

Feature Extraction using CNN for Peripheral Blood Cells Recognition

Mohammed Ammar^{1,*}, Mostafa El Habib Daho², Khaled Harrar¹, Amel Laidi³

¹LIST Laboratory, University M'Hamed Bougara, Boumerdes, Algeria

²Biomedical Engineering Laboratory, University of Tlemcen, Tlemcen, Algeria

³LIMOSE Laboratory M'Hamed Bougara University Boumerdes, Algeria

Abstract

INTRODUCTION: The diagnosis of hematological diseases is based on the morphological differentiation of the peripheral blood cell types.

OBJECTIVES: In this work, a hybrid model based on CNN features extraction and machine learning classifiers were proposed to improve peripheral blood cell image classification.

METHODS: At first, a CNN model composed of four convolution layers and three fully connected layers was proposed. Second, the features from the deeper layers of the CNN classifier were extracted. Third, several models were trained and tested on the data. Moreover, a combination of CNN with traditional machine learning classifiers was carried out. This includes CNN_KNN, CNN_SVM (Linear), CNN_SVM (RBF), and CNN_AdaboostM1. The proposed methods were validated on two datasets. We have used a public dataset containing 12444 images with four types of leukocytes to find the best optimizer function (eosinophil, lymphocyte, monocyte, and neutrophil images). The second dataset contains 17,092 images divided into eight groups: lymphocytes, neutrophils, monocytes). the second public dataset was used to find the best combination of CNN and the machine learning algorithms. the dataset containing 17,092 images: lymphocytes, neutrophils, monocytes, eosinophils, basophils, immature granulocytes, erythroblasts, and platelets.

RESULTS: The results reveal that CNN combined with AdaBoost decision tree classifier provided the best performance in terms of cells recognition with an accuracy of 88.8%, demonstrating the performance of the proposed approach.

CONCLUSION: The obtained results show that the proposed system can be used in clinical practice.

Received on 01 July 2021; accepted on 15 October 2021; published on 20 October 2021

Keywords: Peripheral Blood Cells, CNN, Feature extraction, SVM, KNN, AdaboostM1.

Copyright © 2021 M. Ammar *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](https://creativecommons.org/licenses/by/4.0/), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.20-10-2021.171548

1. Introduction

The Peripheral blood cells can be divided into three classes: erythrocytes (red blood cells), leukocytes (white blood cells), and thrombocytes (platelets). Leukocytes are the cells of the immune system responsible for protecting the body from infections and foreign invaders. Each cell is composed of a nucleus and cytoplasm. The White Blood Cell Differential Count measures the WBCs types present in the bone marrow,

and it provides the necessary information for the diagnosis purposes. This operation is performed manually by hematologists using microscopes. Such a traditional method can be a very time-consuming task, raising the need for a tool that automatically performs this task. Acute lymphoblastic leukemia (ALL) is a rapidly developing form of WBC's malignancy, causing overproduction of the lymphoblasts. According to the World Health Organization (WHO) classification, ALL can be classified into subtypes: T-lymphoblastic leukemia, B-lymphoblastic leukemia, and mature-B lymphoblastic leukemia (mature-B).

The main contribution of our work is the proposition

*Corresponding author. Email: m.ammar@univ-boumerdes.dz

of a hybrid model based on CNN features extraction and AdaboostM1 decision Tree classifier for peripheral blood cells recognition.

We use CNN to extract features from the training images. The network concepts a hierarchical representation of input images. Early layers contain lower-level features that build up into higher-level features as the network goes deeper. The CNN model is composed of four convolutional layers and four fully connected layers.

The features extracted from the third fully connected layer are classified by CNN, CNN_KNN, CNN_SVM with linear kernel, CNN_SVM with RBF kernel, and CNN_AdaboostM1.

The remainder of this paper is organized as follows: Section 2 presents the related works, Section 3 presents the material and methods used in this work, Section 4 presents the obtained result and discussions, and Section 5 presents the conclusion and future work.

2. Related works

Many works have been proposed for white blood cells' classification based on deep learning methods. Among these methods, the Convolutional Neural Networks (CNN) architectures are widely used.

In [1], a new CNN architecture called WBCNet was proposed. This model combines batch normalization, residual convolution architecture, and improved activation function, allowing it to fully extract features from the microscopic WBC image. The WBCNet model contains 33 layers and is remarkably faster than a traditional CNN model during training. The accuracy on the test dataset was 83%.

In the paper [2], the authors introduce PatternNet-fused Ensemble of Convolutional Neural Networks (PECNN), a new architecture for WBCs classification. The proposed method fuses outputs of n randomly generated CNNs using an ensemble method, PatternNet. The PatternNet captures the strengths of each participating model while remaining insensitive to outliers.

Authors in [3] presented an improved hybrid approach for efficiently classifying WBC Leukemia. The model is based on features extraction from WBC images using VGGNet. The approach then uses a statistically enhanced Salp Swarm Algorithm (SESSA) to filter the extracted features. They reported some of the best-ranked results on the used datasets and outperformed several convolutional network models.

In [4], the CNN was paired one at a time with Densenet201, Alexnet, GoogleNet, and Resnet50. On the other hand, the dataset images were treated with Gaussian and median filters separately, then classified by CNN models. The new dataset with filtered images

showed better results than the original data.

An improved algorithm based on feature weight adaptive K-means clustering for extracting complex leukocytes was proposed in [5]. First, the color space was decomposed. Then, the image was segmented using a combination of the color space decomposition and K-means clustering. Then, the watershed algorithm was used to separate adherent complex white blood cells. Finally, classification tasks based on the CNN were performed, and the obtained results were compared with other methods. The proposed method achieved 95.8% of segmentation accuracy.

In the study published in [6], two Convolutional Neural Networks—VGG-16 and ResNet-50—were employed to classify five categories of white blood cells: Basophil, Eosinophil, Neutrophil, Lymphocyte, Monocyte. The results showed that ResNet-50 provided the best performance and achieved 88.29% of accuracy.

Authors in paper [7] applied convolutional neural network (CNN) for the image classification of four types of leucocytes, namely, eosinophils, lymphocytes, monocytes, and neutrophils. They used the Genetic Algorithm (GA) for CNN's hyperparameters optimization. The optimized CNN obtained a classification accuracy of 91% for the validation set and 99% for the training set. It also achieved a sensitivity and specificity of 91% and 97%.

In the paper [8], the authors used a data augmentation method based on generative adversarial networks (GAN), then used VGG-16, ResNet, and DenseNet for the recognition of leukocytes. The proposed method improves the classification performances of WBCs. The DenseNet-169 model provided the best accuracy (98.8%).

Another proposed framework [9] was developed based on a comparative study between Convolutional Neural Networks and other architectures. The proposed method improves the accuracy rate and loss in both detection and classification of WBCs. The proposed system in [10] consisted of a deep neural convolution network (DCNN) enhanced with a modified loss function besides regularization. The proposed system improved the classification accuracy from 96.1% to 98.92% and a decrease in processing time from 0.354 to 0.216 s.

In [11], an automatic approach for WBCs' detection and classification from peripheral blood images was proposed. The detection process used firstly local binary pattern (LBP) features and SVM to separate basophil and eosinophil. As a second step, a convolution neural network was used to automatically extract high-level features from WBCs, and the features were then passed through a random forest to recognize the other types of WBCs.

Another work published in [12] explored the utility of a deep learning model to detect lymphocyte cells and the

subtypes; they proposed using a CNN classifier. This approach was compared to another method based on SVM. The obtained results showed that the proposed classifier offered better performance in identifying normal lymphocytes and pre-B cells. In addition, the sensitivity on the T-cells was quite low.

Recently, authors presented in [13] a system based on a convolutional neural network (CNN) model to identify ALL. They performed different data augmentation techniques to expand the training data. The proposed model achieved an accuracy of 95,54%.

In the paper [14], a VGG19 model was optimized by SSPSO algorithm. As a result, they obtained an accuracy of 98%.

Another work based on the histogram of oriented gradient feature of nuclei shapes was proposed in [15]. The segmentation method used the YCbCr color space and K means clustering technique. Finally, the extracted features were classified with SVM and backpropagation artificial neural network (ANN).

Hegde et al. proposed in [16] a robust model to detect the WBC nuclei in peripheral blood smear images.

As mentioned in this section, the problem of segmentation of white blood cells in blood smears has been largely addressed by classic methods of machine learning or by CNN models. On the other hand, the detection of the various components circulating in the blood has been little treated in the literature.

In this work, We have proposed a CNN model composed of four convolution layers and three fully connected layers in our work.

This model has been used as a feature extractor, and the classification was performed with KNN, SVM with linear kernel, SVM with RBF kernel function, and the Adaboostm1 algorithm.

3. Materials and methods

3.1. Convolutional neural network (CNN)

The convolutional neural network (CNN) is the basis of the most popular algorithms in deep learning. It has been applied with success in different fields. For example, the CNN models have been used in image segmentation, classification, and object detection. The model proposed in this work has the following layers :

The CNN model has 22 layers: input layer, 4 convolution layers, 3 max-pooling layers, and 3 fully connected layers.

3.2. Adaptive moment estimation optimizer

Adaptive Moment Estimation (Adam) [17] is a way of computing adaptive learning rates for each parameter. It combines two stochastic gradient descent approaches,

1	Image Input	100x100x3 images
2	Convolution	8 5x5 convolutions
3	Batch Normalization	Batch normalization
4	ReLU	ReLU
5	Max Pooling	2x2 max pooling
6	Convolution	16 5x5 convolutions
7	Batch Normalization	Batch normalization
8	ReLU	ReLU
9	Max Pooling	2x2 max pooling
10	Convolution	32 5x5 convolutions
11	Batch Normalization	Batch normalization
12	ReLU	ReLU
13	Max Pooling	2x2 max pooling
14	Convolution	64 5x5 convolutions
15	Batch Normalization	Batch normalization
16	ReLU	ReLU
17	Max Pooling	2x2 max pooling
18	Fully Connected	32 fully connected layers
19	Fully Connected	16 fully connected layers
20	Fully Connected	8 fully connected layers
21	Softmax	Softmax
22	Classification	Output

Adaptive Gradients and Root Mean Square Propagation. Adam keeps an exponentially decaying average of past gradients m_t , just like momentum:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2)$$

Where m_t and v_t are parameters calculate the mean and the uncentered variance of the gradients. β_1 and β_2 are the bias

The authors of Adam propose to compute the bias-corrected first and second-moment estimates:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3)$$

$$\widehat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (4)$$

The authors show that Adam provides good results when the default values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are used.

3.3. Stochastic gradient descent with momentum optimizer

The Stochastic gradient descent (SGDM) algorithm makes a parameter update for each training example: $x^{(i)}$ and label $y^{(i)}$:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (5)$$

Batch gradient descent recomputes gradients for similar examples before each parameter update in

large datasets. Thus, SGDM becomes faster by performing one update at a time, then decreasing the redundancy[18].

3.4. Support Vector Machines

Cortes and Vapnik [19],[20] invented support vector machines (SVM) in 1995. They are utilized to solve various learning issues, including pattern recognition, text classification, and even medical diagnosis. The SVM is based on the ideas of the greatest margin and the kernel function. They are used to tackle nonlinear discrimination issues.

The margin is the distance between the separation boundary and the support vectors, the closest samples. In more technical terms, a support vector machine creates a hyperplane or group of hyperplanes in a high-dimensional space that may be used to classify data.

3.5. Adaptive Boosting with Decision Trees

Yoav Freund and Robert Schapire published the Adaboost (for Adaptive Boosting) in 1995 [21], which is a meta-algorithm that employs the boosting principle to enhance the performance of classifiers. The goal is to give each of the learning set's examples a weight.

They all start with the same weight, but as each iteration progresses, the weights of misclassified components will grow while those of correctly classified elements will decrease. AdaBoost can be applied on top of any classifier to learn from its limitations and develop a more accurate model, being the "best out-of-the-box classifier." Therefore, the most suited and most common algorithm used with AdaBoost is decision trees with one level, or what is frequently called decision stumps.

3.6. Datasets

In this work, we used two data sets: The first dataset is divided into four folders, each folder corresponding to a different blood cell type, and containing approximately 2500 images for training and 600 images in train data and test data for testing, respectively.

The training folder contains 2497 eosinophil images, 2483 lymphocyte images, 2478 monocyte images, and 2499 neutrophil images. The testing folder contains 623 eosinophil images, 620 lymphocyte images, 620 monocyte images, and 624 neutrophil images.

The data set is publically available at <https://www.kaggle.com/paultimothymooney/blood-cells>.

The second dataset is available in the Mendeley repository [22][23]: "A dataset for microscopic peripheral blood cell images for the development of automatic recognition systems "Data identification

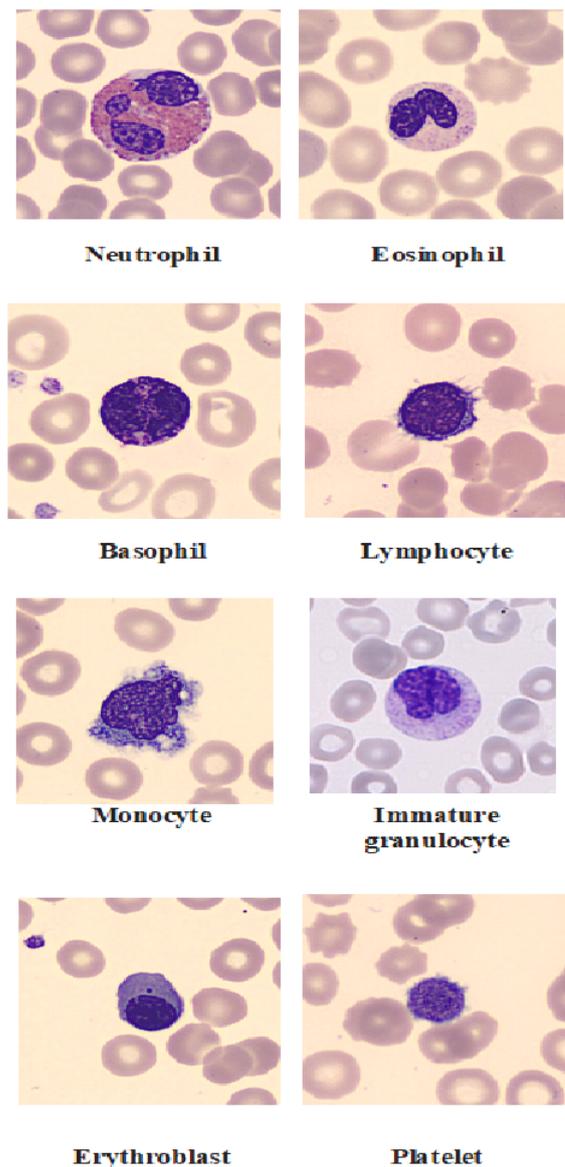


Figure 1. The eight White Blood Cells Types available in the dataset [22]

number: 10.17632/snkd93bnjr.1. The dataset contains 17,092 images :lymphocytes (1214), neutrophils (3329), basophils (1218), monocytes (1420), eosinophils (3117), immature granulocytes (2895), erythroblasts (1551) and platelets (2348). An example of the eight components is presented in figure 1.

The evaluation of the classifiers is based on the following parameters:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (6)$$

$$Sensitivity = \frac{TP}{((TP + FN))} \quad (7)$$

Output Class \ Target Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL	
EOSINOPHIL	556 18.6%	46 1.5%	10 0.3%	58 1.9%	83.0% 17.0%
LYMPHOCYTE	25 0.8%	662 22.2%	1 0.0%	24 0.8%	93.0% 7.0%
MONOCYTE	43 1.4%	16 0.5%	722 24.2%	43 1.4%	87.6% 12.4%
NEUTROPHIL	125 4.2%	21 0.7%	10 0.3%	625 20.9%	80.0% 20.0%
	74.2% 25.8%	88.9% 11.1%	97.2% 2.8%	83.3% 16.7%	85.9% 14.1%

Figure 2. Confusion Matrix for CNN with 20 epochs

$$NPV = \frac{TN}{((FN + TN))} \tag{8}$$

4. Results and discussion

4.1. Influence of the maximum number of epochs

We used the first dataset to study the impact of the maximum number of epochs on the results. This parameter was set to 10 then to 20. The global accuracy (figure 2), sensitivity (figure 3), and PNV (figure 4) were estimated. The Adam function was used with 'InitialLearnRate' = 3e-4, and 'SquaredGradientDecayFactor' = 0.99. The Accuracy accuracy achieved a rate of 70.5% after ten iterations and 85.9% after 20 iterations. These findings lead to the conclusion that the number of maximum epochs impacts the performances of the classifier.

The negative predictive value after 10 iterations: 66,8% for the eosinophil, 83,60% for the lymphocyte, 73,90% for the monocyte, and 61,5% for the neutrophil. The negative predictive value after 20 iterations: 83% for the eosinophil, 93,00% for the lymphocyte, 87,60% for the monocyte, and 80% for the neutrophil.

The sensitivity after 10 iterations: 56,30% for the eosinophil, 71,70% for the lymphocyte, 80,5% for the monocyte, and 73,6% for the neutrophil.

The sensitivity after 20 iterations: 74,20% for the eosinophil, 88,90% for the lymphocyte, 97,2% for the monocyte, and 83,3 % for the neutrophil.

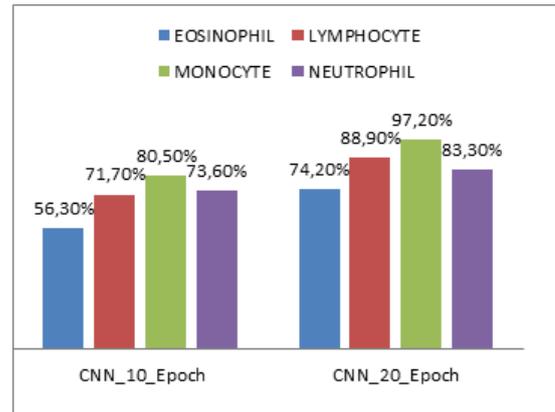


Figure 3. Sensitivity of the four types for CNN with 10 and 20 epochs

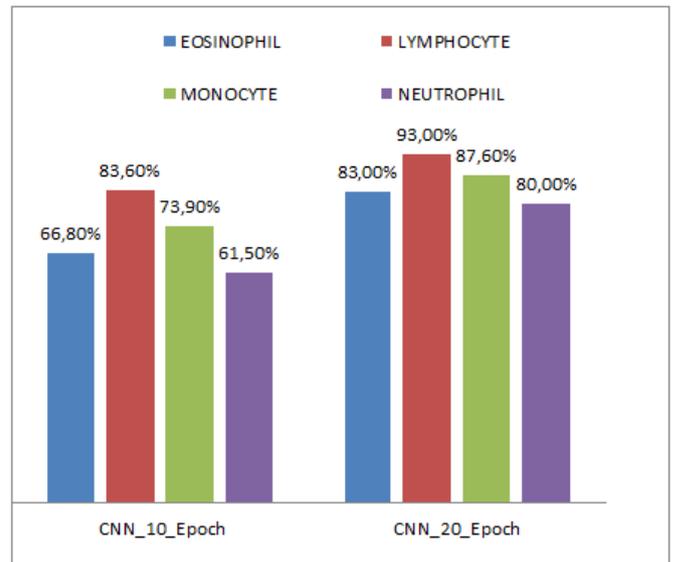


Figure 4. Negative predictive value of the four types for CNN with 10 and 20 epochs

4.2. Influence of the training algorithm

In this section, the influence of the optimizer function on the performances of the CNN classifier was performed. First, two algorithms were carried out: ADAM and SGDM. Then we have tested the CNN model on the first dataset containing 3100 images and four leukocytes. The results were described in 1. In addition, the used parameters were also reported.

From table 1, we can see that using the ADAM optimizer function constantly improves the accuracy and decreases training time compared to using the SGDM function. For 120 epochs, we obtained an accuracy of 92.23.

We reported in table 2 the results obtained with the CNN, CNN_KNN, CNN_SVM (Linear), CNN_SVM (RBF), and CNN_AdaboostM1 models using the second

Table 1. Comparison between ADAM and SGDM

Simple_CNN				
Max Epochs	Training Options "SGDM"		Training Options "ADAM"	
	MiniBatchSize = 120	MiniBatchSize = 64	MiniBatchSize = 120	MiniBatchSize = 64
64	Accuracy	80.05	Accuracy	87.98
100	Accuracy	84.73	Accuracy	89.19
120	Accuracy	82.26	Accuracy	92.23

dataset[23].

The best accuracy were was obtained with CNN_Adaboost 88,8%. The best sensitivity of the eight components are listed as follows:

- Neutrophil: Sensitivity = 99.6% using CNN.
- Eosinophil: Sensitivity = 99.7% using CNN_SVM (linear).
- Basophil: Sensitivity = CNN with 97.9% using CNN.
- Lymphocyte: Sensitivity = 91,3% using CNN_KNN.
- Monocyte: Sensitivity = 85.4% using CNN.
- IG: Sensitivity = 66.1% using CNN_Adaboost.
- Erythroblast: Sensitivity = 86% using CNN_KNN.
- Platelet: Sensitivity =98.8% using CNN.

In practice, diagnosing hematological diseases is based on microscopes' morphological differentiation of the peripheral blood cell in laboratories; the traditional method is very tedious and time-consuming. Therefore, an automatic differential counting system is preferred. Automatic recognition of peripheral blood cells using classical machine-learning approaches has been widely treated in the literature [24, 25].

With the emergence of the CNN architectures, many works have been proposed to perform the automatic segmentation and classification of the white blood cells [8–10].

The classification of the eight components of the peripheral blood cells was proposed in the paper[22]. The authors collected a data set that contained the eight components: neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (metamyelocytes, myelocytes, and promyelocytes), erythroblasts, and platelets (thrombocytes). They have used vgg16 and inceptionv3 as features extractors and the SVM as classifiers.

We have proposed a CNN model composed of four convolution layers and three fully connected layers in our work.

This model has been used as a feature extractor, and the classification was performed with KNN, SVM with linear kernel, SVM with RBF kernel function, and the Adaboostm1 algorithm.

We have found that the Adaboost M1 classifier provides the best results. The CNN and KNN also provided high accuracy (88,6%) and ranked in the second position after AdaboostM1. Moreover, using CNN alone showed a competitive performance in cell recognition, where it gave a rate of 88,4% of accuracy. On the other hand, the combination of SVM and CNN provided different results depending on the nature of the kernel function. For example, for a Gaussian kernel function, the accuracy was promising (88,5%) due to the suitability of the RBF function. However, for a linear function, the output accuracy was poor with a rate of 44,3%. This is explained by the fact that blood cell recognition is not a linearly separable problem.

5. Conclusion

In this work, we have proposed a features extractor based on CNN combined with machine learning methods to recognize peripheral blood cells automatically. At first, we have studied the influence of training parameters on the performance of the CNN classifier, namely, the optimizer function (ADAM and SGDM) and the maximum number of epochs. The best accuracy reached was 92,23% and was obtained with an ADAM function and 120 epochs.

In the second step, we extract the features from the deeper layers of the CNN classifier, and then we classified these features using CNN, CNN_KNN, CNN_SVM (Linear), CNN_SVM (RBF) and CNN_AdaboostM1 classifiers.

The best accuracy was achieved with the AdboostM1 algorithm (88,8%). We propose to test other CNN models such as VGG19, Resnet, and Inception in future works.

References

- [1] JIANG, M., CHENG, L., QIN, F., DU, L. and ZHANG, M. (2018) White blood cells classification with deep convolutional neural networks. *International Journal of Pattern Recognition and Artificial Intelligence* 32(09): 1857006. doi:10.1142/S0218001418570069, URL <https://doi.org/10.1142/S0218001418570069>.
- [2] WANG, J.L., LI, A.Y., HUANG, M., IBRAHIM, A.K., ZHUANG, H. and ALI, A.M. (2018) Classification of white blood cells with patternnet-fused ensemble of convolutional neural networks (pecnn). In

Table 2. Comparison between CNN, CNN_KNN, CNN_SVM (Linear), CNN_SVM (RBF) and CNN_Adaboost

		Neutrophil	Eosinophil	Basophil	Lymphocyte	Monocyte	IG	Erythroblast	Platelet
CNN		Accuracy = 84.6							
	Sensitivity	99,6	98.9	97.9	85.4	85.4	47,9	72	98.8
CNN_KNN		Accuracy = 88.6							
	Sensitivity	98.7	97.3	90.4	91.3	84.4	62.1	86	97.8
CNN_SVM(linear)		Accuracy = 44.3							
	Sensitivity	66	99.7	3.4	11.9	1.9	20	12.3	61.8
CNN_SVM(RBF)		Accuracy = 88.5							
	Sensitivity	99.1	97.6	91.6	91.3	84.2	61.2	83.2	89.3
CNN_Adaboost		Accuracy = 88.8							
	Sensitivity	98.7	97.6	90	88.4	83.4	66.1	84.3	97.7

2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT): 325–330. doi:10.1109/ISSPIT.2018.8642630.

- [3] AHMED T., S., PHILIP, K. and AHMED A., E. (2020) Efficient classification of white blood cell leukemia with improved swarm optimization of deep features. *Scientific Reports* 10(1). doi:10.1038/s41598-020-59215-9, URL <https://doi.org/10.1038/s41598-020-59215-9>.
- [4] YILDIRIM, M. and ÇINAR, A. (2019) Classification of white blood cells by deep learning methods for diagnosing disease. *Revue d'Intelligence Artificielle* 33(5): 335–340. doi:10.18280/ria.330502, URL <https://doi.org/10.18280/ria.330502>.
- [5] LIN, L., WANG, W. and CHEN, B. (2019) Leukocyte recognition with convolutional neural network. *Journal of Algorithms & Computational Technology* 13: 1748301818813322. doi:10.1177/1748301818813322, URL <https://doi.org/10.1177/1748301818813322>.
- [6] SUPAWIT, V., SUCHAT, T. and KITSUCHART, P. (2018) White blood cell classification: A comparison between vgg-16 and resnet-50 models. In *The 6th Joint Symposium on Computational Intelligence (JSCI6)*.
- [7] DANA, B.H., NASEEM, K., FATIMAH, A., SHREYA, K. and NAGARUR, N. (2018) Classification of leucocytes using convolutional neural network optimized through genetic algorithm. In *Proceedings of the 7th Annual World Conference of the Society for Industrial and Systems Engineering, Binghamton, NY, USA*.
- [8] ALMEZHGHWI, K. and SERTE, S. (2020) Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network. *Computational Intelligence and Neuroscience* 2020.
- [9] KOUSALYA, B., DINESH, K., MOHANA, R. and KRISHNAKUMAR, B. (2020) Blood cells classification using convolutional neural network architecture. *International Journal of Advanced Science and Technology* 29(3s): 261–267.
- [10] BASNET, J., ALSADOON, A., PRASAD, P., ALOUSSI, S. and ALSADOON, O.H. (2020) A novel solution of using deep learning for white blood cells classification: Enhanced loss function with regularization and weighted loss (elfrwl). *Neural Processing Letters* : 1 – 37.
- [11] ZHAO, J., ZHANG, M., ZHOU, Z., CHU, J. and CAO, F. (2016) Automatic detection and classification of leukocytes using convolutional neural networks. *Medical & Biological Engineering & Computing* 55: 1287–1301.
- [12] PANSOMBUT, T., WIKAISUKSAKUL, S., KHONGKRAPHAN, K. and PHON-ON, A. (2019) Convolutional neural networks for recognition of lymphoblast cell images. *Computational Intelligence and Neuroscience* 2019.
- [13] ANWAR, S. and ALAM, A. (2020) A convolutional neural network-based learning approach to acute lymphoblastic leukaemia detection with automated feature extraction. *Medical & biological engineering & computing* .
- [14] KUMAR, R., JOSHI, S. and DWIVEDI, A. (2021) Cnn-sspso: A hybrid and optimized cnn approach for peripheral blood cell image recognition and classification. *International Journal of Pattern Recognition and Artificial Intelligence* 35(05): 2157004. doi:10.1142/S0218001421570044, URL <https://doi.org/10.1142/S0218001421570044>.
- [15] MOHD NOOR, A., YAZID, H., ZAKARIA, Z. and MOHD NOOR, A. (2020) Classifying white blood cells from a peripheral blood smear image using a histogram of oriented gradient feature of nuclei shapes. *Engineering, Technology and Applied Science Research* : 129–136doi:10.14456/easr.2020.13.
- [16] HEGDE, R., PRASAD, K., HEBBAR, H. and SINGH, B. (2019) Development of a robust algorithm for detection of nuclei of white blood cells in peripheral blood smear images. *Multimedia Tools and Applications* 78(13): 17879–17898. doi:10.1007/s11042-018-7107-x.
- [17] KINGMA, D.P. and BA, J. (2015) Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- [18] SUTTON, R.S. (1986) Two problems with backpropagation and other steepest-descent learning procedures for networks. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (Hillsdale, NJ: Erlbaum).

- [19] CORTES, C. and VAPNIK, V. (1995) Support-vector networks. *Machine Learning* 20(3): 273–297. doi:10.1023/A:1022627411411, URL <https://doi.org/10.1023/A:1022627411411>.
- [20] BOTTOU, L., CHAPPELLE, O., DE COSTE, D. and WESTON, J. (2007) Support vector machine solvers.
- [21] FREUND, Y. and SCHAPIRE, R. (1996) Experiments with a new boosting algorithm. In *ICML*: 148–156.
- [22] ACEVEDO, A., ALFÉREZ, S., MERINO, A., PUIGVÍ, L. and RODELLAR, J. (2019) Recognition of peripheral blood cell images using convolutional neural networks. *Computer methods and programs in biomedicine* 180: 105020.
- [23] (2020), A dataset for microscopic peripheral blood cell images for development of automatic recognition systems. doi:10.17632/snkd93bnjr.1, URL <https://data.mendeley.com/datasets/snkd93bnjr/1>.
- [24] RAMESH, N., DANGOTT, B.J., SALAMA, M. and TASHIZEN, T. (2012) Isolation and two-step classification of normal white blood cells in peripheral blood smears. *Journal of Pathology Informatics* 3.
- [25] SU, M., CHENG, C.Y. and WANG, P.C. (2014) A neural-network-based approach to white blood cell classification. *The Scientific World Journal* 2014.