Few-Shot Learning using Siamese Twin Network for the Classification of Blood Cells

Sudhakar Tummala (✉ sudhakar.t@srmap.edu.in)  
SRM University AP  https://orcid.org/0000-0001-5735-9418

Anil K Suresh  
SRM University AP

Research Article

Keywords: Siamese network, EfficientNet, few-shot learning, blood cell typing, explainability

Posted Date: October 31st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2215631/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Abstract

Automated classification of blood cells from microscopic images is an interesting research area owing to advancements of efficient neural network architectures. Here, we developed a few-shot contrastive learning model for the classification of peripheral blood cells including lymphocytes, monocytes, basophils, eosinophils, neutrophils, immature granulocytes, erythroblasts, and platelets using EfficientNet as a base model and contrastive loss as a loss function. A total of 17092 publicly accessible images acquired using the Cellavision DM96 were analyzed. From 125 images of each cell type, 20000 image pairs are generated for Siamese twin network (STN) training and another 125 images from each cell type are used for few-shot validation. Therefore, out of 17092 images, 6% were used for training, 6% for few-shot validation and rest 88% for few-shot testing. This architecture demonstrates an overall accuracy of 97.21% during 8-way 3-shot testing for the classification of all cell types with an accuracy of 97.72% for the classification of white blood cells alone. Further, we propose a novel class activation mapping scheme for the interpretability of the model decisions suitable for STN. To conclude, the proposed framework based on contrastive learning could be used for the fully automated self-exploratory classification and identification of peripheral blood cells.

Introduction

Automated classification of peripheral blood cells (e.g., basophils, neutrophils, eosinophils, monocytes, lymphocytes, immature granulocytes, erythroblasts, and platelets) is a prime requisite for medical researchers. As, it provides hassle-free support to cytologists and pathologists in analyzing blood smears under normal and disease states [1, 2]. Additionally, understanding the structural integrity of peripheral blood cells is crucial for the identification and monitoring of several diseases including hematologic conditions [3]. With the success of AlexNet on the ImageNet classification challenge [4], advancements in computational infrastructure, and parameter optimization methods, researchers across the globe are developing novel neural network architectures for various computer vision tasks. Prominent successful models after AlexNet are VGGNets, ResNets, DenseNets, MobileNets, Inception, Xception networks, and EfficientNets that not only outperform on a specific computer vision task, but the knowledge gained by them can be used to tackle other computer vision tasks with the advent of transfer learning approaches [5–7].

Previous studies for the classification of normal peripheral blood cell images employed few of the aforementioned state-of-the-art architectures and their variants as base models using transfer learning [3, 8, 9]. Studies also investigated the feasibility of deep learning models for the exclusive classification of white blood cells [10–13]. However, all these methods used multitudes of images for training leaving behind only few images for testing. Typically, large neural networks demand huge data to achieve better generalizability, and in medical imaging, data availability is a scarce and it is time consuming to collect large-scale data with quality acquaintance. Lately, few-shot learning (FSL) based methods are gaining tremendous attention as they relatively require less data for model training and yet are robust and efficient. The FSL based methods can be trained via contrastive learning (CL) of the input data either in pairs or triplets or quadruplets and by performing quantitative comparisons of their embeddings and representations [14–16]. If the embeddings are adjacent in the latent space, they belong to the same class and vice versa. Therefore, the objective of CL is to make the embeddings of the same class nearer and embeddings of dissimilar classes farther in the latent
space under some constraints [17, 18]. Hence, Siamese twin network (STN) can be constructed for CL as few recent studies have successfully employed these networks for several medical imaging investigations [19–23].

Several methods are proposed to highlight various important regions in the image for model decision in a classification task such as class activation mapping (CAM) [24], gradCAM [25], and gradCAM++ [26]. CAM uses the trained weights between the final convolution layer and the dense layer as the weights to be multiplied with the final activation maps. Whereas gradCAM uses the gradients obtained during backpropagation as weights which is the generalized version of CAM and gradCAM ++ and general form of gradCAM. All these methods are extensively applied in single feed forward network architectures for visual saliency mappings. Whereas, in STN models, the saliency mapping methods are sparse and adapt already existing gradCAM methods [27, 28]. EfficientNets have been a versatile neural network architectures that outperformed the aforementioned novel architectures, and the network has relatively few parameters. It also exhibits better speed during various image recognition tasks and performs satisfactorily on transfer learning diverse datasets [29]. EfficientNets work on more principled compound scaling of width, depth of the network, and resolution of the input images while maintaining model efficiency. Moreover, the visual saliency maps generated from EfficientNets show that the model focuses on more relevant object regions [29]. To the best of our knowledge, FSL with deep contrastive representation learning is not yet been explored for multi-labeled classification of normal peripheral blood cells along with visual saliency mappings for interpretability of the model predictions.

Therefore, herein, we conducted the following: (a) Developed an STN with EfficientNet as the base model and contrastive loss as a loss function for the classification of normal peripheral blood cells, (b) Eight-way few-shot verification for the identification of specific cell image type from the eight types of peripheral blood cells. The few-shot verification involves few-shot validation for hyperparameter tuning and few-shot testing, and eventually, c) Developed a novel visual saliency mapping scheme without necessitating gradients computation for explanation of the proposed STN decisions during few-shot verification.

**Methods**

This section describes the dataset of peripheral blood cell images, Siamese twin network training with EfficientNet as the base model, N-way few-shot validation and testing, performance metrics, and generation of class activation maps between query image and support set.
Table 1
Summary of the cell types in the dataset, number of images for each cell type, number of images used for Siamese twin network training, few-shot validation, and few-shot testing. N: number of images.

<table>
<thead>
<tr>
<th>Blood Cell Type</th>
<th>Training (N)</th>
<th>Validation (N)</th>
<th>Testing (N)</th>
<th>Total (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basophils</td>
<td>125</td>
<td>125</td>
<td>968</td>
<td>1218</td>
</tr>
<tr>
<td>Eosinophils</td>
<td>125</td>
<td>125</td>
<td>2867</td>
<td>3117</td>
</tr>
<tr>
<td>Neutrophils</td>
<td>125</td>
<td>125</td>
<td>3079</td>
<td>3329</td>
</tr>
<tr>
<td>Lymphocytes</td>
<td>125</td>
<td>125</td>
<td>964</td>
<td>1214</td>
</tr>
<tr>
<td>Monocytes</td>
<td>125</td>
<td>125</td>
<td>1170</td>
<td>1420</td>
</tr>
<tr>
<td>Immature Granulocytes</td>
<td>125</td>
<td>125</td>
<td>2645</td>
<td>2895</td>
</tr>
<tr>
<td>Erythroblasts</td>
<td>125</td>
<td>125</td>
<td>1301</td>
<td>1551</td>
</tr>
<tr>
<td>Platelets</td>
<td>125</td>
<td>125</td>
<td>2098</td>
<td>2348</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1000</strong></td>
<td><strong>1000</strong></td>
<td><strong>15092</strong></td>
<td><strong>17092</strong></td>
</tr>
</tbody>
</table>

**Dataset**

In the current work, an openly accessible dataset of normal peripheral blood cells was utilized from the Hospital Clinic of Barcelona. The dataset contains 17092 RGB images that were captured using the Cellavision DM96 [30]. The images of different types of blood cells include neutrophils, basophils, eosinophils, lymphocytes, monocytes, immature granulocytes, erythroblast, and platelets. Predominant image resolution was $360 \times 363$ with very few images having the resolution of $360 \times 360$ or $359 \times 360$. The images were graded with pre-determined cell types by expert clinical pathologists from the same clinic and were used as ground truth labels. Complete details about the dataset are given in Table 1. Sample images (one image per class) from the dataset are shown in Fig. 1.

**Generation of Image Pairs**

We have considered 125 images from each class with a total of 1000 images for training. Twenty image pairs are created for each image cumulating to a total of 20000 image pairs for STN training. When both the paired images belong to the same cell type it is termed as a positive pair labelled as 0 and if the images are dissimilar, they are designated as a negative pair labelled as 1. To avoid imbalance in the classes, 10000 positive pairs and 10000 negative pairs are randomly produced. Out of the 20 generated pairs for each image, ten were same pairs and ten were different pairs. To maintain uniformity in differently paired, we ensured that each specific cell was paired with rest of the seven cell types at least once. Before feeding the image pairs to the STN model, the images were initially verified to ensure that their intensity range is between zero and 255, a primary requirement for EfficientNets.

**Siamese Neural Network Architecture**
Figure 2 shows the proposed STN architecture with EfficientNet-B3 as the base model. The final softmax layer of the base model was discarded and a global average pooling layer is incorporated. The EfficientNet-B3 model was employed here to transform the input sample from image space to embedding space with a mapping \( \varphi(. \) ). For the input image pair \( X_1 \) and \( X_2 \), \( \varphi(X_1) \) and \( \varphi(X_2) \) are their embeddings in the latent space respectively. Since the goal of the STN model is to make the embeddings of similar pairs closer and vice versa, the quantitative comparison of the embeddings via absolute differences is implemented using the lambda layer. After which, a sigmoid neuron is placed leading to a probability between zero and one, where a value less than 0.5 indicates a positive pair and vice versa.

**EfficientNet and Tuning of Hyperparameters**

Since the softmax layer of the EfficientNet-B3 model was eliminated, it outputs a feature tensor of size \( 10 \times 10 \times 1536 \) which is also the output for the final convolution layer. Further, this output feature tensor is global averaged to achieve a feature vector of the size 1536. Furthermore, the 1536 feature tensor is connected to the lambda layer for its quantitative comparison with another feature tensor of length 1536 as described earlier. The base model approximately contains twenty million parameters and therefore to reduce the computational time as well as to leverage the power of transfer learning, selected parameters (weights and biases) of the final ten percent layers from the EfficientNet-B3 model were allowed to get updated during backpropagation and rest of the model parameters were nontrainable.

There are several hyperparameters in the proposed model that can be tuned such as mini-batch size, learning rate, number of epochs, and the choice of optimizer during gradient descent. We evaluated models with different possible combinations with respect to four adaptive gradient descent optimizers namely RMSprop [31], Adam [32], Nadam [33], Adadelta [34]. Finally, for few-shot testing, we selected the model that gave better accuracies during few-shot validation as described in Table 2.

**Contrastive Training**

For contrastive training, NVIDIA Tesla P100 GPU with 26 GB RAM available in Google Colab Pro is utilized. It has a TensorFlow backend with Keras API. As required by EfficientNet-B3, the images are center cropped to attain a resolution of 300×300. Further, the loss metric for updating the model parameters through backpropagation is contrastive loss \( (C_l) \) that is computed using Eq. (1) given below:

\[
C_l = (1 - y) \times \sigma ( d ( \varphi(X_1), \varphi(X_2)) )^2 + y \times \left\{ \max (0, m - \sigma (d(\varphi(X_1), \varphi(X_2))))\right\}^2
\]

where \( d(\varphi(X_1), \varphi(X_2)) \) is the distance metric and in this study, it is the weighted sum of the absolute differences between the embeddings \( \varphi(X_1) \) and \( \varphi(X_2) \). Above, \( y \) indicates the true label of the image pair, \( m \) is the distance margin that is set to one in the current study and \( \sigma \) is the sigmoid function as described in Eq. (2).

\[
\sigma (d(.)) = \frac{1}{1 + e^{-d(.)}}
\]

Let \( a_d \) is a tensor represents the absolute differences between the embedding tensors as given in Eq. (3).
Eventually, the distance metric $d(\varphi(X_1), \varphi(X_2))$ is represented using the expression in Eq. (4).

$$d(\varphi(X_1), \varphi(X_2)) = \sum_{i=1}^{N} w_i a_d^i + b_i \quad (4)$$

In Equations (3) and (4), $\varphi(X_1^i)$ and $\varphi(X_2^i)$ are the $i$th values of the embedding vectors $\varphi(X_1)$ and $\varphi(X_2)$ respectively, $N$ is the length of the tensor $a_d$ and $w_i$ and $b_i$ are the weights and biases that needs to be learned during backpropagation.

**Support Set**

A support set that is necessary for few-shot validation and testing containing eight images is formed from the images in the test set. To represent each class in the support set, one image is randomly sampled from the images of the corresponding class for $\mathcal{N}$-way $k$-shot validation and testing as detailed below. Whenever $k$ is greater than one, we used entirely new support set for each shot.

**N-way k-Shot Validation and Testing**

The value of $\mathcal{N}$ in $\mathcal{N}$-way is the number of classes which is set to eight in the present study and $k$ in $k$-shot is between one to five for few-shot learning. If $k$ is equal to one, it is called one-shot learning and if $k$ is equal to two, it is two-shot learning, and so on. In this study, we performed 8-way 1-shot, 8-way 2-shot, 8-way 3-shot, and 8-way 5-shot validation and testing for multiclass classification of eight cell types. Finally, the class of the query image is decided based on the highest similarity with respect to the images in the support set.

Mathematically, the class prediction for a query image $x_q$ for 8-way $k$-shot learning is given in Eq. (5).

$$y_q = \text{argmin} \left\{ \sum_{i=1}^{k} \Psi(X_q, X_s)_i \right\} \quad (5)$$

Above, $X_q = \{ x_q^c = x_q : 1 \leq c \leq N \}$, that means $X_q$ is the set of images where the query image $x_q$ is repeated $N$ times to match with the number of images in the support set $X_s = \{ x_s^c : 1 \leq c \leq N \}$ so that the comparison of $x_q$ with $X_s$ would happen in one epoch, $\Psi(X_q, X_s)$ is the prediction result of STN model which is a vector of $N$ similarity values between $X_q$ and $X_s$. Finally, $Y_q$ is the predicted label for the query image $x_q$ that is between zero and seven. The calculation of overall accuracy for predicting a class $c$ during $k$-shot validation/testing is implemented using expression (6), where $N'_c$ is correctly predicted images for class $c$ and $N_c$ is the total number of images in class $c$.

$$\text{Overall Accuracy} = \frac{\sum_{i=1}^{k} \left( \frac{N'_c}{N_c} \right)_i}{k} \quad (6)$$

**Creation of Visual Saliency Maps**
For the creation of saliency maps, to infer explainability to the network while making decisions, the output of the lambda layer of the STN is used as the weight tensor. The activation maps $A^{H\times W \times C}$ of the final convolution layer of the EfficientNet-B3 for the query image are multiplied with the weight vector to get weighted activation maps $A_w^{H\times W \times C}$ as described by Eq. (7).

$$A_w^{H\times W \times i} = A^{H\times W \times i} \ast a_d^i \forall i \quad (7)$$

In Eq. (5), $a_d^i$ is the weight tensor as already described in Eq. (3). Afterward, an average activation map $A_m^{H\times W}$ of spatial size of $H \times W$ is obtained by averaging all $C$ weighted activation maps as described in Eq. (8).

$$A_m^{H\times W} = ReLU\left(\frac{1}{C} \sum_{i=1}^{C} A_w^{H\times W \times i}\right) \quad (8)$$

The negative values in the mean activation map are removed using the $ReLU$ (rectified linear unit) activation function, which is given in Eq. (9).

$$ReLU \left( z \right) = \begin{cases} z & if \; z > 0 \\ 0 & if \; z \leq 0 \end{cases} \quad (9)$$

For EfficientNet-B3, $H \times W \times C = 10 \times 10 \times 1536$. The depth of the activation maps $C$ and length of the weight tensor $N$ are identical. Finally, the 10×10 coarse activation map $A_m^{H\times W}$ is resized to match with the spatial resolution of the input query image which is 300×300 using python based scikit-image toolbox. For identification of highly activated regions in the query image when it is compared with the images in the support set, the resized heatmaps are overlaid onto the corresponding RGB images in the support set showing the most similar/dissimilar regions.

**Results**

The 8-way few-shot validation with the overall accuracies of the proposed STN architecture using EfficientNet-B3 as the base model are given in Table 2. For different combinations of mini batch gradient descent optimizers and other model hyperparameters where Adadelta performed relatively better are ascribed. Hence, the selected hyperparameters for few-shot testing along with Adadelta optimizer are mini-batch size: 16, learning rate: 1.0, number of epochs: 15. The overall validation accuracy values are in the range of 94.7%-97.1%. The 8-way few-shot test overall accuracies are given in Fig. 6 and the best overall test accuracy over 97.2% is obtained when $k$ is equal to 3. The few-shot test overall accuracy for the classification of white blood cells alone is 97.7%. The classification value is highest for platelets with 99.8%. Table 3 shows the results for overall test accuracies obtained in comparison with the previous studies. The number of images used for training, validation, and testing are also mentioned for a fair comparison.

The saliency maps highlighting the most discriminative/similar regions between the query image and the images in the support sets for basophil and erythroblast cells are given in Fig. 3 and Fig. 4 respectively. The 1D tensor $a_d$ is reshaped into a 2D tensor of size $48 \times 32$ and is also plotted along with the activation maps. Further, Fig. 5 shows the saliency maps highlighting the regions for each cell type comparatively with another
image of the same cell type in the support set to understand the reason for their similarity. As anticipated, the regions where most of the representative features located for decision making are highlighted and those regions are cell bodies and their surroundings.
Table 2
Different optimizers and hyperparameters along with accuracy values during Siamese network training and overall accuracy values during 8-way 1-shot, 2-shot, 3-shot, and 5-shot validation on 1000 images with 125 samples from each class. *SGD*; stochastic gradient descent, *lr*; learning rate, *ne*; number of epochs, *mbs*; mini-batch size.

<table>
<thead>
<tr>
<th>Optimizer &amp; Hyperparameters</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Siamese training</td>
</tr>
<tr>
<td>RMSprop</td>
<td>99.80</td>
</tr>
<tr>
<td></td>
<td>99.89</td>
</tr>
<tr>
<td></td>
<td>99.93</td>
</tr>
<tr>
<td>lr = 0.001, ne = 15, mbs = 16</td>
<td>99.80</td>
</tr>
<tr>
<td>lr = 0.001, ne = 15, mbs = 32</td>
<td>99.89</td>
</tr>
<tr>
<td>lr = 0.0005, ne = 20, mbs = 64</td>
<td>99.93</td>
</tr>
<tr>
<td>Adam</td>
<td>99.74</td>
</tr>
<tr>
<td></td>
<td>99.79</td>
</tr>
<tr>
<td></td>
<td>99.91</td>
</tr>
<tr>
<td>lr = 0.001, ne = 15, mbs = 16</td>
<td>99.74</td>
</tr>
<tr>
<td>lr = 0.001, ne = 15, mbs = 32</td>
<td>99.79</td>
</tr>
<tr>
<td>lr = 0.0005, ne = 20, mbs = 64</td>
<td>99.91</td>
</tr>
<tr>
<td>Nadam</td>
<td>99.84</td>
</tr>
<tr>
<td></td>
<td>99.81</td>
</tr>
<tr>
<td></td>
<td>99.74</td>
</tr>
<tr>
<td>lr = 0.001, ne = 15, mbs = 16</td>
<td>99.84</td>
</tr>
<tr>
<td>lr = 0.01, ne = 15, mbs = 32</td>
<td>99.81</td>
</tr>
<tr>
<td>lr = 0.01, ne = 20, mbs = 64</td>
<td>99.74</td>
</tr>
<tr>
<td>Adadelta</td>
<td>99.97</td>
</tr>
<tr>
<td></td>
<td>99.94</td>
</tr>
<tr>
<td></td>
<td>99.95</td>
</tr>
<tr>
<td>lr = 1.0, ne = 15, mbs = 16</td>
<td>99.97</td>
</tr>
<tr>
<td>lr = 1.0, ne = 15, mbs = 32</td>
<td>99.94</td>
</tr>
<tr>
<td>lr = 0.9, ne = 20, mbs = 64</td>
<td>99.95</td>
</tr>
</tbody>
</table>
Discussion

In this study, we assessed multi label classification of normal peripheral blood cell images that include white blood cells, immature granulocytes, erythroblasts, and platelets using STN architecture based on contrastive learning. Extensive evaluations for the choice of the optimizer and other model hyperparameters were performed that suggested the optimal choice based on their performance during few-shot validation. In general, the performance scores of the models given in Table 2 and were determined to be decent irrespective of the optimizer and other hyperparameters, with Adadelta optimizer as a slightly better variation. As expected, based on \( a_2 \) plots from Figs. 3 and 4, it can be inferred that the embeddings are closer for similar pairs and apart for dissimilar pairs. Our proposed method performed better [3] or slightly worse [9] compared to previous studies that were tested on the same dataset. Even though the overall few-shot testing accuracy is two percent less than the test accuracies from the reported study [8], our methodology is completely different where we have utilized only 6\% of the data for training to further propel the feasibility of CL based models in the medical domain.

As also discussed in [3], due to the nucleus shape and structure similarities among igs, basophils, neutrophils, and monocytes, there is a higher percent of misclassifications among these cell types. From Fig. 6, we can notice that 2.27\%, 2.53\% of igs are wrongly classified as monocytes and neutrophils respectively. Similarly, 2.48\% of basophils are misclassified as igs. The classification accuracy for platelets is 99.8\% since their size and morphology are entirely different compared to other cell types. In terms of overall test accuracy for the classification of only white blood cells, our results are comparable with several studies that used deep transfer learning approaches or much complicated neural network designs [10–13]. Our study achieved an accuracy of 97.7\% for white blood cell classification which is better than some of the reported accuracies 96.1\% and 96.9\% respectively [10, 11]. Our overall accuracy is comparable with the reported average accuracy of 98.8\% in a recent study that used Siamese network [35], however, that study was restricted to white blood cells, and it used only 430 images for testing and moreover, no visual saliency mappings is implemented for explainability.

<table>
<thead>
<tr>
<th>Study</th>
<th>Test Accuracy (%)</th>
<th>Training (N)</th>
<th>Validation (N)</th>
<th>Testing (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acevedo et.al., 2019 [3]</td>
<td>96.20</td>
<td>11077</td>
<td>4096</td>
<td>1919</td>
</tr>
<tr>
<td>Ucar, 2020 [9]</td>
<td>97.94</td>
<td>13674</td>
<td>1709</td>
<td>1709</td>
</tr>
<tr>
<td>Long et.al., 2021 [8]</td>
<td>99.30</td>
<td>13674</td>
<td>-</td>
<td>3418</td>
</tr>
<tr>
<td>Our study</td>
<td>97.21</td>
<td>1000</td>
<td>1000</td>
<td>15092</td>
</tr>
</tbody>
</table>

Going beyond the accuracy values for comparison of current studies with previous works, and even though we have used 20000 generated image pairs, they are obtained just using 125 images from each class and yet achieved comparable test performance metrics both while classifying all cell types and white blood cells exclusively. After extensive few-shot validation, hyperparameters set based on the Adadelta optimizer performed well and in addition, it can essentially eliminate learning rate selection by fixing it to one. Further, as expected, the 8-way \( k \)-shot validation and testing values are better when \( k \) is greater than 2 since the greater
number of images per class in the support set could lead to greater classification performance. To our knowledge, none of the previous studies listed in Table 3 based on traditional CNN based classification had explored the class activation mapping using either novel methods or already existing schemes such as CAM [24], gradCAM [25], and gradCAM++ [36]. Our proposed saliency mapping method is simple as it does not require gradients computation unlike gradCAM. Since we adapted the $a_d$ tensor values as the weights to be multiplied with the final convolution layer activation maps of the query image, explicit retraining of the model with the query image and the support set is not essential. Moreover, from Figures (3), (4), and (5), we could use the proposed saliency mapping method for cell localization in the image, and eventually, the framework may be adapted for coarse level cell segmentation. In the future, the proposed framework could be tested on blood cell images of disease conditions such as anemia and leukemia.

**Conclusions**

We proposed a contrastive learning based approach that relatively require fewer samples for training with finetuned EfficientNet as the base model for classification of peripheral blood cell images with the overall few-shot test accuracy of 97.21%. We employed state-of-the-art EfficientNet-B3 deep learning model and finetuned the last ten percent layers to learn more representative features of blood cells. In our 8-way few-shot experiments, the Adadelta optimizer with the choice of other hyperparameters provided slightly better results with the highest accuracy being 99.8% for classifying platelets. Three-shot and 5-shot testing accuracies are superior to 1-shot and 2-shot testing. Further, we demonstrated the creation of visual saliency maps without computing the gradients to interpret the decision of the Siamese network. Hence, our procedure may facilitate as a reference for deep CL based peripheral blood cell image classification against which the future research can be compared. In conclusion, we believe that our methodology could facilitate to development of deep learning based methods for classification of peripheral blood cells with fewer data samples and with interpretability.

**Declarations**

**Competing Interests:**

The authors have no competing interests to declare.

**Funding:**

No funding was received for conducting this study.

**Compliance with Ethical Standards**

This research study was conducted retrospectively using human subject data made available in open access by Hospital Clinic Barcelona. Ethical approval was not required as confirmed by the license attached with the open access data.

**Acknowledgements**
We would like to thank Hospital Clinic Barcelona for providing the dataset and SRM University—AP for providing the research infrastructure.

References


33. Dozat T (2015) Incorporating Nesterov Momentum into Adam

34. Zeiler MD (2012) ADADELTA: An Adaptive Learning Rate Method


Figures
Figure 1

Microscopy images of different blood cells as per their class randomly extracted from the clinical dataset. a) basophil, b) eosinophil, c) neutrophil, d) lymphocyte, e) monocyte, f) immature granulocyte, g) erythroblast, and h) platelet.
Figure 2

Proposed Siamese twin network architecture with EfficientNet-B3 as the base model for contrastive training. $\phi(X_1)$ is the embedding of image $X_1$ and $\phi(X_2)$ is the embedding of image $X_2$. GAP: global average pooling, $\sigma$: sigmoid neuron.
Figure 3

Reshaped values of absolute differences between the embeddings \((ad)\) and most activated regions during one-shot testing of a basophil query image. These regions are overlaid onto the images in the support set. The differences are lower when the basophil query image is compared with the basophil support image and the differences are higher when the query image is compared with non-basophil images in the support set. a: basophil, b: eosinophil, c: immature granulocyte, d: erythroblast, e: lymphocyte, f: monocyte, g: platelet, h: neutrophil.
Figure 4

Reshaped values of absolute differences between the embeddings ($ad$) and most discriminative regions during one-shot testing of an erythroblast query image. The discriminative regions are overlaid onto the images in the support set. The differences are lower when the erythroblast query image is compared with the erythroblast support image and differences are higher when the query image is compared with non-basophil images in the support set. a: basophil, b: eosinophil, c: immature granulocyte, d: erythroblast, e: lymphocyte, f: monocyte, g: platelet, h: neutrophil.
Figure 5

Query cell image and the corresponding same cell image in the support set with highlighted regions that contributed to the most similarity between the two images during one-shot testing. a: basophil, b: eosinophil, c: immature granulocyte, d: erythroblast, e: lymphocyte, f: monocyte, g: platelet, h: neutrophil.
Figure 6

Confusion matrices with accuracy values along the diagonal during 8-way few-shot testing on 15092 samples. Off diagonal values indicate percent of misclassifications. Top row is for 1-shot and 2-shot testing and bottom row is for 3-shot, and 5-shot testing.