL-IDB dataset was used to train classification models based on techniques such as support vector machines (SVM), k-nearest neighbors (KNN), random forests, and ensemble classifiers. More recent approaches use deep learning algorithms such as convolutional neural networks (CNN) to build models to solve this problem.

Putzu et al. [7] proposed a leukocyte classification method using image features such as color, texture, and shape. An SVM, a KNN, and decision tree models were trained using these features to classify leukocytes and detect malignant cells. The dataset used in his work was ALL-IDB1. Accuracy of 93.63% was achieved using an SVM with a radial basis function kernel to analyze 267 leukocytes of the dataset.

Mishra et al. [8] presented a CADx system for detecting leukemia using the features extracted from a discrete cosine transform (DCT) of grayscale lymphocyte images. This paper proposed using DCT values with an SVM for the lymphocyte classification. The ALL-IDB2 dataset was used in the k-fold cross-validation strategy to split the data into training and testing datasets. His system achieved an accuracy of 89.76%.

MoradiAmin et al. [9] presented a CADx to distinguish between healthy and diseased cells. The proposed system aggregates first and second-order statistical, morphological and geometric features extracted from the nucleus images. These feature sets are used to train an SVM using different kernels. After training, these classifiers are ensembled to build a single classifier based on majority voting. The research team used a private image dataset of 958 lymphocyte images (315 healthy and 643 malignant) divided into test and training datasets. Finally, the author used the k-fold cross-validation strategy to achieve an accuracy of 96.37%.

Shafique and Teshin [10] deployed an AlexNet, a pre-trained classifier, to achieve an accuracy of 99.5%. The authors used the ALL-IDB2 dataset and increased the number of images from 260 to 760 (500 malignant and 260 healthy) using mirroring and rotation operations. The augmented dataset was split into training and test sets, with 60% and 40% of the images respectively.

Moshvash et al. [4] developed a system that used a set of features composed of 32 textural, 15 shape, and six color descriptors. The texture data such as energy, correlation, homogeneity and contrast were extracted from a gray level co-occurrence matrix (GLCM). Each matrix component indicates the probability of two pixels having particular gray levels at a particular spatial relationship. These features were used to train naive Bayes (NB), KNN, decision tree and SVM models. Then these classifiers were combined into an ensemble classifier. The ALL-IDB1 dataset was used to achieve an accuracy of 98.10%.

Mourya et al. [11] used DCT features and a CNN to distinguish malignant and healthy cells in a hybrid classifier called Leukonet. To train this architecture, they developed a dataset with 9211 cancer cells from 65 subjects and 4528 healthy cells from 52 subjects; these images are separated into different folders. Different subjects were divided into training and validation sets. They achieved an accuracy of 89.70% and an F1-score of 91.95% for the cancer cell class.

Many works were developed using small datasets. The use of the ALL-IDB dataset, with only 510 lymphocyte samples, has been commonplace. There are many feature extraction-based approaches. However, none of these studies have combined these features in a single feature vector, and no methodology has ensembled a feature-based neural network with other classifier algorithms. CNNs or deep convolutional architectures are widely used and achieve high accuracy but require considerable amount of training time and more specific, restricted, and expensive hardware. According to Dongyu [12], a convolutional neural network training trial may take a couple of weeks. This time can amount to months of searching for different parameters.

Garcia et al. [13] showed the benefits of combining various classification models to improve the results in complex and imbalanced datasets. This combination of models improves the results as long as the results of individual models are not too close to perfection. Many works combine different algorithms to achieve a single ensemble solution. For breast

cancer detection, Abdar [14] proposed a CWV-BANN-SVM that combined two SVM and a boosting ANN to achieve an accuracy of 100%. Hsieh [15] developed, for the same problem, an algorithm that combines a neuro-fuzzy, KNN, and a quadratic classifier to obtain a 97.17% accuracy. Neuro-fuzzy refers to a combination of an artificial neural network and fuzzy logic. These studies presented ensemble classifiers with high-performance using lightweight classifiers. On the other hand, Moon et al. [16] presented a CADx system to classify breast ultrasound images using an ensemble of different deep CNN architectures, including VGGNet, ResNet and DenseNet, obtaining excellent results on different datasets.

Several studies use ensemble classifiers to solve the leukocyte classification problem and similar problems. However, many of these studies combined heavyweight classifiers [11,17,18], increasing the training time even more, and increasing the need for high-performance hardware. Approaches that combine lightweight algorithms on an ensemble classifier generally use the same algorithm with different configurations [4] or only classical classifiers [9] without exploring a lightweight artificial neural network (ANN) on the ensemble solution.

In a previous study [19], we proposed a lightweight neural network classifier to classify images of lymphocytes into malignant or healthy. The classifier used as input a feature vector with 108 low-order statistical, 20 morphological, 75 textural, 1024 DCT, and 160 contour features extracted from the lymphocyte images.

This study extended that previous work [19] by combining the neural network with three traditional lightweight classifiers—KNN, SVM, and NB—into an ensemble classifier to improve the classification results. The methodology achieved state-of-the-art performance using a fraction of the computational time cost necessary to train a convolutional neural network. Finally, a principal component analysis (PCA) was used to select the most important features, and to test the possibility of reducing the number of features required in order to reduce the classifier's complexity.

## 2. Materials and Methods

The proposed methodology comprises two steps: feature extraction and ensemble classifier.

In the feature extraction step, we extracted low-order statistical, morphological, DCT, and contour data from images of lymphocytes. These features were combined into a single feature vector, normalized, and used to train a set of classifiers. The feature extraction step is described in Section 3.

The ensemble classifier uses ANN, KNN, SVM, and NB classifiers. The ANN approach was selected from a combination of networks evaluated in a grid search. The best ANN solution was fine-tuned to obtain the final neural network. The KNN, SVM, and NB were trained with the same feature set as the ANN. Finally, we used a PCA to find the most relevant features to the classification process, as an attempt to obtain a light and interpretable model. With these features, a new and simpler classifier was obtained with performance similar to the complete one. The ensemble classifier step is described in Section 4.

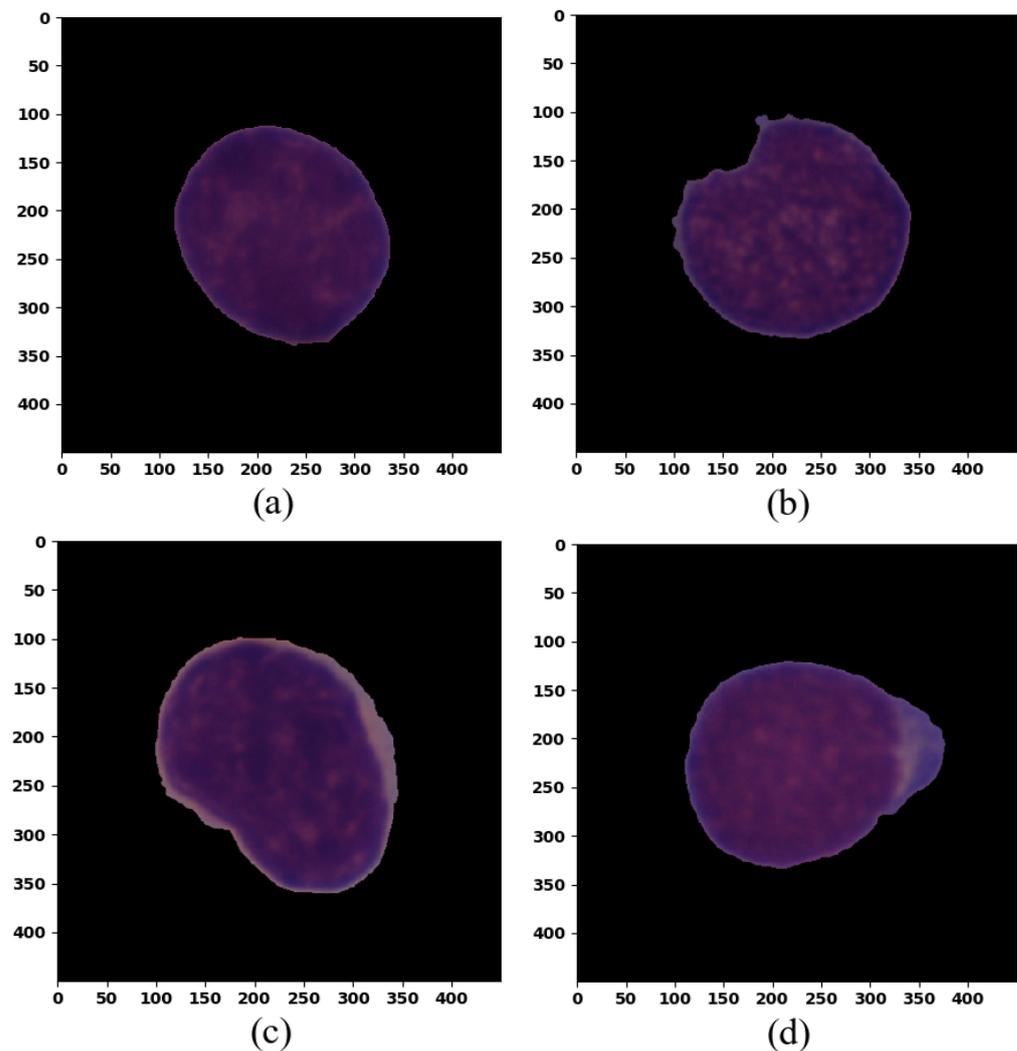
The implementation of this methodology is publicly available online at <https://github.com/yurifarod/ISBI-2019> (accessed on 8 June 2022). The code was developed in Python 3.7 using machine learning libraries such as Tensorflow, Keras, and Mlxtend; the image processing libraries OpenCV, PIL, and Pyradiomics [20]; and many other libraries responsible for data processing and manipulations, such as Numpy, Scipy, Pandas, CSV, Os, Multiprocessing, Queue, and Timeit.

The dataset used was the publicly available C-NMC 2019 dataset, which is described in the Section 2.1. We used a data augmentation strategy to balance the training and validation image sets, as described in Section 2.2.

### 2.1. C-NMC 2019 Dataset

The dataset used in this study was provided by the research team of SBILab [21]. The C-NMC 2019 dataset [22] consists of 15,114 images of lymphocytes collected from 118 subjects. These images were split into a training, preliminary, and test sets. Each image set contains single-cell images of healthy or malignant lymphocytes previously labeled by a team of oncologists.

The cells were dyed using the Jenner–Giemsa stain technique [23]. The SBILab team preprocessed these images using segmentation, image enhancement, and normalization techniques [24–26]. Individual lymphocytes were segmented from blood smear images and placed in the center of them; each picture has  $450 \times 450$  pixels and a black background. Figure 1 shows samples of both healthy and malignant cells from this dataset.



**Figure 1.** C-NMC 2019 dataset samples. The images (a,c) are malignant lymphocytes, and (b,d) are healthy lymphocytes. Reproduced with permission from Ref. [19]. 2021, IEEE.

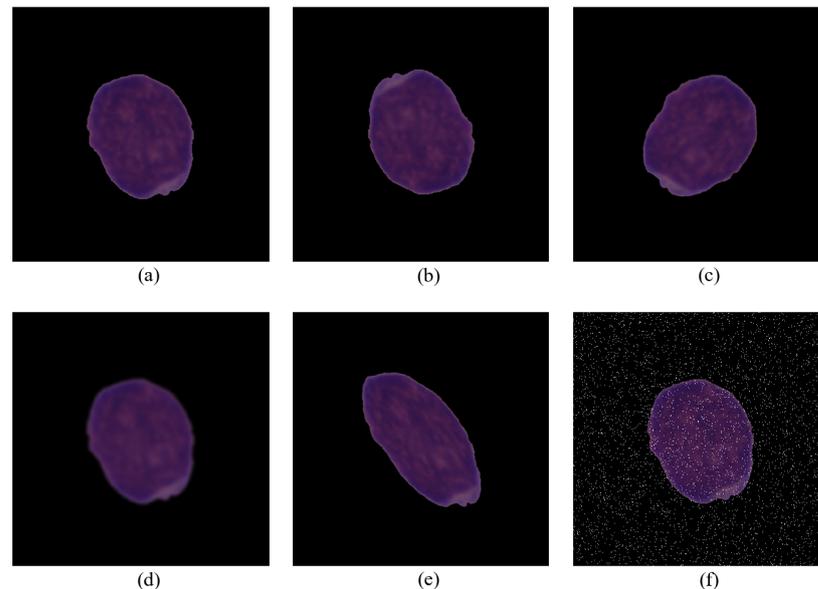
The data of the final test set were unlabeled and can be used to submit results to the website of the “C-NMC challenge: Classification of Normal versus Malignant Cells in B-ALL White Blood Cancer Microscopic Images” organized by the SBILab [27].

This work used the labeled data, i.e., train and preliminary sets, with 4037 healthy lymphocytes from 41 subjects and 8491 malignant lymphocytes from 60 patients. These images were split into training, validation, and test sets. The pictures of the same patient were placed in the same group, as done by Mourya et al. [11]. The subjects were divided into training–validation–test in the ratio of 7:2:1, as shown in Table 1.

## 2.2. Data Augmentation

Sometimes, a classifier may work very well on the training data while performing poorly on previously unseen data. When this happens, we say that the model does not generalize well; i.e., it is overfitted. When a model is complex to the point that it models noise in the training data instead of smooth decision surfaces, it is probably overfitted. The model probably memorized the samples present in the training set instead of learning to generalize from them. To avoid overfitting, we may use several regularization techniques, such as dropout layers in the ANN, lasso, and ridge regression. Another possibility is to use data augmentation. It consists of augmenting the dataset with new samples obtained from the original ones by adding noise or doing some transformation. With data augmentation, it is possible to increase the dataset's size when it is too small. It is also useful to balance the number of samples of the classes, as an unbalanced training set may generate a biased model [28].

Data augmentation was used to balance the training and validation sets and was not applied to the test images. New images were created using and combining rotation, blurring, mirroring, shearing transformation, and addition of salt-and-pepper noise. Examples of these images appear in Figure 2. Table 1 shows the sizes of the augmented sets.



**Figure 2.** Examples of augmented images: (a) source image; (b) vertical and horizontal mirroring; (c) 60° clockwise rotation; (d) Gaussian blur with  $17 \times 17$  kernel; (e) shear transformation with a factor of 0.3; and (f) salt-and-pepper noise. Reproduced with permission from Ref. [19]. 2021, IEEE.

**Table 1.** Numbers of samples in the training, validation, and test sets. The number of patients is shown in parenthesis. Reproduced with permission from Ref. [19]. 2021, IEEE.

	Original		Data-Augmentation	
	Malignant	Healthy	Malignant	Healthy
Training	5923 (42)	3035 (29)	20,000	20,000
Validation	1531 (12)	506 (8)	5000	5000
Test	1007 (6)	496 (4)	N/A	N/A

## 3. Feature Extraction

The first step of the proposed classification method is the feature extraction. From each image contained in the dataset, we extracted an array of 1387 features. The features used were combinations of several found in previous leukocyte classification studies, and were the same as the ones used in our previous study [19].

We used low-order statistical, textural, morphological, contour, and DCT features extracted from each lymphocyte image. Table 2 shows the number of features used of each type.

**Table 2.** Number of features of each type.

Feature Type	Number
Low-order statistical	108
Textural	75
Morphological	20
Contour	160
DCT	1024
Total	1387

We obtained the low-order statistics from each channel of the images in both RGB and HSV formats. These statistics provide information about the image histograms, such as energy, entropy, skewness, kurtosis, mean, and standard deviation, as defined by the Image Biomarker Standardisation Initiative (IBSI) [29].

The textural features were calculated using the coefficients of co-occurrence matrices. These coefficients represent the different gray level combinations that occur in the image and can be used in image classification tasks [9,30]. We used features obtained from the gray level co-occurrence matrix (GLCM), gray level run length matrix (GLRLM), gray level dependence matrix (GLDM), gray level size zone matrix (GLSZM), and neighboring gray-tone difference matrix (GLDM) [20,29].

The morphological features used—rectangularity, eccentricity, elongation, compactness, etc.—indicate the general shape of a lymphocyte, and also have been used in the cell classification task by other authors [7,9,31].

The contour features were obtained from the discrete Fourier transform of the centroid distance function (CDF) of the lymphocyte. The CDF represents the distance between the lymphocyte centroid and each pixel of its contour. This kind of shape signature was first proposed by Cosgriff [32] as a technique to identify objects and has been used to classify cells [33,34].

The discrete cosine transform (DCT), due to its energy compactness in the frequency domain, is widely used in image and video compression [35,36]. In this study, we calculated the DCT from the lymphocyte image converted to grayscale, producing a matrix with 202,500 DCT coefficients. The size of this matrix was the same as the number of pixels in each image ( $450 \times 450$ ). We mapped the coefficients to a 1D array using a zigzag scan and used only the first 1024 lowest frequency coefficients.

Finally, all the features were combined into a unique vector for each sample image to train the different classifiers. The features from all samples were combined into a matrix with one sample per row and one feature per column. Feature values were normalized by subtracting each value from the column's mean and dividing it by the column's standard deviation.

#### 4. Ensemble Classifier

The study of Garcia et al. [13] showed that it is possible to achieve high-performance results by combining different lightweight classifiers into a single solution. These classifiers obtain better results if trained with diverse data and applied to a complex and unbalanced problem. The simple vote scheme is a light and fast method to combine these classifiers. In this type of ensemble solution, it is possible to train all the classifiers with the same data in parallel, saving processing time. The ensemble classification result of a certain input is given by the class with the most votes from the different classifiers [15]. A criterion could be determined to be used in case of ties.

In this study, we combined an NB, a KNN, an SVM, and an ANN into four different ensemble classifiers, each one composed by three classifiers. Our classification problem has only two classes, so an odd number of classification models ensured that there would be no ties.

#### 4.1. Naive Bayes Classifier

The naive Bayes classifier is one of the simplest and most widely used algorithms of pattern recognition. It is a probabilistic approach that calculates, for each possible class, the probability of an object belonging to it. The classification result is the class with the highest probability [37].

This algorithm is based on the Thomas Bayes decision theory, which assumes that the decision problem is posed in probabilistic terms and that all relevant probability values are known [38]. In simpler words, the Bayesian classifier maps decision boundaries based on the information given by labeled data and calculates the probabilities of new objects being allocated into a certain class. In this study, we used a Gaussian classifier with no prior class and a smoothing value of  $10^{-9}$ .

#### 4.2. K-Nearest Neighbor

Proposed in 1951, the KNN is another machine learning algorithm used in many works of supervised classification problems. This method has a simple logical structure and classifies a given object as the most frequent occurring class in its neighborhood [38].

In other words, the KNN determines the class of a sample by finding the most frequent class among the  $K$  nearest objects to the sample. These neighbors are the ones used in the training step, and are already labeled. In the particular case where  $K = 1$ , the KNN is equivalent to the nearest neighbor algorithm, and the chosen class is defined by the neighbor closest to the sample to be classified [37].

#### 4.3. Support Vector Machine

The third lightweight classifier trained from the feature vector was the SVM classifier. The central idea of this algorithm is to obtain hyperplanes that separate the samples used for training into their respective classes [38].

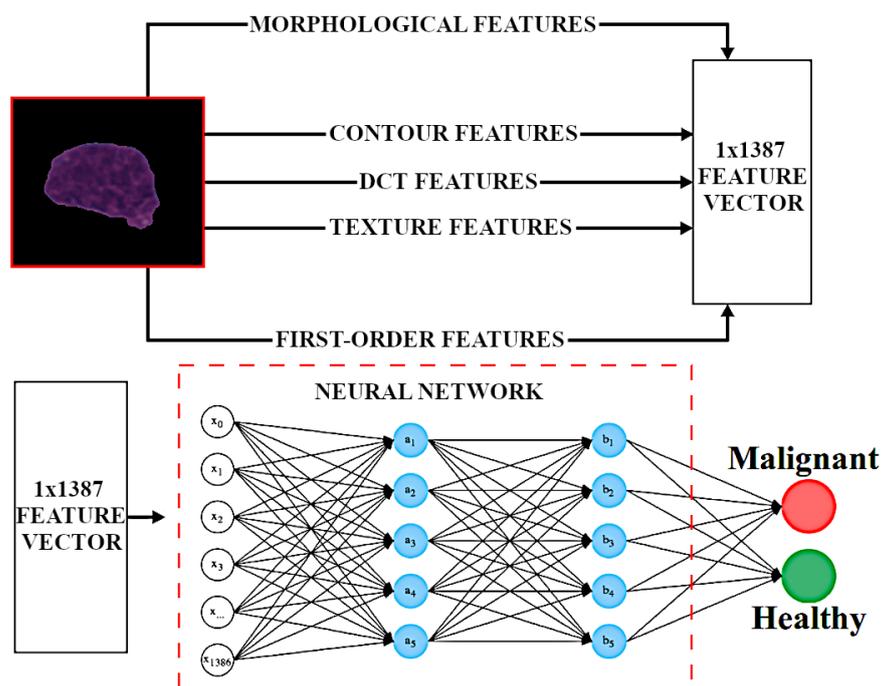
The points closest to the discrimination hyperplane are called support vector points, and the distances between these points and a hyperplane are called margins. The support vector machine technique searches for a separation hyperplane that maximizes the margins [37]. In this study, we used a canonical SVM classifier with default parameters and linear discrimination.

#### 4.4. Neural Network Training and Fine-Tuning

The previously extracted feature matrix was fed into an ANN that discriminates the lymphocytes as either malignant or healthy. The classification scheme is represented by Figure 3.

To find the best architecture to solve our problem, we did an extensive search in the hyper-parameter space of our network. Table 3 shows all the evaluated values and the best parameters found. The grid search executed 25 epochs for each data point to finish the process timely.

After finding the best architecture among all ANN possibilities, a fine-tuning step was implemented to obtain, among other values, the best number of epochs. Since the best optimization method was the Adam function, it was essential to choose the best values for the learning rate,  $\beta_1$ , and  $\beta_2$ . These coefficients are responsible for controlling the exponential decay rates of the moving averages [39]. The values tested were 0.01, 0.001, 0.005, 0.0001, 0.0005 for the learning rate. The values tested for  $\beta_1$  and  $\beta_2$  were 0.99, 0.98, and 0.97. The best value found for  $\beta_1$  and  $\beta_2$  was 0.97, and for the learning rate the best was 0.001.



**Figure 3.** Artificial neural network scheme. Reproduced with permission from Ref. [19]. 2021, IEEE.

**Table 3.** Grid-search execution. Reproduced with permission from Ref. [19]. 2021, IEEE.

Parameter	Values	Chosen Value
Hidden Layers	1, 2, 3, 4	1
Batch Size	250, 750, 1000, 1500	250
Dropout	0.1, 0.25, 0.3, 0.5	0.1
Neurons Number	1024, 1536, 2048, 2560	2560
Activation	Prelu, Relu, Sigmoid, Softmax	Relu
Optimizer	Adamax, Adam, SGD	Adam
Kernel_INITIALIZER	Random Uniforme, Normal	Normal

Finally, we searched for the minimum number of epochs necessary to maximize the F1-score. A number of epochs that is too big could cause excessive specialization on the training dataset. This condition leads to an incapacity in generalizing and errors when evaluating new images. This phenomenon is called overfitting [38]. The number of epochs started at 50, and at each iteration increased by 50 until the F1-score remained stable. The test showed that 150 is the best number of epochs for this ANN, and after this threshold, additional training could cause overfitting.

#### 4.5. Ensemble Learning

According to Dietterich et al. [40], ensemble learning algorithms differ from other approaches because they do not use a single model to explain the data. Instead, they construct a set of classifiers and combine them in some fashion to classify new data points. As previously mentioned, other ensemble classifiers were proposed to solve the C-NMC problem, but none of them combined a neural network with other lightweight models [4,9] or with dense convolutional networks [11,17,18].

The literature presents several ways to combine a classifier into a single solution. It is possible to combine the results of the classifiers, use them as input to a new classification algorithm, use a function with different weights for each classifier [41], or use more sophisticated approaches, such as alpha-integration [42].

In this study, we trained—using the same vector of features—three of the most notorious and simple classifiers, a Gaussian NB with no prior class, a KNN, and a linear SVM classifier. These three classifiers plus the ANN were combined into four different ensemble classifiers, each one composed by three primitive ones, using a simple voting scheme. This procedure ensures the absence of ties and provides a fast and light solution without giving preference or a bigger weight to a particular classifier. The four ensemble classifiers created were

- ANN + SVM + NB (full ensemble model, with best F1-score).
- ANN + SVM + KNN.
- ANN + KNN + NB.
- SVM + KNN + NB.

Each primitive model used in the ensemble was trained separately, as described in the previous sections, without further optimization or fine-tuning. The result of the ensemble is the class with the most votes from each primitive model after their individual decision, a method known as late hard fusion [43].

#### 4.6. Principal Component Analysis and Interpretable Models

The study of Ruding [44] explains the importance of using interpretable models instead the black-box trained machine learning to better understand the relevant features and to study which characteristics are decisive for the discrimination of classes. This practice can generate better and more applicable solutions to real world scenarios.

As an attempt to find the most important features for the classification task at hand, we started by choosing the full ensemble model, i.e., the ensemble classifier with best F1-score, composed by the ANN, SVM, and NB.

We used the Mlxtend feature selection Python library to find the principal components of our best ANN configuration. The Mlxtend sequential feature selection (SFS) removes one of the features at each iteration, returning a list of features and a score obtained with them. This process was interrupted when we observed the reduction in the quality metric. The minimum number of features selected to minimize the loss of precision of the ANN had about 15% of the total number of features.

Afterwards, we used the Keras features selector to find the 15% most relevant features for the SVM and NB classifiers [45]. The final reduced list of features was obtained from the union of the reduced lists of the individual classifiers. This procedure returned an array with 268 features, as shown in Table 4. These features are listed with the source code in descending order of importance at [https://github.com/yurifarod/ISBI-2019/blob/main/z\\_interpretable\\_ensemble\\_analysis.txt](https://github.com/yurifarod/ISBI-2019/blob/main/z_interpretable_ensemble_analysis.txt) (accessed on 8 June 2022). We used this reduced feature array to train a new, reduced version of our the full ensemble classifier:

- PCA, ANN + SVM + NB (reduced ensemble model).

The new reduced model is a lighter, interpretable, and faster solution. The hyperparameters used to train the ANN with the reduced parameter set were the same as the ones used for the full-set training.

**Table 4.** Number of features of each type in the reduced ensemble model.

Feature Type	Number
Low-order statistical	33
Textural	45
Morphological	4
Contour	4
DCT	182
Total	268

## 5. Results and Discussion

As the C-NCM 2019 dataset was unbalanced, the obtained accuracy may not reflect reality. F1-score was chosen as the metric to overcome this problem. The metric also allows the comparison with other studies, as the teams who participated in SBILab's challenge also used the F1-score. Gupta [46] edited a book where the participants' results can be found.

The optimal hyper-parameters, i.e., the number of hidden layers, the number of neurons per layer, and the optimizer function, were chosen based on the performances of several ANNs generated by combining the parameters presented in Table 3. The table also shows the optimal parameters.

The experiments were done on an Intel Core i7-7500U CPU @ 2.7 GHz  $\times$  4, with 32 GB of RAM, without a dedicated GPU, as we aimed to evaluate a low-cost setup.

The published F1-scores obtained from the preliminary test set in SBILab's challenge are shown in Table 5. Compared to these results, the best ensemble classifier trained using the proposed feature extraction method achieved the the highest F1-score. The ANN and two other ensemble classifiers had among the five best F1-scores. Notice that these approaches use less computational power than all convolutional neural network approaches submitted to this challenge. Another interesting point is the high performance achieved by the reduced ensemble model, which evidences its viability.

The time spent in training is omitted in most papers, but it is possible to compare the sizes of the neural networks. The number of parameters in our biggest model is smaller than those in most convolutional networks submitted to the SBILab challenge, and the reduced model is significantly smaller than all approaches. The proposed approach could potentially become a portable solution, and even be used in a low-cost device, such as a smartphone. Table 6 indicates the network sizes as computed by Tensorflow.

**Table 5.** Performance of participants in the C-NMC challenge hosted by the SBILab. Proposed classifiers in boldface.

SBILab Challenger	F1-Score	Methodology
<b>ANN + SVM + NB</b>	<b>93.70%</b>	<b>Feature extraction and ensemble classifier</b>
[47]	92.50%	Transfer learning with ResNets
[48]	91.70%	Transfer learning with VGG16
<b>ANN + SVM + KNN</b>	<b>91.80%</b>	<b>Feature extraction and ensemble classifier</b>
<b>ANN</b>	<b>91.20%</b>	<b>Feature extraction and ANN</b>
<b>ANN + KNN + NB</b>	<b>90.60%</b>	<b>Feature extraction and ensemble classifier</b>
[17]	90.30%	Deep multi-model ensemble network
<b>PCA, ANN + SVM + NB</b>	<b>89.87%</b>	<b>Reduced feature vector and ensemble classifier</b>
[49]	89.47%	Transfer learning with MobileNetV2
[50]	87.89%	ResNeXt50
<b>SVM + KNN + NB</b>	<b>87.60%</b>	<b>Feature extraction and ensemble classifier</b>
[51]	87.58%	Transfer learning with CNN and recurrent ANN
[23]	87.46%	Transfer learning with ResNet18
[52]	86.74%	InceptionV3 + DenseNet + InceptionResNetV2
[53]	85.70%	ResNeXt50 + ResNeXt101
[18]	84.00%	Transfer learning with Inception + ResNets
[54]	81.79%	Transfer learning with ResNets + SENets
<b>SVM</b>	<b>79.53%</b>	<b>Feature extraction and SVM</b>
<b>KNN</b>	<b>76.66%</b>	<b>Feature extraction and KNN</b>
<b>NB</b>	<b>74.25%</b>	<b>Feature extraction and NB</b>

**Table 6.** Size comparison with other network architectures. Proposed classifiers in boldface. Adapted from Ref. [19]. 2021, IEEE.

Network	Number of Parameters
VGG16	138,357,544
ResNet152	60,380,648
InceptionResNetV2	55,873,736
ResNet50	25,636,712
Xception	22,910,480
DenseNet201	20,242,984
<b>ANN and full ensemble</b>	<b>9,177,601</b>
DenseNet121	8,062,504
<b>Reduced ensemble model</b>	<b>2,775,553</b>

In comparison with a CNN approach [55], using the same computational setup, all of our ensemble classifiers had similar F1-scores. The best ensemble learning model showed a better F1-score, and far smaller computational time and network size. Table 7 shows the results of this comparison.

**Table 7.** VGG16 Comparison. Adapted from Ref. [19]. 2021, IEEE.

Metric	Reduced Ensemble	ANN	Full Ensemble	VGG16
Feature extraction time	16 min	1 h 2 min	1 h 2 min	-
Training time	8 min	9 min	9 min	16 h 20 min
Number of parameters	2,775,553	9,177,601	9,177,601	66,358,593
F1-score	89.87%	91.20%	93.70%	92.60%

For the malignant class, our best model achieved an F1-score of 93.70%. This result was obtained by combining an NB, an SVM, and a neural network with three layers (one input, one hidden, and one output). The output layer consisted of one neuron and used a sigmoid activation function. The hidden layers contained 2560 neurons each and used the ReLU activation function. The ANN was trained from scratch using an Adam optimizer, a learning rate of 0.001 over 150 epochs, and 0.97 for  $\beta_1$  and  $\beta_2$ .

The training of all approaches can be executed in parallel on a multi-core CPU; it took a maximum of 9 min in the CPU used in the experiments. Each sample consisted of an array with size  $1 \times 1387$  for the complete classifiers and size  $1 \times 268$  for the reduced approach.

Table 5 shows us that a reduced version of our best ensemble learning classifier (ANN + SVM + NB), although not as good as the version trained with all the features, can achieve a high performance. The F1-score is 4% smaller than our best result, with faster training and a lighter structure than the version trained with all 1387 features. Table 8 shows the quality metrics obtained with our best ensemble model, its reduced version, and the ANN.

**Table 8.** Metric comparison between the full ensemble, reduced ensemble, and ANN.

Metric	Reduced Ensemble	ANN	Full Ensemble
F1-Score	89.87%	91.20%	93.70%
Accuracy	83.19%	86.82%	88.13%
Sensitivity	86.60%	88.11%	95.47%
AUC	75.10%	84.68%	88.36%
Kappa	44.47%	56.45%	67.79%
Precision	94.36%	96.48%	97.32%
Specificity	65.43%	77.92%	80.25%

We evaluated the variability of the best methods by doing a Monte Carlo experiment with 100 repetitions. The training and validation sets were mixed together. At each

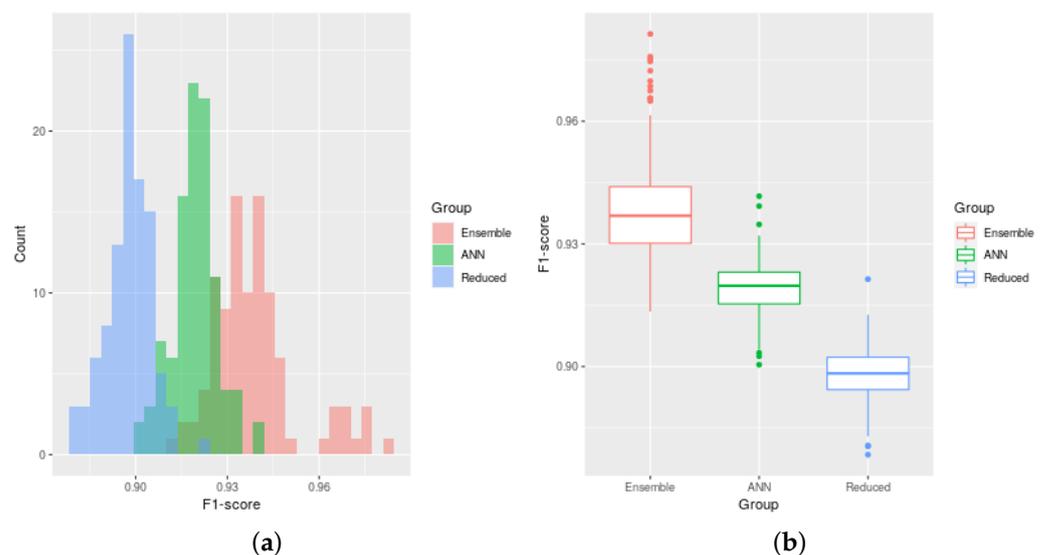
step, new random training and test sets were obtained and used to train and evaluate new classifiers.

Table 9 shows the results obtained in the Monte Carlo experiment with the full ensemble classifier (ANN + NB + SVC), ANN, and reduced ensemble classifier. A nonparametric Mann–Whitney U-test was used to verify if the ensemble model is significantly better than the ANN. We can see that the full ensemble model always achieved better metrics than the ANN. Besides that, considering a significance level of 0.01, the  $p$ -values obtained were all smaller than the threshold for all metrics, which confirms that the improvements in the metrics from the usage of the full ensemble model were statistically significant over using simply the ANN.

**Table 9.** Metrics obtained with the Monte Carlo experiment.

Metric	Reduced Ensemble Mean (SD)	Reduced vs. ANN $p$ -Value	ANN Mean (SD)	ANN vs. Full $p$ -Value	Full Ensemble Mean (SD)
F1-Score	89.78% (0.74%)	$1.6 \times 10^{-31}$	91.94% (0.75%)	$1.0 \times 10^{-24}$	93.88% (1.41%)
Accuracy	83.05% (1.16%)	$6.7 \times 10^{-32}$	86.54% (1.19%)	$1.1 \times 10^{-24}$	89.67% (2.32%)
Sensitivity	85.24% (1.13%)	$9.8 \times 10^{-28}$	87.89% (1.18%)	$9.2 \times 10^{-25}$	90.89% (2.21%)
AUC	76.62% (1.99%)	$1.3 \times 10^{-32}$	82.57% (1.94%)	$3.2 \times 10^{-17}$	86.12% (3.02%)
Kappa	40.99% (3.40%)	$1.2 \times 10^{-32}$	51.79% (3.53%)	$2.2 \times 10^{-23}$	61.08% (7.64%)
Precision	94.85% (0.58%)	$2.6 \times 10^{-34}$	96.38% (0.56%)	$2.6 \times 10^{-34}$	97.09% (0.72%)
Specificity	68.01% (3.57%)	$2.3 \times 10^{-31}$	77.25% (3.48%)	$2.1 \times 10^{-10}$	81.35% (4.41%)

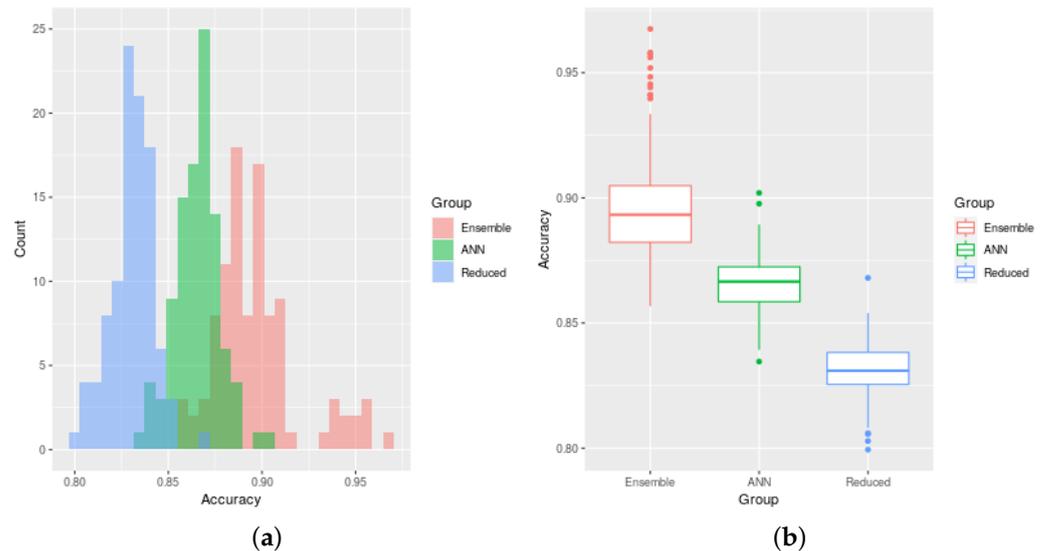
Figures 4 and 5 show, respectively, the F1-scores and accuracies obtained in the Monte Carlo experiment. The figures show histograms and boxplots of the metrics obtained.



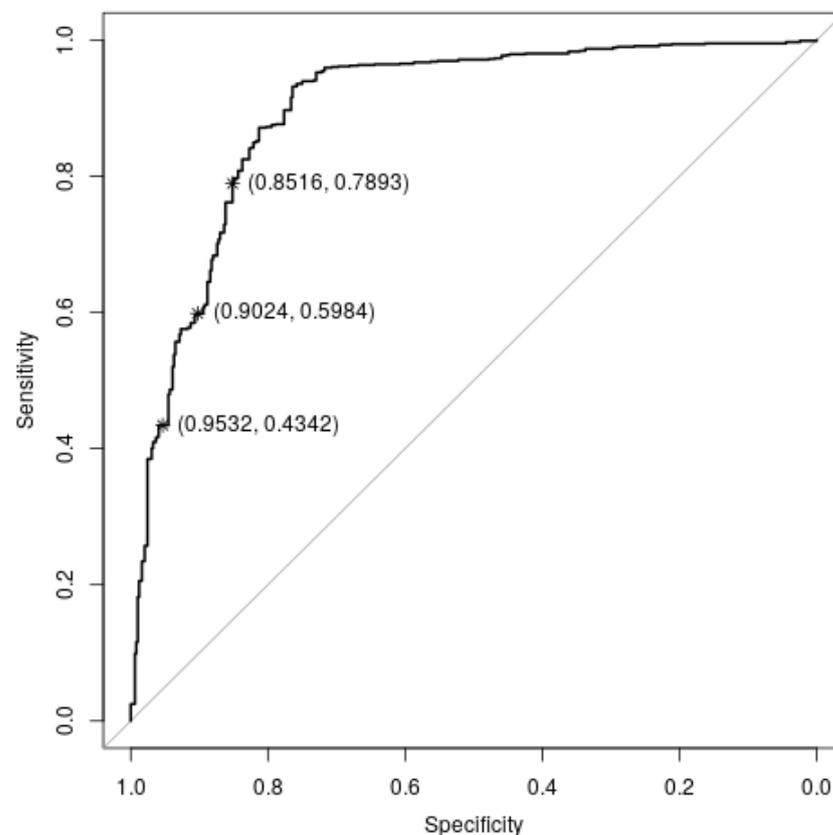
**Figure 4.** Histograms (a) and boxplots (b) of the F1-scores obtained in the Monte Carlo experiment with the full ensemble, ANN, and reduced ensemble models.

Figure 6 shows the receiver operating characteristic (ROC) curve of the full ensemble classifier. The proposed test, when set to a specificity of 90%, has a sensitivity of 60%. Diminishing the specificity to 85% raises the sensitivity to 79%. Depending on the usage, we may want a test that is more sensitive or more specific. Consider the situation where a patient is under treatment, and we want to know whether the treatment must proceed or stop. A highly specific test has a low probability of classifying a healthy patient as diseased, which avoids unnecessary procedures, is often invasive, is costly, and is stressful [56]. Thus, in case of a positive result, we may assume that the treatment must proceed. On the other

hand, a highly specific test may have low sensitivity or a high probability of classifying a diseased patient as healthy, i.e., a false negative. In the case of a negative result, as stopping the treatment of a diseased patient may cause grave risks, a new test may be done, such as a bone marrow aspirate (BMA), which is more invasive, but has a high performance.



**Figure 5.** Histograms (a) and boxplots (b) of the accuracies obtained in the Monte Carlo experiment with the full ensemble, ANN, and reduced ensemble models.



**Figure 6.** ROC curve of the full ensemble classifier. The highlighted points are approximately at the specificity levels of 0.95, 0.90 and 0.85. test.

## 6. Conclusions and Future Work

In this work, we demonstrated that a set of lightweight classifiers combined with a multilayer neural network, associated with a standard image processing feature extraction procedure, works as well as deep convolutional learning models. Our results indicate that the proposed methodology can accurately classify the lymphocytes as healthy or malignant. The rich textural, frequency-domain, and statistical data used by our method can be applied to many other problems besides cell classification. Our study of the PCA gives us a way to select the most relevant features for the classification problem, achieving good performance with a relatively small number of parameters and a short training time.

State-of-the-art techniques typically use deep convolutional neural networks, which may require long training, depending on the computer used. The performances of all proposed methods are comparable to the best approaches in the literature, yet they require a few minutes to train and seconds to run on a simple Core i7 CPU.

Many studies only tested their methodologies on a few sample images or their private datasets. On the other hand, our study was done with a large and public set of images, making our results more general and easily replicated. It must also be noted that images from different patients were presented from the training and test datasets. This procedure emulates a real-life scenario well [11].

Although BMA is the gold standard for leukemia diagnosis, it is an invasive procedure done under anesthesia. Exams done with peripheral blood are less invasive and may sometimes be preferred, even being less accurate than BMA. Recent studies reported very good results obtained with peripheral blood flow cytometry (PBFC) [57]. Lam et al. reported a sensitivity of 99.7% and a specificity of 98.5% obtained with PBFC [58]. A disadvantage of flow cytometry is the requirement of marker reagents that may not be readily available in all laboratories, especially in third world countries [59], so blood smear image analysis may be an alternative. An F1-score of 93.70% is not accurate enough for disease diagnosis but can serve as a tool for assisting oncologists.

Future works may refine our methodology by focusing on adding features similar to the 268 best features chosen for our interpretable model, especially textural and low-order statistical features. We may also test the inclusion of other classification models, such as decision trees, and linear and quadratic discrimination analysis. We may test our approach in a more complex dataset and try to solve a multi-class classification problem. A possible way to improve our results is to use the scores returned by the classifiers before the decision, i.e., late soft fusion, and to integrate them in different ways, such as using their averages. It is also possible to use more sophisticated techniques, such as alpha-integration, and to optimize the weight of each primitive model in order to minimize the least mean squared error (LMSE) or the minimum probability of error (MPE) [43,60]. The alpha-integration uses a family of alpha-means that generalizes many widely used means, e.g., arithmetic, geometric, and harmonic. The alpha is a continuous value that defines which kind of mean is used and can be used to minimize the alpha-divergence of the distributions of the classifier's results, improving the performance of the ensemble [61].

**Author Contributions:** Conceptualization, D.O.D. and Y.F.D.d.S.; methodology, Y.F.D.d.S., J.E.M.d.O. and D.O.D.; software, Y.F.D.d.S., J.E.M.d.O. and D.O.D.; validation, Y.F.D.d.S.; formal analysis, D.O.D. and Y.F.D.d.S.; investigation, Y.F.D.d.S.; resources, Y.F.D.d.S.; data curation, Y.F.D.d.S.; writing—original draft preparation, Y.F.D.d.S. and J.E.M.d.O.; writing—review and editing, Y.F.D.d.S. and D.O.D.; visualization, Y.F.D.d.S. and D.O.D.; supervision, D.O.D.; project administration, D.O.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Source code is available at <https://github.com/yurifarod/ISBI-2019> (accessed on 8 June 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hoffbrand, A.V.; Moss, P.A.H. *Essential Haematology*, 6th ed.; Wiley: Hoboken, NJ, USA, 2013.
2. Instituto Nacional do Câncer. Tipos de Câncer: Leucemia. 2022. Available online: <https://www.inca.gov.br/tipos-de-cancer/leucemia> (accessed on 8 June 2022).
3. Mishra, S.; Majhi, B.; Sa, P.K. Texture feature based classification on microscopic blood smear for acute lymphoblastic leukemia detection. *Biomed. Signal Process. Control* **2019**, *47*, 303–311. [[CrossRef](#)]
4. Moshavash, Z.; Danyali, H.; Helfroush, M.S. An Automatic and Robust Decision Support System for Accurate Acute Leukemia Diagnosis from Blood Microscopic Images. *J. Digit. Imaging* **2018**, *31*, 702–717. [[CrossRef](#)] [[PubMed](#)]
5. Labati, R.D.; Piuri, V.; Scotti, F. ALL-IDB: The Acute Lymphoblastic Leukemia Image Database for Image Processing. In Proceedings of the 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, 11–14 September 2011; pp. 2045–2048.
6. DI-UNIMI. ALL-IDB: Acute Lymphoblastic Leukemia Image Database for Image Processing. 2020. Available online: <https://homes.di.unimi.it/scotti/all/> (accessed on 8 June 2022).
7. Putzu, L.; Caocci, G.; Ruberto, C.D. Leucocyte classification for leukaemia detection using image processing techniques. *Artif. Intell. Med.* **2014**, *62*, 179–191. [[CrossRef](#)] [[PubMed](#)]
8. Mishra, S.; Sharma, L.; Majhi, B.; Sa, P.K. Microscopic Image Classification Using DCT for the Detection of Acute Lymphoblastic Leukemia (ALL). In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2016; pp. 171–180.
9. MoradiAmin, M.; Memari, A.; Samadzadehaghdam, N.; Kermani, S.; Talebi, A. Computer aided detection and classification of acute lymphoblastic leukemia cell subtypes based on microscopic image analysis. *Microsc. Res. Tech.* **2016**, *79*, 908–916. [[CrossRef](#)] [[PubMed](#)]
10. Shafique, S.; Tehsin, S. Acute Lymphoblastic Leukemia Detection and Classification of Its Subtypes Using Pretrained Deep Convolutional Neural Networks. *Technol. Cancer Res. Treat.* **2018**, *17*, 1533033818802789. [[CrossRef](#)] [[PubMed](#)]
11. Mourya, S.; Kant, S.; Kumar, P.; Gupta, A.; Gupta, R. LeukoNet: DCT-based CNN architecture for the classification of normal versus Leukemic blasts in B-ALL Cancer. *arXiv* **2018**, arXiv:1810.07961.
12. Liu, D.; Cui, W.; Jin, K.; Guo, Y.; Qu, H. DeepTracker: Visualizing the Training Process of Convolutional Neural Networks. *ACM Trans. Intell. Syst. Technol.* **2018**, *10*, 6. [[CrossRef](#)]
13. Garcia, N.F.; Tiggeman, F.; Borges, E.N.; Lucca, G.; Santos, H.; Dimuro, G. Exploring the relationships between data complexity and classification diversity in ensembles. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS), Online, 26–28 April 2021.
14. Abdar, M.; Makarenkov, V. CWV-BANN-SVM ensemble learning classifier for an accurate diagnosis of breast cancer. *Measurement* **2019**, *146*, 557–570. [[CrossRef](#)]
15. Hsieh, S.L.; Hsieh, S.H.; Cheng, P.H.; Chen, C.H.; Hsu, K.P.; Lee, I.S.; Wang, Z.; Lai, F. Design ensemble machine learning model for breast cancer diagnosis. *J. Med. Syst.* **2011**, *36*, 2841–2847. [[CrossRef](#)]
16. Moon, W.K.; Lee, Y.W.; Ke, H.H.; Lee, S.H.; Huang, C.S.; Chang, R.F. Computer-aided diagnosis of breast ultrasound images using ensemble learning from convolutional neural networks. *Comput. Methods Programs Biomed.* **2020**, *190*, 105361. [[CrossRef](#)]
17. Xiao, F.; Kuang, R.; Ou, Z.; Xiong, B. DeepMEN: Multi-model Ensemble Network for B-Lymphoblast Cell Classification. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 83–93.
18. Liu, Y.; Long, F. Acute Lymphoblastic Leukemia Cells Image Analysis with Deep Bagging Ensemble Learning. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 113–121.
19. Sant’Anna, Y.F.D.; Oliveira, J.E.M.; Dantas, D.O. Lightweight Classification of Normal Versus Leukemic Cells Using Feature Extraction. In Proceedings of the 2021 IEEE Symposium on Computers and Communications (ISCC), Athens, Greece, 5–8 September 2021; pp. 1–7. [[CrossRef](#)]
20. van Griethuysen, J.J.M.; Fedorov, A.; Parmar, C.; Hosny, A.; Aucoin, N.; Narayan, V.; Beets-Tan, R.G.H.; Fillion-Robin, J.C.; Pieper, S.; Aerts, H.J. Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Res.* **2017**, *77*, e104–e107. [[CrossRef](#)]
21. SBILab. *Signal Processing and Biomedical Imaging Lab*. 2022. Available online: <http://sbilab.iiitd.edu.in/> (accessed on 8 June 2022).
22. Mourya, S.; Kant, S.; Kumar, P.; Gupta, A.; Gupta, R. ALL Challenge Dataset of ISBI. 2019. Available online: <https://wiki.cancerimagingarchive.net/x/zwYLAw> (accessed on 8 June 2022).
23. Marzahl, C.; Aubreville, M.; Voigt, J.; Maier, A. Classification of Leukemic B-Lymphoblast Cells from Blood Smear Microscopic Images with an Attention-Based Deep Learning Method and Advanced Augmentation Techniques. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 13–22.
24. Gupta, R.; Mallick, P.; Duggal, R.; Gupta, A.; Sharma, O. Stain Color Normalization and Segmentation of Plasma Cells in Microscopic Images as a Prelude to Development of Computer Assisted Automated Disease Diagnostic Tool in Multiple Myeloma. *Clin. Lymphoma Myeloma Leuk.* **2017**, *17*, e99. [[CrossRef](#)]
25. Duggal, R.; Gupta, A.; Gupta, R.; Wadhwa, M.; Ahuja, C. Overlapping cell nuclei segmentation in microscopic images using deep belief networks. In Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), Guwahati, India, 18–22 December 2016; ACM Press: New York, NY, USA, 2016.

26. Duggal, R.; Gupta, A.; Gupta, R.; Mallick, P. SD-Layer: Stain Deconvolutional Layer for CNNs in Medical Microscopic Imaging. In Proceedings of the Medical Image Computing and Computer Assisted Intervention (MICCAI), Quebec City, QC, Canada, 11–13 September 2017; Springer: Cham, Switzerland, 2017; pp. 435–443.
27. SBILab. Classification of Normal vs Malignant Cells in B-ALL White Blood Cancer Microscopic Images: ISBI 2019. 2019. Available online: <https://competitions.codalab.org/competitions/20395> (accessed on 8 June 2022).
28. Jacobusse, G.; Veenman, C. On Selection Bias with Imbalanced Classes. In Proceedings of the International Conference on Discovery Science, Kyoto, Japan, 15–17 October 2016; pp. 325–340. [CrossRef]
29. Zwanenburg, A.; Leger, S.; Vallières, M.; Löck, S. Image Biomarker Standardisation Initiative. *arXiv* **2016**, arXiv:1612.07003.
30. Aggarwal, N.; Agrawal, R.K. First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images. *J. Signal Inf. Process.* **2012**, *3*, 146–153. [CrossRef]
31. Houbay, E.M.F.E. Framework of Computer Aided Diagnosis Systems for Cancer Classification Based on Medical Images. *J. Med. Syst.* **2018**, *42*, 157. [CrossRef]
32. Cosgriff, R.L. *Identification of Shape*; Technical Report, Report 820-11; Ohio State University Research Foundation: Columbus, OH, USA, 1960.
33. Alhilal, M.S.; Soudani, A.; Al-Dhelaan, A. Image-Based Object Identification for Efficient Event-Driven Sensing in Wireless Multimedia Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 850–869. [CrossRef]
34. Klinzmann, A.; Bhonsle, S. *Centroid Distance Function and the Fourier Descriptor with Applications to Cancer Cell Clustering*; Technical Report; UCI Department of Mathematics: Irvine, CA, USA, 2011.
35. Wu, Y.G. Medical image compression by sampling DCT coefficients. *IEEE Trans. Inf. Technol. Biomed.* **2002**, *6*, 86–94.
36. Vishwakarma, V.P.; Pandey, S.; Gupta, M. A Novel Approach for Face Recognition Using DCT Coefficients Re-scaling for Illumination Normalization. In Proceedings of the 15th International Conference on Advanced Computing and Communications (ADCOM), Guwahati, India, 18–21 December 2007; pp. 535–539.
37. Kubat, M. *An Introduction to Machine Learning*, 2nd ed.; Springer International Publishing: Cham, Switzerland, 2017.
38. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*, 2nd ed.; Wiley: New York, NY, USA, 2001.
39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
40. Dietterich, T.G. *Ensemble Learning*, 6th ed.; The MIT Press: Cambridge, MA, USA, 2002.
41. Mohandes, M.; Deriche, M.; Aliyu, S.O. Classifiers Combination Techniques: A Comprehensive Review. *IEEE Access* **2018**, *6*, 19626–19639. [CrossRef]
42. Soriano, A.; Vergara, L.; Ahmed, B.; Salazar, A. Fusion of Scores in a Detection Context Based on Alpha Integration. *Neural Comput.* **2015**, *27*, 1983–2010. [CrossRef]
43. Safont, G.; Salazar, A.; Vergara, L. Vector score alpha integration for classifier late fusion. *Pattern Recognit. Lett.* **2020**, *136*, 48–55. [CrossRef]
44. Ruding, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [CrossRef]
45. Raschka, S. SequentialFeatureSelector: The Popular Forward and Backward Feature Selection Approaches Incl. Floating Variants. 2020. Available online: [http://rasbt.github.io/mlxtend/user\\_guide/feature\\_selection/SequentialFeatureSelector/](http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/) (accessed on 8 June 2022).
46. Gupta, A.; Gupta, R. (Eds.) *ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging*; Springer: Singapore, 2019.
47. Pan, Y.; Liu, M.; Xia, Y.; Shen, D. Neighborhood-Correction Algorithm for Classification of Normal and Malignant Cells. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 73–82.
48. Honnalgere, A.; Nayak, G. Classification of Normal Versus Malignant Cells in B-ALL White Blood Cancer Microscopic Images. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 1–12. [CrossRef]
49. Verma, E.; Singh, V. ISBI Challenge 2019: Convolution Neural Networks for B-ALL Cell Classification. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 131–139.
50. Prellberg, J.; Kramer, O. Acute Lymphoblastic Leukemia Classification from Microscopic Images Using Convolutional Neural Networks. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 53–61.
51. Shah, S.; Nawaz, W.; Jalil, B.; Khan, H.A. Classification of Normal and Leukemic Blast Cells in B-ALL Cancer Using a Combination of Convolutional and Recurrent Neural Networks. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 23–31.
52. Ding, Y.; Yang, Y.; Cui, Y. Deep Learning for Classifying of White Blood Cancer. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 33–41.
53. Kulhalli, R.; Savadikar, C.; Garware, B. Toward Automated Classification of B-Acute Lymphoblastic Leukemia. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 63–72.
54. Khan, M.A.; Choo, J. Classification of Cancer Microscopic Images via Convolutional Neural Networks. In *Lecture Notes in Bioengineering*; Springer: Singapore, 2019; pp. 141–147.
55. de Oliveira, J.E.M.; Dantas, D.O. Classification of Normal versus Leukemic Cells with Data Augmentation and Convolutional Neural Networks. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP, INSTICC, Online, 8–10 February 2021; SciTePress: Setúbal Municipality, Portugal, 2021; Volume 4, pp. 685–692. [CrossRef]

56. Pepe, M.S. *The Statistical Evaluation of Medical Tests for Classification and Prediction*; Oxford Statistical Sciences Series; Oxford University Press: New York, NY, USA, 2003.
57. Metrock, L.K.; Summers, R.J.; Park, S.; Gillespie, S.; Castellino, S.; Lew, G.; Keller, F.G. Utility of peripheral blood immunophenotyping by flow cytometry in the diagnosis of pediatric acute leukemia. *Pediatr. Blood Cancer* **2017**, *64*, e26526. [[CrossRef](#)]
58. Lam, G.; Punnett, A.; Stephens, D.; Sung, L.; Abdelhaleem, M.; Hitzler, J. Value of flow cytometric analysis of peripheral blood samples in children diagnosed with acute lymphoblastic leukemia. *Pediatr. Blood Cancer* **2018**, *65*, e26738. [[CrossRef](#)]
59. Beltrame, M.P.; Souto, E.X.; Yamamoto, M.; Furtado, F.M.; Costa, E.S.; Sandes, A.F.; Pimenta, G.; Cavalcanti Júnior, G.B.; Santos-Silva, M.C.; Lorand-Metze, I.; et al. Updating recommendations of the Brazilian Group of Flow Cytometry (GBCFLUX) for diagnosis of acute leukemias using four-color flow cytometry panels. *Hematol. Transfus. Cell Ther.* **2021**, *43*, 499–506. [[CrossRef](#)]
60. Safont, G.; Salazar, A.; Vergara, L. New Applications of Late Fusion Methods for EEG Signal Processing. In Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2019; pp. 617–621. [[CrossRef](#)]
61. Amari, S. Integration of Stochastic Models by Minimizing alpha-Divergence. *Neural Comput.* **2007**, *19*, 2780–2796. [[CrossRef](#)]