

Classification of Poikilocytes Using ResNet18 and MobileNetV2 Optimized by a Multi-Objective Approach (Pareto Front and TOPSIS)

Patrice Kouamé Koffi¹, Sié Ouattara^{2,3} , Joseph Wognin Vangah¹, Alain Clement⁴, Nango Jérôme Alico⁵

¹Digital Research and Expertise Unit (UREN), Virtual University of Côte d'Ivoire (UVCI), Abidjan, Côte d'Ivoire

²Institut National Polytechnique Houphouët (INPHB), Yamoussoukro, Côte d'Ivoire

³School of Geomatics and Territory (EGT), Abidjan, Côte d'Ivoire

⁴LARIS, SFR Mathstic, University of Angers, Angers, France

⁵Preparatory Class for Grandes Écoles (CPGE), University of San Pédro (USP), San Pédro, Côte d'Ivoire

Email: sie_ouat@yahoo.fr

How to cite this paper: Koffi, P.K., Ouattara, S., Vangah, J.W., Clement, A. and Alico, N.J. (2026) Classification of Poikilocytes Using ResNet18 and MobileNetV2 Optimized by a Multi-Objective Approach (Pareto Front and TOPSIS). *Advances in Bioscience and Biotechnology*, 17, 586-613. <https://doi.org/10.4236/abb.2026.1710034>

Received: August 17, 2026

Accepted: October 13, 2026

Published: October 16, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Morphological and colorimetric analysis of red blood cells from blood smears constitutes an essential step of hematological diagnosis, yet it remains a lengthy and subjective task that is strongly dependent on the observer's expertise. To address these limitations, we propose an automated approach combining two (02) pre-trained deep convolutional neural networks, ResNet18 and MobileNetV2, for the classification of five (05) pathological erythrocyte morphologies (acanthocytes, drepanocytes, elliptocytes, ovalocytes and echinocytes). The database used comprises 560 red-blood-cell image patches extracted from blood smears, pre-processed and resized to $224 \times 224 \times 3$ pixels. Training relies on a 5-fold stratified cross-validation and two optimizers, SGDM and Adam, whose hyperparameters (number of epochs, learning rate, batch size) are selected through a grid search followed by a multi-objective optimization combining the Pareto Front and the TOPSIS method on a seven (07)-criterion vector integrating accuracy (F1, MCC, AUC), generalization gaps (GapAcc, GapLoss) and computational cost (number of parameters, latency). The results show that the four retained configurations achieve high and broadly comparable performance ($F1 > 0.95$; $MCC > 0.93$; $AUC > 0.995$); moreover, the model-optimizer pairs (ResNet18, Adam) and (MobileNetV2, Adam) provide the best overall compromise ($F1 \approx 0.966$). Beyond the above performance, ResNet18 offers the best generalization robustness, while MobileNetV2, with

nearly five times fewer parameters, constitutes a relevant alternative for resource-constrained environments, although its latency measured in practice remains, counter-intuitively, higher than that of ResNet18. These results confirm the value of transfer learning and multi-objective optimization for developing robust, interpretable and reproducible erythrocyte classification systems, while identifying additional validation avenues (per-class confusion matrix, statistical tests) that are necessary before clinical deployment.

Keywords

Red Blood Cell Classification, ResNet18, MobileNetV2, Deep Learning, Optimizers, SGDM, Adam, 5-Fold Stratified Cross-Validation, Multi-Objective Optimization, Pareto Front, TOPSIS

1. Introduction

Morphological analysis of red blood cells by means of blood smears is an essential step of hematological diagnosis. Certain anemias and pathologies result in visible alterations of the shape of red blood cells, such as echinocytes, acanthocytes, elliptocytes, drepanocytes and ovalocytes. These clinically significant morphologies point to a variety of conditions that are difficult to distinguish by microscopic examination alone. Manual identification remains lengthy, subjective and strongly dependent on the observer's expertise, while also being affected by smear quality, staining artifacts and the morphological proximity between certain classes.

These limitations motivate automated approaches to red blood cell isolation and classification, which can reduce inter-observer variability and make hematological expertise more accessible. Among them, modern convolutional neural architectures, such as ResNet18 and MobileNetV2, have demonstrated their effectiveness in biomedical image classification. Their use, combined with transfer learning techniques and rigorous validation strategies, constitutes a promising avenue for improving the detection and classification of erythrocyte abnormalities. The objective of this work is to propose and evaluate an integrated approach combining pre-trained convolutional neural networks and multi-objective hyperparameter optimization, in order to develop a robust, interpretable and reproducible system for digital hematology.

The contributions of this work are threefold. First, we propose a complete pipeline for isolating and classifying the five main pathological erythrocyte morphologies, based on transfer learning using ResNet18 and MobileNetV2. Second, we introduce a multi-objective hyperparameter optimization strategy combining grid search, the Pareto Front and the TOPSIS method on a seven-criterion vector integrating both classification performance (F1, MCC, AUC) and computational cost (number of parameters, latency, generalization gaps). Third, we provide a systematic comparison of four architecture-optimizer configurations (ResNet18/Mo-

bileNetV2 $\times$ SGDM/Adam), in order to objectify the trade-off between accuracy and computational efficiency for deployment in a resource-limited environment.

The remainder of this article is organized as follows: Section 2 presents a synthesis of the state of the art on the segmentation and classification of red blood cells; Section 3 details the material, the experimental protocol and the multi-objective optimization methodology; Section 4 presents the isolation and classification method implemented, then discusses the results obtained; finally, Section 5 concludes this work and proposes research perspectives.

2. Related Works

Morphological analysis of red blood cells on blood smears remains a fundamental step of hematological diagnosis [1]. Certain anemias result in visible alterations of the shape of red blood cells, among which acanthocytes, echinocytes, drepanocytes, elliptocytes and ovalocytes can notably be distinguished [2] [3]. These morphologies are clinically significant, as they point to various conditions that are sometimes difficult to distinguish by microscopic examination alone [3]. However, their manual identification remains lengthy, subjective and strongly dependent on the observer's expertise [4]. It is further affected by smear quality, staining artifacts, cell overlaps and the morphological proximity between certain classes, such as acanthocytes and echinocytes, or elliptocytes and ovalocytes [1]-[3].

In this context, automating the isolation and classification of red blood cells appears to be a relevant solution for reducing inter-observer variability, speeding up analysis and making hematological expertise more accessible in resource-limited environments [1] [4]. Existing approaches fall into several main families: classical segmentation and classification methods, hybrid approaches combining segmentation and machine learning, and recent deep-learning-based methods [1].

Traditional segmentation methods historically constitute the basis of red blood cell isolation [1]. Otsu's method relies on maximizing inter-class variance and remains a reference for global thresholding [5]. However, in real blood smears, intensity variations, staining heterogeneity and background noise often limit its performance [5]. For this reason, multi-threshold variants optimized by metaheuristics have been proposed to improve the separation between cells and background [5]. Local thresholding methods, such as Niblack, Sauvola or Bernsen, adapt the threshold according to the neighborhood and offer better robustness to local lighting variations [6]-[8]. Likewise, thresholding in color spaces, notably RGB and HSV, exploits the chromatic properties of red blood cells to strengthen segmentation [1]. Morphological operations, such as opening, closing and the morphological gradient, are then used to remove artifacts and correct certain binarization imperfections [9]. Finally, the Watershed algorithm, often combined with the distance transform and morphological markers, remains one of the most effective techniques for separating cells that are in contact or partially overlapping [9].

Hybrid approaches combine classical segmentation and machine learning in

order to leverage the simplicity of traditional methods while adding a higher discrimination capacity [3]. Some strategies rely on an initial segmentation of red blood cells, followed by the extraction of morphological parameters such as area, perimeter, circularity, eccentricity or convexity, before classification using algorithms such as SVM or random forests [10] [11]. These methods have the advantage of being relatively interpretable and effective on modest-sized datasets [3] [10] [11]. They nevertheless remain sensitive to the quality of the initial segmentation, which can limit their robustness in the presence of overlaps, artifacts or poor-quality slides [3].

Deep-learning-based automatic segmentation methods represent a major advance in blood-cell image analysis [1]. U-Net-type architectures and their variants have become reference approaches for semantic segmentation in medical imaging [12]. Their ability to learn multi-scale representations enables them to better handle irregular boundaries and complex shapes [12]. For instance segmentation, Mask R-CNN provides a robust solution for detecting and separating individual cells [13]. More recent approaches based on hybrid CNN-Transformer architectures seek to combine the local feature-extraction capabilities of convolutional networks with the global dependency modeling offered by Transformers [14]. Specialized tools such as CellPose, StarDist, and CellViT have also demonstrated strong potential for segmenting cells with diverse shapes and handling overlapping objects [14]-[16]. This methodological evolution is particularly relevant to red blood cells, whose morphology can vary considerably according to the pathological condition and slide-preparation conditions [1] [2] [15].

Morphological classification by deep learning strongly depends on the quality of the dataset used [1]. Classes must be well defined, sufficiently represented and ideally balanced [1] [17]. However, in medical applications, annotated data are often scarce, costly to produce and imbalanced [17]. This is why data augmentation, oversampling and loss-weighting techniques are frequently used to improve generalization [18]. Residual architectures, such as ResNet-18 and ResNet-50, as well as lighter models such as MobileNetV2, benefit from transfer learning from generic datasets such as ImageNet [19]-[21]. DenseNet121 stands out for its dense connections, which reduce feature redundancy, while some lighter CNNs can achieve better performance than deeper architectures on small datasets [22]. The most recent trends are moving towards hybrid models combining detection, segmentation and classification within a single architecture, in order to simultaneously handle cell localization and morphology identification [1] [14] [16].

Another essential aspect concerns dataset splitting and the validation strategy [1]. Fixed 70/15/15 or 80/10/10 splits are simple to implement, but they can produce unstable estimates when data are limited [1]. K-fold cross-validation, in particular stratified 10-fold, is generally more robust because it reduces estimation variance and allows a more reliable performance evaluation [1]. LOOCV provides an almost unbiased estimate, but its computational cost makes it poorly suited to

deep networks [1]. In medical deep learning, the choice between holdout, k-fold and cross-validation therefore depends on the trade-off between statistical stability and computational feasibility [1].

Hyperparameter optimization also plays a decisive role in the final quality of the model [23]. Among the most sensitive parameters are the learning rate, the batch size and the number of epochs [23]. The most commonly used search methods are Grid Search, Random Search and Bayesian optimization [23] [24]. The learning rate often remains the critical parameter, as a value that is too high can prevent convergence, while a value that is too low slows down learning [23]. Optimizers such as Adam, SGD with momentum and AdamW are frequently used depending on the size of the dataset and the complexity of the model [25] [26]. Finally, regularization techniques such as Dropout, L2 regularization and Batch Normalization help limit overfitting and improve generalization [27] [28].

In summary, the state of the art shows a clear evolution from classical segmentation methods towards increasingly automated pipelines, and then towards deep-learning solutions capable of integrating segmentation, detection and classification [1] [14]. For the isolation of red blood cells and their morphological classification, hybrid approaches and deep architectures currently appear to be the most promising, provided that sufficiently well-annotated data, rigorous validation and an appropriate optimization strategy are available [1] [17] [23]. The main challenge remains to develop robust, interpretable and reproducible systems capable of operating reliably on real, often noisy and heterogeneous smears [1] [2].

3. Materials and Methods

The experiments were conducted on color microscopic images of blood smears, from which an image-patch dataset was constructed using the improved algorithm of Alico *et al.* (2020) [29]. The dataset comprises five (05) classes of abnormal red blood cells: acanthocytes, drepanocytes, elliptocytes, ovalocytes, and echinocytes. The image patches were preprocessed and then used to train the ResNet18 and MobileNetV2 convolutional neural networks under 5-fold stratified cross-validation. Hyperparameter selection was based on a multi-objective vector optimized using the Pareto Front and TOPSIS. **Figure 1** presents the flowchart of the adopted processing pipeline.

3.1. Data Provenance and Reference Standard

Data provenance. The blood-smear images used to construct the dataset were provided by the Blood Transfusion Center (CTS) and the Regional Hospital Center (CHR) of Yamoussoukro, with an additional occasional contribution from the CTS of Abidjan. The authors thank Dr. D. Bosso (CTS Yamoussoukro), Dr. C. Houedanou (CHR Yamoussoukro), K. Hermann (CHR Yamoussoukro), Dr. B. Dembélé (CTS Abidjan), K. Eugène (CHR Yamoussoukro), and T. Abalé (CTS Abidjan) for their collaboration in the production of this work.

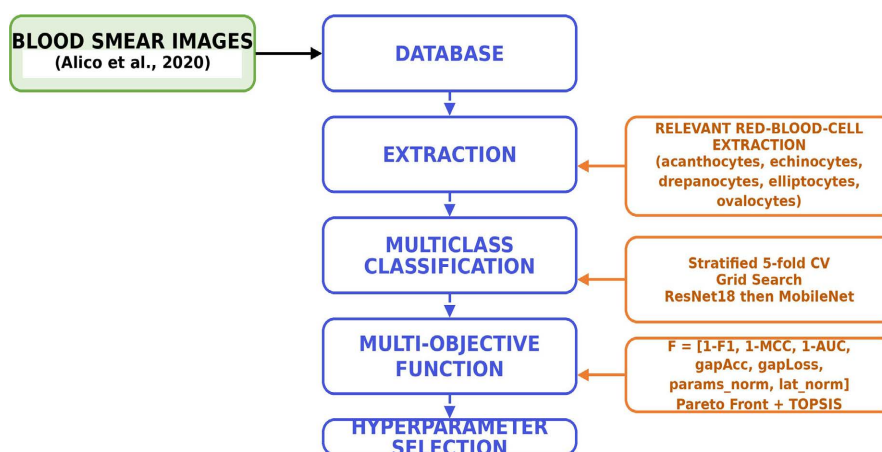


Figure 1. Flowchart of the blood-smear red-blood-cell classification process.

Ethical status. Data collection was carried out as part of a previous collaboration of the research group, with authorization from the CTS and CHR laboratories in Yamoussoukro to use blood-smear images for research purposes. This authorization was granted institutionally by the partner laboratories; it does not constitute formal approval by an independent ethics committee, and no record of individual patient consent was available to the authors of the present study. The images used contain no nominative or identifying information that would allow them to be traced back to an individual patient.

Acknowledged limitation—patient/smear traceability and selection criteria. The smear images were provided to the authors as an already assembled collection, without information on the number of distinct patients from whom the 470 smears originated or on the criteria used to select these smears from all examinations performed by the partner institutions.

Diagnostic reference and adjudication process. Assignment of each image patch to one of the five morphological classes followed a three-level hierarchical procedure: first, a literature review was used to establish the reference morphological criteria for each class; laboratory technicians then annotated the smear images according to these criteria; the annotations were subsequently confirmed or rejected by biological-analysis engineers; finally, hematology laboratory physicians adjudicated the remaining disputed cases. Cells for which disagreement or ambiguity persisted after this procedure were removed from the dataset rather than assigned a default label—this exclusion concerned only a marginal number of cells (approximately 20).

Annotator qualifications. The annotation team consisted of two laboratory physicians specialized in hematology, two biological-analysis engineers from the hematology department, and three laboratory technicians.

Classifier scope. The classifier developed in this study operates exclusively on pre-isolated single-cell image patches and recognizes only the five morphological classes included in the training data. A cell exhibiting a pathological morphology outside these five classes would, in the current version of the system, necessarily

be incorrectly assigned to one of the five known classes; accordingly, the clinical-diagnostic claims made in this manuscript are restricted to this scope.

3.2. Dataset and Acquisition Equipment

The raw image consists of 470 color BMP smear images with a resolution of 1024×768 pixels, acquired without Giemsa staining. Images were captured at $1000\times$ total magnification using a $100\times$ objective lens and a $10\times$ eyepiece. Illumination was provided by a 6V-220W halogen lamp. Acquisition was performed with a Moticam2 2 MP color CMOS camera, and images were transferred to a computer via USB 3.0.

The improved cell-isolation algorithm of Alico *et al.* (2020) [29] was applied to generate the image patches composing the dataset. **Figure 2** shows an example of a blood-smear image, and **Figure 3** presents examples of image patches after cell isolation.

Red blood cell extraction was performed in MATLAB on an HP computer with an Intel i5-3230M CPU, 12 GB RAM, and a 477 GB SSD. Model training was conducted on an MSI workstation with a Core i7 processor and an NVIDIA GTX 1070 GPU.

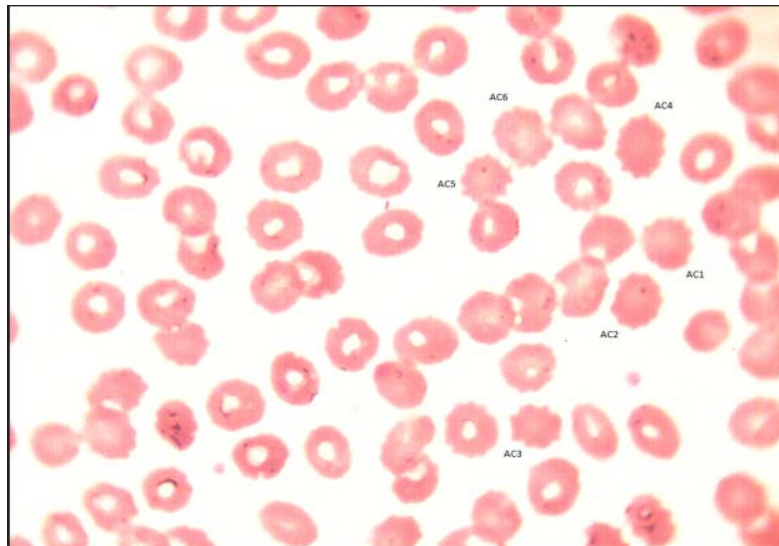


Figure 2. Example blood-smear image obtained from the acquisition protocol.

Furthermore, the distribution of our dataset by morphological class is as follows: echinocytes (86 images), acanthocytes (92 images), elliptocytes (138 images), drepanocytes (106 images) and ovalocytes (138 images), for a total of 560 image patches. To ensure a balanced representation of each class during model evaluation, a 5-fold stratified cross-validation was applied, preserving the relative proportions of the five classes in each fold. This approach guarantees that the minority classes (echinocytes and acanthocytes) are systematically present in the training and validation subsets, see **Table 1**.

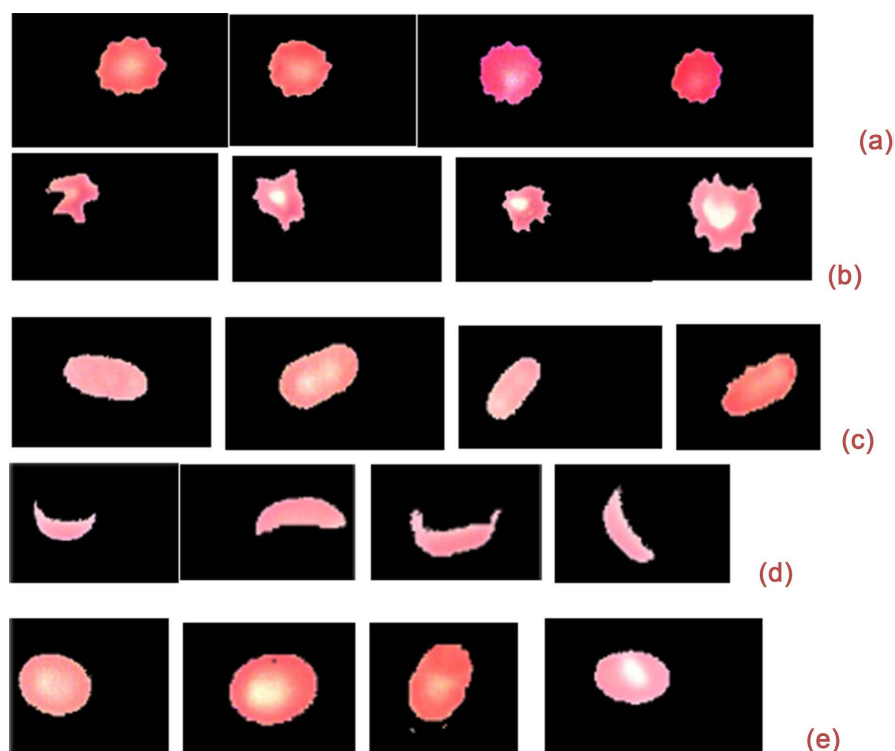


Figure 3. Red blood cells isolated from blood-smear images (Alico *et al.*, 2020 [29]). (a): Echinocytes, (b): Acanthocytes, (c): Elliptocytes, (d): Drepanocytes, (e): Ovalocytes.

Table 1. Distribution of image patches by class and by cross-validation fold.

RBC class	Total	Validation per fold	Training per fold
Echinocytes	86	≈17	≈69
Acanthocytes	92	≈18	≈74
Elliptocytes	138	≈28	≈110
Drepanocytes	106	≈21	≈85
Ovalocytes	138	≈28	≈110

3.3. Preprocessing

The acquired images comprise several red blood cells at once, with different morphological and colorimetric shades, and these occupy on average a memory space of 2.305 MB. First, pre-processing was carried out, consisting of extracting each cell from its surroundings, without resizing the original image. The dataset obtained consists of individual red blood cells, called image patches, showing various morphologies, distributed among the five (05) classes retained. Next, in order to reduce the number of parameters of the models to be estimated, the image patches were resized to $224 \times 224 \times 3$ pixels, in accordance with the input data requirements of the ResNet18 and MobileNetV2 models. Finally, each image was labeled according to its class for model training.

3.4. CNN Architecture

Automatic classification of pathological red blood cells is carried out using two deep convolutional neural architectures: ResNet18 and MobileNetV2. This choice is based on the relatively limited size of the database (560 images distributed across five morphological classes), the need to obtain good generalization capacity, and the use of transfer learning from models pre-trained on ImageNet.

Indeed, ResNet18 (Residual Network with 18 weighted layers), proposed by He *et al.* (2016), introduces the concept of residual connections, allowing deep networks to learn residual functions rather than complete transformations. This innovation facilitates gradient propagation and limits the performance degradation observed as network depth increases. With approximately 11.7 million parameters, ResNet18 offers an attractive trade-off between representation capacity, computational cost and the risk of overfitting on a limited database. The overall architecture of ResNet18 used in this study is shown in **Figure 4**.

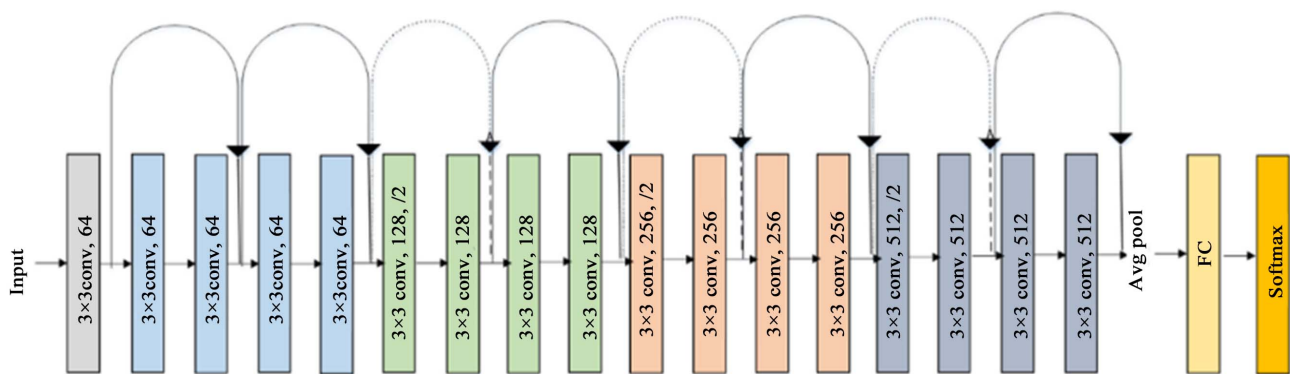


Figure 4. ResNet18 architecture.

As for MobileNetV2, it was developed by Sandler *et al.* (2018) and designed to offer excellent computational efficiency while maintaining good classification performance. Its architecture relies on inverted residual blocks combined with linear bottlenecks and depthwise separable convolutions, which considerably reduces the number of parameters and the cost of operations. With approximately 3.5 million parameters, MobileNetV2 constitutes a compact architecture, favoring reduced training and inference times. The corresponding MobileNetV2 architecture is illustrated in **Figure 5**.

The choice of these two architectures is motivated by their complementarity: ResNet18 favors a deep representation of morphological features thanks to its residual connections, while MobileNetV2 relies on an efficient representation based on a lightweight architecture. Since few studies have jointly evaluated these two architectures on the simultaneous classification of the five erythrocyte morphologies considered here, this comparison appears relevant. Before training, all images are resized to $224 \times 224 \times 3$ pixels; the features extracted by the convolutional layers are passed to a classification layer whose outputs are converted into proba-

bilities by the Softmax function.

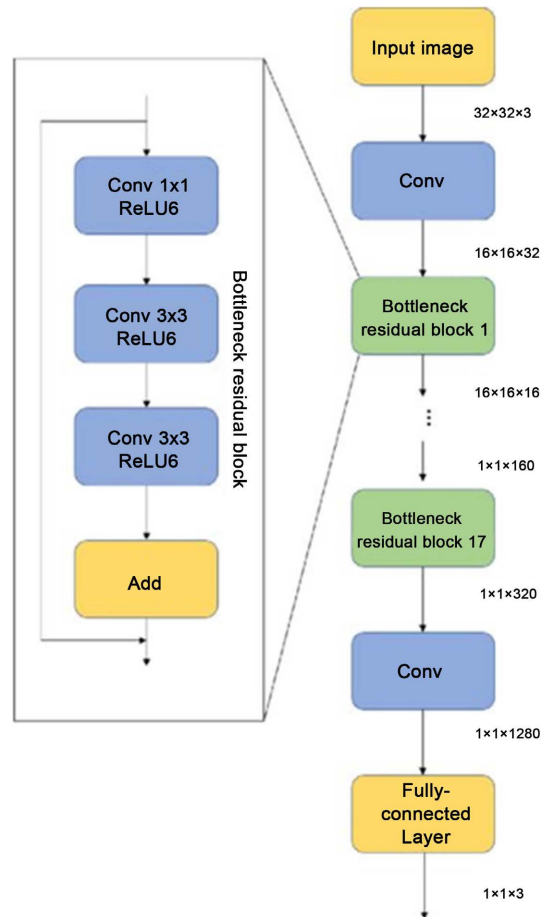


Figure 5. MobileNetV2 architecture.

3.5. Transfer Learning

The limited availability of large annotated databases constitutes one of the main difficulties of automatic classification of pathological red blood cells. Since our database comprises only 560 images patches distributed across five classes, training from a random initialization would expose the deep architectures to a high risk of overfitting. We therefore resort to transfer learning, which consists of re-using the representations learned by a deep network on a large-scale source task (ImageNet, approximately 1.28 million images distributed across 1000 categories) for a target task with few examples.

The adopted process comprises five steps: 1) loading the ResNet18 and MobileNetV2 weights pre-trained on ImageNet; 2) replacing the final classifier (1000 categories) with a new five-neuron layer corresponding to the five morphological classes studied; 3) feature extraction, where the pre-trained convolutional layers are frozen and only the new classification layers are trained; 4) fine-tuning, where part of the last blocks is progressively unfrozen to adapt the deep representations to erythrocyte morphologies; 5) final classification via the Softmax function. This

principle is applied identically to both architectures, with the only difference being the nature of the blocks unfrozen during fine-tuning (residual blocks for ResNet18, inverted residual blocks for MobileNetV2).

3.6. Hyperparameter Optimization

Optimization of the training hyperparameters—learning rate, number of epochs and mini-batch size—is carried out by grid search. The values tested are the batch size (12, 16, 24, 28, 32) and the learning rate (10^{-4} , $3 \cdot 10^{-4}$, $5 \cdot 10^{-4}$, 10^{-3}), *i.e.* 20 combinations per architecture-optimizer pair. The number of epochs is not fixed manually: the 5-fold stratified cross-validation incorporates an early-stopping mechanism, which halts training as soon as validation performance stops improving, thus limiting overfitting. For $k = 5$, each configuration in the table produces 5 training runs, *i.e.* 100 training runs per architecture-optimizer pair (5×20), and 400 training runs in total across the four configurations studied (ResNet18/MobileNetV2 $\times$ SGDM/Adam).

Two optimizers are compared. Stochastic gradient descent with momentum (SGDM) introduces an inertia term that dampens gradient-descent oscillations and accelerates convergence (Figure 6). The Adam optimizer (Adaptive Moment Estimation) computes an adaptive learning-rate step for each parameter from exponential moving averages of the first and second moments of the gradient, with bias correction.



Figure 6. Damping of stochastic-gradient-descent oscillations by the momentum method (SGDM).

Formulation of the Adam optimizer (first- and second-order moments, bias correction)

Adam [26], is a method that computes a learning-rate step described as adaptive for each parameter. During training, in addition to storing an exponentially decaying average of the squares of the previous gradients as with RMSprop, the algorithm also keeps the exponentially decaying average of the previous gradients according to the equations:

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

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

It should be noted that m_t updates the exponential moving average of the gradient and is referred to as the first-order (“1”) moment estimate. As for v_t , it updates the exponential moving average of the squared gradients and is referred

to as the second-order (“2”) moment estimate. According to the authors, m_t and v_t are initialized as vectors of “0” and are biased towards this point during the initial steps, particularly when the decay coefficients (β_1, β_2) are small (close to “1”). They propose a bias correction for the estimates using the following relations:

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

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

Then comes the parameter update (θ):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \cdot m_t \quad (5)$$

- η , the learning rate,
- β_1 and β_2 , damping coefficients (typically close to “1”, for example, 0.9 and 0.999),
- ε , a small constant added to avoid division by zero.

For the parameters β_1 , β_2 and ε , the authors propose default values of 0.9, 0.999, and 10^{-8} .

The advantages provided by *Adam*, make it the most widely used optimizer in neural network training.

3.6.1. Multi-Objective Vector

Each hyperparameter configuration x = (Epoch, Learning Rate, Batch Size) is evaluated using seven criteria representing both classification performance and model complexity:

$$f(x) = (1 - F1, 1 - MCC, 1 - AUC, GapAcc, GapLoss, Params_norm, Latency_norm) \quad (6)$$

- 1 – F1: error associated with the F1-score (macro average over the five classes);
- 1 – MCC: error associated with the Matthews correlation coefficient;
- 1 – AUC: error associated with the area under the ROC curve (macro, one-vs-rest average);
- GapAcc: gap between training and validation accuracy;
- GapLoss: gap between training and validation loss;
- Params_norm: number of model parameters, normalized by the maximum observed;
- Latency_norm: inference latency, normalized by the maximum observed.

As all components are formulated as minimization criteria, the problem is written as: minimize $f(x)$.

3.6.2. Pareto Front and TOPSIS Method

In multi-criteria optimization, a configuration A dominates a configuration B ($A \leq B$) if $f_i(A) \leq f_i(B)$ for all criteria i , and if there exists at least one criterion j such that $f_j(A) < f_j(B)$. The non-dominated solutions constitute the Pareto Front, which

reduces the search space to the optimal trade-offs only.

The Pareto Front identifies optimal solutions but does not allow them to be ranked. The TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method ranks these solutions using two fictitious references: an ideal solution (best values observed for each criterion) and an anti-ideal solution (worst values observed). For each configuration, the Euclidean distance D^+ to the ideal solution and D^- to the anti-ideal solution are computed, then the closeness coefficient $C_i = D^-/(D^+ + D^-)$, with $0 \leq C_i \leq 1$. The closer C_i is to 1, the more the configuration is simultaneously close to the ideal and far from the anti-ideal. The Pareto Front thus acts as a qualitative filter, and TOPSIS as a geometric quantifier used to designate the best configuration within each search block.

3.7. Evaluation Metrics

Classification performance is evaluated from the multi-class confusion matrix, aggregated over the 5 cross-validation folds, using the following metrics:

- F1-score (macro):

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

It is averaged over the five classes (computed from the cumulative confusion matrix of the 5 folds). Assesses the precision/recall balance, giving equal weight to each class, even minority ones.

MCC (Matthews Correlation Coefficient):

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (8)$$

Measures the model's overall consistency, robust to class imbalance.

- AUC: area under the ROC curve

$$AUC = \int_0^1 TPR(FPR) d(FPR) \quad (9)$$

where $TPR = TP/(TP + FN)$ and $FPR = FP/(FP + TN)$.

Computed from the cumulative confusion matrix of the 5 folds. Indicates the model's ability to correctly separate classes, independently of the chosen threshold.

$$\text{GapAcc} = |\text{Accuracy}_{\text{training}} - \text{Accuracy}_{\text{validation}}| \quad (10)$$

Overfitting indicator based on accuracy.

$$\text{GapLoss} = |\text{Loss}_{\text{training}} - \text{Loss}_{\text{validation}}| \quad (11)$$

overfitting indicator based on the loss function.

$$\text{Params}_{\text{norm}} = \text{Nb}_{\text{model_parameters}} / \text{Nb}_{\text{max_parameters}} \quad (12)$$

Number of model parameters normalized with respect to a fixed reference ceiling of 12×10^6 , selected as a safety margin above the parameter count of the largest architecture evaluated in this study (ResNet18, ≈ 11.18 million parameters after adaptation to a five-class output layer).

$$\text{Latency_norm} = \text{Model_latency}/\text{Max_latency} \quad (13)$$

Inference time normalized relative to the slowest configuration tested.

The selection of the optimal configurations combines the Pareto Front, which retains the non-dominated solutions, and the TOPSIS method, which ranks them according to their proximity to the ideal solution, in order to identify the best trade-off between accuracy, generalization capacity and computational cost.

4. Results and Discussion

4.1. Experimental Setup

The experiments are conducted on the 560 image patches described in Section 3.1, using a 5-fold stratified cross-validation applied independently to each of the four architecture-optimizer configurations (ResNet18/MobileNetV2 $\times$ SGDM/Adam). For each configuration, a grid search of 20 combinations (5 batch sizes $\times$ 4 learning rates) is evaluated, *i.e.* 100 training runs per configuration and 400 training runs in total. The number of epochs is determined automatically by early stopping. Each combination is summarized by the seven-component multi-objective vector defined in Section 3.5, then filtered by the Pareto Front and ranked by TOPSIS in order to identify, for each learning-rate value, the best (epoch, batch size) combination, before a final selection among the four retained candidates.

4.2. Results

4.2.1. Hyperparameter Optimization via Grid Search

Tables 2-5 present, for each of the four configurations, the best (epoch, batch size) combination retained by the Pareto Front + TOPSIS for each of the four learning-rate values tested, along with the associated metrics. For the sake of readability, the 20 raw training runs per configuration (100 per architecture-optimizer pair in total) are not fully reproduced here; they are available as supplementary material.

4.2.2. Comparison of the Optimal Configurations

Table 6 gathers, for the four configurations ultimately retained following the TOPSIS selection, the values converted into directly interpretable metrics as well as the complexity and generalization indicators.

Table 2. ResNet18 + SGDM—best configuration for each learning-rate value (Pareto Front + TOPSIS). Selected configuration: Epoch = 25, LR = 10^{-4} , BS = 24.

Learning rate	Epoch*	Mini-batch size*	F1	MCC	AUC	GapAcc	GapLoss	Pareto status	Ci
1.10^{-4}	25	24	0.9626	0.9533	0.9975	0.0188	0.0405	Non-dominated	0.7144
3.10^{-4}	27	32	0.9665	0.9580	0.9962	0.0170	0.0792	Dominated	0.2258
5.10^{-4}	20	32	0.9682	0.9607	0.9969	0.0192	0.0743	Non-dominated	0.3820
1.10^{-3}	18	24	0.9667	0.9585	0.9967	0.0165	0.0783	Non-dominated	0.3124

Table 3. ResNet18 + Adam—best configuration for each learning-rate value. Selected configuration: Epoch = 14, LR = 10^{-4} , BS = 32.

Learning rate	Epoch*	Mini-batch size*	F1	MCC	AUC	GapAcc	GapLoss	Pareto status	Ci
1.10^{-4}	14	32	0.9659	0.9574	0.9970	0.0210	0.0907	Non-dominated	0.9957
3.10^{-4}	14	32	0.9386	0.9251	0.9933	0.0246	0.1013	Non-dominated	0.5304
5.10^{-4}	17	28	0.9378	0.9223	0.9954	0.0259	0.0908	Dominated	0.6163
1.10^{-3}	16	28	0.9181	0.8984	0.9882	0.0313	0.1197	Dominated	0.0063

Table 4. MobileNetV2 + SGDM—best configuration for each learning-rate value. Selected configuration: Epoch = 40, LR = 10^{-4} , BS = 28.

Learning rate	Epoch*	Mini-batch size*	F1	MCC	AUC	GapAcc	GapLoss	Pareto status	Ci
1.10^{-4}	40	28	0.9512	0.9384	0.9968	0.0241	0.0458	Non-dominated	0.8420
3.10^{-4}	21	12	0.9527	0.9409	0.9973	0.0263	0.0770	Non-dominated	0.9995
5.10^{-4}	21	12	0.9559	0.9449	0.9964	0.0214	0.0823	Non-dominated	0.8430
1.10^{-3}	21	16	0.9472	0.9349	0.9960	0.0335	0.0872	Non-dominated	0.6830

Table 5. MobileNetV2 + Adam—best configuration for each learning-rate value. Selected configuration: Epoch = 16, LR = 3.10^{-4} , BS = 32.

Learning rate	Epoch*	Mini-batch size*	F1	MCC	AUC	GapAcc	GapLoss	Pareto status	Ci
1.10^{-4}	18	12	0.9476	0.9350	0.9957	0.0170	0.0894	Non-dominated	0.9856
3.10^{-4}	16	32	0.9659	0.9570	0.9954	0.0201	0.0623	Non-dominated	0.9202
5.10^{-4}	15	16	0.9182	0.8996	0.9891	0.0237	0.1253	Non-dominated	0.7569
1.10^{-3}	18	24	0.9172	0.8977	0.9914	0.0237	0.0659	Non-dominated	0.8255

Table 6. Comparison of the four optimal configurations (architecture $\times$ optimizer).

Configuration	Epoch	LR	BS	F1	MCC	AUC	GapAcc	GapLoss	Params_norm	Lat_norm	Pareto status	Ci
ResNet18 + SGDM	25	10^{-4}	24	0.9626	0.9533	0.9975	0.0188	0.0405	0.9320	0.1786	Non-dominated	0.9134
ResNet18 + Adam	14	10^{-4}	32	0.9659	0.9574	0.9970	0.0210	0.0907	0.9320	0.1774	Non-dominated	0.5307
MobileNetV2 + SGDM	40	10^{-4}	28	0.9512	0.9384	0.9968	0.0241	0.0458	0.1873	0.3254	Non-dominated	0.4913
MobileNetV2 + Adam	16	3.10^{-4}	32	0.9659	0.9570	0.9954	0.0201	0.0623	0.1873	0.3206	Non-dominated	0.4470

4.2.3. Cumulative Confusion Matrices (5 Folds)

In order to document the nature of the residual errors between morphologically close classes, the cumulative confusion matrix over the 5 cross-validation folds is presented below for each of the four optimal configurations retained (**Table 6**). Each matrix aggregates the validation predictions of the 5 folds; the percentages at the bottom of each column indicate the precision per predicted class (proportion of image patches correctly classified among those predicted in that class), the classes being coded ACH (Acanthocytes), DRE (Drepanocytes), ECH (Echinocytes), ELL (Elliptocytes) and OVA (Ovalocytes).

Confusion Matrix-Cumulee des 5 stratified folds

True Class	ACH	88	1	1		1	96.7%	3.3%
	DRE	1	105		1		98.1%	1.9%
	ECH	1		79		6	91.9%	8.1%
	ELL		2		133	3	96.4%	3.6%
	OVA			1	3	134	97.1%	2.9%

97.8%	97.2%	97.5%	97.1%	93.1%
2.2%	2.8%	2.5%	2.9%	6.9%
ACH	DRE	ECH	ELL	OVA

Predicted Class

Figure 7. Cumulative confusion matrix over 5 folds: Res-Net18 + SGDM (Epoch = 25, LR = 10^{-4} , BS = 24).

Confusion Matrix-Cumulee des 5 stratified folds

True Class	ACH	88		2		2	95.7%	4.3%
	DRE		105		1		99.1%	0.9%
	ECH	2		81		3	94.2%	5.8%
	ELL	1	1		133	3	96.4%	3.6%
	OVA			1	3	134	97.1%	2.9%

96.7%	99.1%	96.4%	97.1%	94.4%
3.3%	0.9%	3.6%	2.9%	5.6%
ACH	DRE	ECH	ELL	OVA

Predicted Class

Figure 8. Cumulative confusion matrix over 5 folds: Res-Net18 + Adam (Epoch = 14, LR = 10^{-4} , BS = 32).

Confusion Matrix-Cumulee des 5 stratified folds

True Class	ACH	87	2	2		1	94.%	5.4%
	DRE		102		1	3	96.2%	3.8%
	ECH	1		80		6	93.0%	7.0%
	ELL				129	9	93.5%	6.5%
	OVA				5	133	96.4%	3.6%

98.9%	98.1%	97.6%	95.6%	88.1%
1.1%	1.9%	2.4%	4.4%	11.9%
ACH	DRE	ECH	ELL	OVA

Predicted Class

Figure 9. Cumulative confusion matrix over 5 folds: MobileNetV2 + SGDM (Epoch = 40, LR = 10^{-4} , BS = 28).

Confusion Matrix-Cumulee des 5 stratified folds

True Class	ACH	87	1	2	1	1	94.6%	5.4%
	DRE		106				100.0%	
	ECH			82		4	95.3%	4.7%
	ELL		2		131	5	94.9%	5.1%
	OVA				4	134	97.1%	2.9%
		100.0%	97.2%	97.6%	96.3%	93.1%		
			2.8%	2.4%	3.7%	6.9%		
		ACH	DRE	ECH	ELL	OVA		
		Predicted Class						

Figure 10. Cumulative confusion matrix over 5 folds: MobileNetV2 + Adam (Epoch = 16, LR = 3.10^{-4} , BS = 32).

Table 7. Precision per predicted class (%), read at the bottom of the cumulative confusion matrices (Figures 7-10).

Configuration	ACH (%)	DRE (%)	ECH (%)	ELL (%)	OVA (%)
ResNet18 + SGDM	97.8	97.2	97.5	97.1	93.1
ResNet18 + Adam	96.7	99.1	96.4	97.1	94.4
MobileNetV2 + SGDM	98.9	98.1	97.6	95.6	88.1
MobileNetV2 + Adam	100.0	97.2	97.6	96.3	93.1

Reading **Figures 7-10** and **Table 7** confirms the hypotheses formulated in Section 2: the Ovalocytes (OVA) class systematically shows the lowest precision of the five classes, regardless of configuration (88.1% to 94.4%), which is consistent with the morphological proximity between elliptocytes and ovalocytes already reported in the literature. This effect is most pronounced for MobileNetV2 + SGDM (88.1%), a configuration for which the Ovalocytes class visibly concentrates the most prediction errors among the four configurations studied. Conversely, the Acanthocytes (ACH) and Drepanocytes (DRE) classes achieve a precision above 96.7% in all configurations, and up to 100% for Acanthocytes with MobileNetV2 + Adam, indicating that these morphologies are the most easily discriminated by both architectures. These observations qualify the ranking obtained on the global metrics (**Table 6**): two configurations displaying an almost identical global F1 (ResNet18 + Adam and MobileNetV2 + Adam, $F1 \approx 0.966$) may nevertheless show different per-class error profiles, a criterion to take into account depending on the targeted clinical use (for example, if reliable detection of ovalocytes is a priority).

4.2.4. Learning Curves (Representative Fold)

Figures 11-14 illustrate, for a representative fold among the five cross-validation folds, the evolution of accuracy and the loss function during training for each of the four optimal configurations. These curves make it possible to qualitatively ver-

ify the stability of convergence and the absence of divergence between the training and validation curves, complementing the GapAcc and GapLoss indicators reported in **Tables 2-6**.

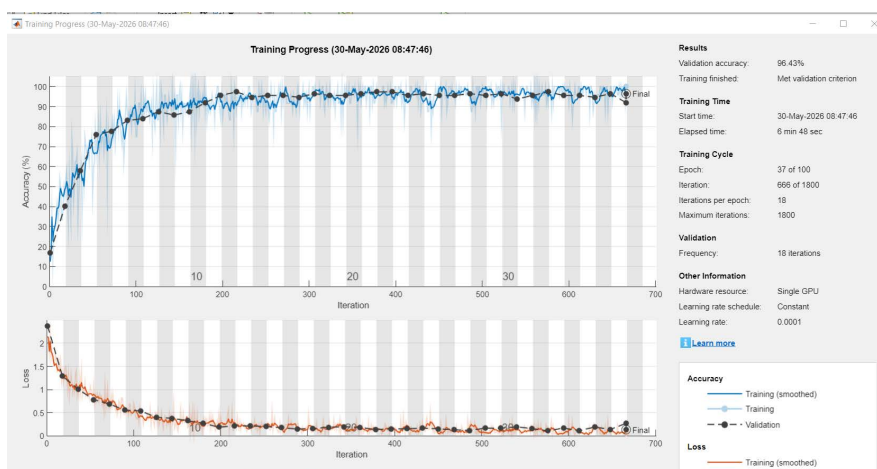


Figure 11. Learning curves (representative fold): ResNet18 + SGDM.

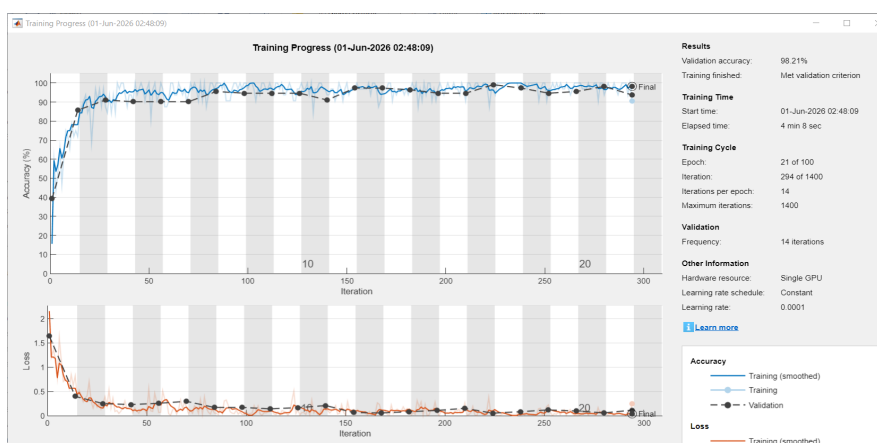


Figure 12. Learning curves (representative fold): ResNet18 + Adam.

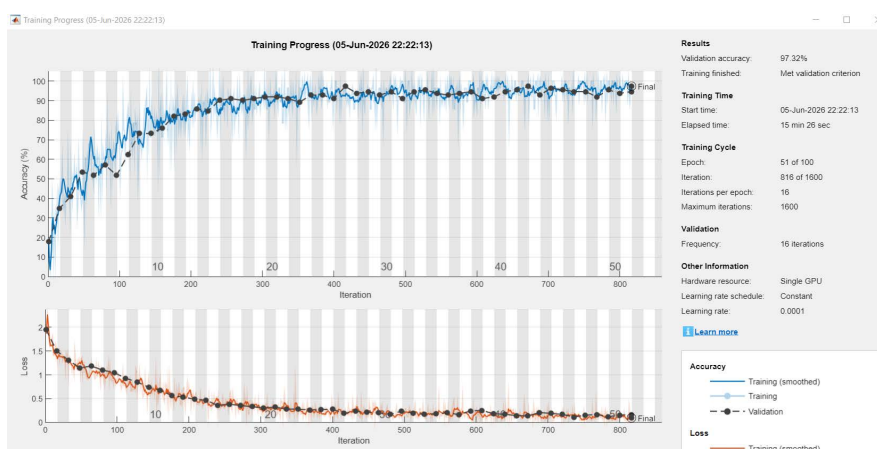


Figure 13. Learning curves (representative fold): MobileNetV2 + SGDM.

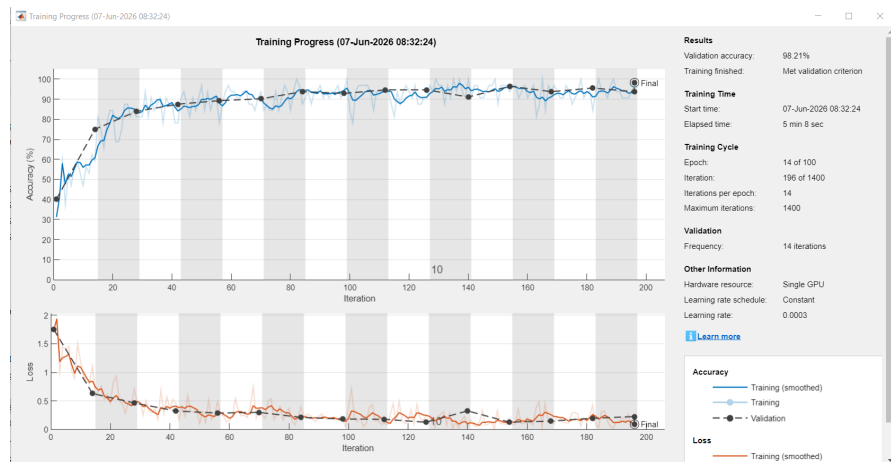


Figure 14. Learning curves (representative fold): MobileNetV2 + Adam.

In line with the GapLoss values reported in **Table 6**, the curves of the configurations trained with Adam (**Figure 12** and **Figure 14**) reach a high validation accuracy in a smaller number of epochs, at the cost of a slightly more pronounced training/validation gap on the loss function than for the SGDM configurations (**Figure 11** and **Figure 13**), whose convergence is more gradual but results in a smaller final gap.

4.3. Comparison with Previous Studies

The performance obtained in this study (F1 between 0.951 and 0.968, MCC between 0.938 and 0.961, AUC above 0.995 for all four configurations) lies in the upper range of recent work using residual or lightweight architectures for deep-learning-based classification of erythrocyte morphologies [3] [4]. Unlike several earlier studies that evaluate ResNet18 and MobileNetV2 separately or on a reduced number of classes, this work proposes a joint and systematic comparison of the two architectures on five pathological morphologies simultaneously, combined with multi-objective hyperparameter optimization—an approach still uncommon in the literature dedicated to digital hematology [3] [4]. This direct comparison, although carried out on a modest-sized dataset as is often the case in this field, makes it possible to document an accuracy/complexity trade-off that is rarely quantified as explicitly in existing work.

Recent peer-reviewed studies further help position this contribution within the broader landscape of multiclass red-blood-cell morphology classification. Dhar *et al.* [30] proposed a hyperparameter-optimized KNN classifier for poikilocytosis classification, whereas Dhar *et al.* [31] developed a hybrid CNN-SVM approach for the same task; both studies, evaluated on datasets of comparable size, reported competitive classification performance without explicitly addressing the trade-off among predictive accuracy, generalization, model complexity, and inference latency. At a larger scale, Routt *et al.* [4] achieved approximately 98% accuracy using an ensemble of four CNNs for a seven-class red-blood-cell morphology classification task on a substantially larger dataset. Compared with these studies, the pre-

sent work introduces an explicit multi-objective selection framework (Pareto Front and TOPSIS) that jointly balances classification performance, generalization indicators, model complexity, and inference latency—a dimension not addressed in [30] [31] or [4].

4.4. Discussion

The four optimal configurations achieve an F1 above 0.95 and an MCC above 0.93, with an AUC systematically above 0.995 (**Table 6**). This level of performance, obtained on a database limited to 560 image patches distributed across five morphological classes, confirms the value of transfer learning: the generic representations inherited from ImageNet, combined with partial fine-tuning, are sufficient to effectively discriminate morphologies that are nonetheless visually close (echinocytes/acanthocytes, elliptocytes/ovalocytes), despite the small size of the dataset. The AUC varies very little between configurations (0.9954 to 0.9975), which indicates that it saturates quickly and loses its discriminating power for distinguishing between hyperparameters: it is essentially the pair $(1 - F1, 1 - MCC)$, more sensitive to errors on minority classes, that drives the TOPSIS selection—consistent with the attention paid to preserving minority classes (echinocytes, acanthocytes) through stratified cross-validation (Section 3.1).

For the same Adam optimizer, ResNet18 and MobileNetV2 achieve an almost identical F1 (0.9659 versus 0.9659) and a very close MCC (0.9574 versus 0.9570). The deeper architecture of ResNet18 therefore does not provide a measurable discriminative advantage here over the lighter architecture of MobileNetV2, which confirms a posteriori the complementarity hypothesis formulated in Section 3.3: on a five-class problem with a limited dataset, the additional depth of ResNet18 does not bring a decisive representational gain. On the other hand, the gap is clear in terms of model complexity: the Params_norm parameter goes from 0.932 (ResNet18) to 0.187 (MobileNetV2), a direct reflection of the roughly 3.3 ratio between the ~11.7 million parameters of ResNet18 and the ~3.5 million of MobileNetV2. A more unexpected result concerns the normalized latency: it is systematically higher for MobileNetV2 (0.32 - 0.33) than for ResNet18 (0.177 - 0.179), whereas this architecture is designed for fast inference. This counter-intuitive result is likely explained by the fact that the depthwise separable convolutions of MobileNetV2, although less costly in terms of parameters and theoretical operations, are often less well optimized than classical dense convolutions by the libraries and hardware used (MATLAB implementation, GTX 1070 GPU), which do not necessarily benefit from the specialized kernels available on more recent frameworks. The theoretical gain in lightness therefore does not automatically translate into a measured gain in inference time, which constitutes a limitation to be documented more precisely in future work, through a framework-independent latency measurement.

The Adam optimizer consistently converges in a markedly lower number of epochs than SGDM, for an equivalent architecture: 14 epochs versus 25 for Res-

Net18 (a reduction of about 44%), and 16 epochs versus 40 for MobileNetV2 (a reduction of about 60%). This acceleration of convergence is consistent with Adam's adaptive properties (Section 3.5), which individually adjust the learning-rate step per parameter, unlike SGDM, whose inertia improves the stability of the descent without necessarily accelerating its speed. This speed, however, has an observable counterpart on GapLoss, the overfitting indicator: for ResNet18, Adam shows a GapLoss of 0.0907 versus 0.0405 for SGDM, more than double; for MobileNetV2, the gap goes in the same direction although less pronounced (0.0623 versus 0.0458). Adam thus appears to converge more quickly towards an optimum that generalizes slightly less well than the one reached more gradually by SGDM, even though the early-stopping mechanism limits this risk by halting training as soon as validation stops improving. This trade-off between convergence speed and generalization qualifies the common belief that Adam is strictly superior to SGDM.

Examination of **Tables 2-5** shows that the value $LR = 10^{-4}$ recurs frequently among the configurations selected by TOPSIS (three of the four optimal configurations), confirming its position as a robust convergence value. Higher learning rates ($5 \cdot 10^{-4}$, 10^{-3}) tend to reduce the number of epochs required but generally degrade F1 and MCC, reflecting the learning instability inherent to high learning rates. The optimal batch size varies more depending on the architecture and the optimizer (12 to 32 depending on the case), without as clear a trend as for the learning rate, which suggests a secondary sensitivity of this parameter within the tested range.

5. Conclusions

This study proposed an integrated approach for the isolation and morphological classification of pathological red blood cells from blood-smear images. By combining pre-trained convolutional neural network architectures (ResNet18 and MobileNetV2) with a multi-objective hyperparameter optimization strategy (Pareto Front and TOPSIS), we showed that it is possible to obtain robust performance ($F1 > 0.95$; $MCC > 0.93$; $AUC > 0.995$) despite the limited size of the dataset. The use of 5-fold stratified cross-validation helped strengthen the reliability of the results and reduce the risk of overfitting.

The results obtained confirm the relevance of transfer learning for biomedical applications where annotated databases are scarce and costly to produce. ResNet18 stood out for its ability to extract deep, discriminative representations, with the lowest GapLoss, while MobileNetV2 demonstrated almost equivalent performance with nearly five times fewer parameters, although its latency measured in practice remains higher than that of ResNet18. This complementarity opens up prospects for the development of accessible and reproducible automated systems in clinical hematology, with the choice between the two architectures ultimately depending on deployment constraints (maximum accuracy sought versus model lightness).

Conflicts of Interest

The authors declare that they have no conflict of interest related to this work.

References

- [1] Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., *et al.* (2017) A Survey on Deep Learning in Medical Image Analysis. *Medical Image Analysis*, **42**, 60-88. <https://doi.org/10.1016/j.media.2017.07.005>
- [2] Bessis, M. (1977) Blood Smears Reinterpreted. Translated by G. Brecher, Springer-Verlag.
- [3] Petrović, N., Moyà-Alcover, G., Jaume-i-Capó, A. and González-Hidalgo, M. (2020) Sickle-Cell Disease Diagnosis Support Selecting the Most Appropriate Machine Learning Method: Towards a General and Interpretable Approach for Cell Morphology Analysis from Microscopy Images. *Computers in Biology and Medicine*, **126**, Article ID: 104027. <https://doi.org/10.1016/j.combiomed.2020.104027>
- [4] Routt, A.H., Yang, N., Piety, N.Z., Lu, M. and Shevkoplyas, S.S. (2023) Deep Ensemble Learning Enables Highly Accurate Classification of Stored Red Blood Cell Morphology. *Scientific Reports*, **13**, Article No. 3152. <https://doi.org/10.1038/s41598-023-30214-w>
- [5] Otsu, N. (1979) A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**, 62-66. <https://doi.org/10.1109/tsmc.1979.4310076>
- [6] Niblack, W. (1986) An Introduction to Digital Image Processing. Prentice-Hall, 115-116.
- [7] Bernsen, J. (1986) Dynamic Thresholding of Grey-Level Images. *Proceedings of the 8th International Conference on Pattern Recognition (ICPR)*, Paris, 27-31 October 1986, 1251-1255.
- [8] Sauvola, J. and Pietikäinen, M. (2000) Adaptive Document Image Binarization. *Pattern Recognition*, **33**, 225-236. [https://doi.org/10.1016/s0031-3203\(99\)00055-2](https://doi.org/10.1016/s0031-3203(99)00055-2)
- [9] Vincent, L. and Soille, P. (1991) Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 583-598. <https://doi.org/10.1109/34.87344>
- [10] Cortes, C. and Vapnik, V. (1995) Support-Vector Networks. *Machine Learning*, **20**, 273-297. <https://doi.org/10.1023/a:1022627411411>
- [11] Breiman, L. (2001) Random Forests. *Machine Learning*, **45**, 5-32. <https://doi.org/10.1023/a:1010933404324>
- [12] Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A., Eds., *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015)*, Springer, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28
- [13] He, K., Gkioxari, G., Dollar, P. and Girshick, R. (2017) Mask R-CNN. 2017 *IEEE International Conference on Computer Vision (ICCV)*, Venice, 22-29 October 2017, 2961-2969. <https://doi.org/10.1109/iccv.2017.322>
- [14] Hörst, F., Rempe, M., Heine, L., Seibold, C., Keyl, J., Baldini, G., *et al.* (2024) Cellvit: Vision Transformers for Precise Cell Segmentation and Classification. *Medical Image Analysis*, **94**, Article ID: 103143. <https://doi.org/10.1016/j.media.2024.103143>
- [15] Stringer, C., Wang, T., Michaelos, M. and Pachitariu, M. (2021) Cellpose: A Generalist Algorithm for Cellular Segmentation. *Nature Methods*, **18**, 100-106.

- <https://doi.org/10.1038/s41592-020-01018-x>
- [16] Schmidt, U., Weigert, M., Broaddus, C. and Myers, G. (2018) Cell Detection with Star-Convex Polygons. In: Frangi, A., Schnabel, J., Davatzikos, C., Alberola-López, C. and Fichtinger, G., Eds., *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2018)*, Springer, 265-273.
https://doi.org/10.1007/978-3-030-00934-2_30
 - [17] He, H. and Garcia, E.A. (2009) Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, **21**, 1263-1284.
<https://doi.org/10.1109/tkde.2008.239>
 - [18] Shorten, C. and Khoshgoftaar, T.M. (2019) A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, **6**, Article No. 60.
<https://doi.org/10.1186/s40537-019-0197-0>
 - [19] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 770-778. <https://doi.org/10.1109/cvpr.2016.90>
 - [20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520.
<https://doi.org/10.1109/cvpr.2018.00474>
 - [21] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015) ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
 - [22] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q. (2017). Densely Connected Convolutional Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 2261-2269.
<https://doi.org/10.1109/cvpr.2017.243>
 - [23] Bergstra, J. and Bengio, Y. (2012) Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, **13**, 281-305.
 - [24] Snoek, J., Larochelle, H. and Adams, R.P. (2012) Practical Bayesian Optimization of Machine Learning Algorithms. *26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, 3-6 December 2012, 2951-2959.
 - [25] Polyak, B.T. (1964) Some Methods of Speeding up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, **4**, 1-17.
[https://doi.org/10.1016/0041-5553\(64\)90137-5](https://doi.org/10.1016/0041-5553(64)90137-5)
 - [26] Kingma, D.P. and Ba, J. (2015) Adam: A Method for Stochastic Optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, 7-9 May 2015.
 - [27] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, 6-11 July 2015, 448-456.
 - [28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, **15**, 1929-1958.
 - [29] Alico, J.N., Ouattara, S. and Clement, A. (2020) A New Strategy for the Morphological and Colorimetric Recognition of Erythrocytes for the Diagnosis of Forms of Anemia Based on Microscopic Color Images of Blood Smears. *International Journal of Advanced Computer Science and Applications*, **11**, 517-527.
<https://doi.org/10.14569/ijacsa.2020.0110762>

- [30] Dhar, P., Kothandapani, S.D., Satti, S.K. and Padmanabhan, S. (2023) HPKNN: Hyper-Parameter Optimized KNN Classifier for Classification of Poikilocytosis. *International Journal of Imaging Systems and Technology*, **33**, 928-950.
<https://doi.org/10.1002/ima.22841>
- [31] Dhar, P., Suganya Devi, K. and Srinivasan, P. (2023) CNN-RSVM: A Hybrid Approach for Classification of Poikilocytosis Using Convolutional Neural Network and Radial Kernel Basis Support Vector Machine. *Computer Methods in Biomechanics and Biomedical Engineering. Imaging & Visualization*, **11**, 2118-2129.
<https://doi.org/10.1080/21681163.2023.2219755>

Supplementary Material

Pareto status and TOPSIS closeness coefficient (Ci) for all 20 hyperparameter configurations tested for each architecture-optimizer combination (5 mini-batch sizes $\times$ 4 learning rates), provided to ensure full reproducibility of the two-stage selection process described in Section 3.6. Within each learning-rate block (5 rows), the highlighted row corresponds to the configuration selected by TOPSIS for that learning rate.

Table S1. ResNet18 + SGDM—20 configurations tested.

LR	BS	Epoch	Pareto status	Ci
1e-4	12	21	Non-dominated	0.5878
1e-4	16	22	Non-dominated	0.4875
1e-4	24	25	Non-dominated	0.9767
1e-4	28	25	Non-dominated	0.2206
1e-4	32	28	Dominated	0.0909
3e-4	12	18	Non-dominated	0.7096
3e-4	16	19	Non-dominated	0.6201
3e-4	24	28	Non-dominated	0.2158
3e-4	28	19	Non-dominated	0.4015
3e-4	32	27	Non-dominated	0.7902
5e-4	12	20	Non-dominated	0.6110
5e-4	16	18	Dominated	0.0390
5e-4	24	17	Dominated	0.4401
5e-4	28	17	Non-dominated	0.8868
5e-4	32	20	Non-dominated	0.8927
1e-3	12	12	Dominated	0.0086
1e-3	16	16	Non-dominated	0.6448
1e-3	24	18	Non-dominated	0.9677
1e-3	28	19	Non-dominated	0.9116
1e-3	32	19	Non-dominated	0.9172

Table S2. ResNet18 + Adam—20 configurations tested.

LR	BS	Epoch	Pareto status	Ci
1e-4	12	17	Non-dominated	0.8525
1e-4	16	16	Dominated	0.7642
1e-4	24	15	Non-dominated	0.3872
1e-4	28	16	Dominated	0.3864
1e-4	32	14	Non-dominated	0.8803
3e-4	12	14	Non-dominated	0.6638
3e-4	16	15	Non-dominated	0.5414
3e-4	24	13	Dominated	0.6063
3e-4	28	14	Dominated	0.0130
3e-4	32	14	Non-dominated	0.9825
5e-4	12	10	Non-dominated	0.2905
5e-4	16	13	Dominated	0.1997
5e-4	24	13	Dominated	0.7328
5e-4	28	17	Non-dominated	0.9794
5e-4	32	12	Non-dominated	0.8841
1e-3	12	12	Non-dominated	0.4117
1e-3	16	12	Dominated	0.3176
1e-3	24	19	Non-dominated	0.5253
1e-3	28	16	Non-dominated	0.8428
1e-3	32	13	Dominated	0.2720

Table S3. MobileNetV2 + SGDM—20 configurations tested.

LR	BS	Epoch	Pareto status	Ci
1e-4	12	24	Dominated	0.3057
1e-4	16	27	Non-dominated	0.8069
1e-4	24	36	Dominated	0.2063
1e-4	28	40	Non-dominated	0.8417
1e-4	32	40	Dominated	0.3742
3e-4	12	21	Non-dominated	0.9995
3e-4	16	20	Non-dominated	0.0707
3e-4	24	21	Dominated	0.5755
3e-4	28	25	Dominated	0.5110
3e-4	32	28	Dominated	0.6748
5e-4	12	21	Non-dominated	0.8432
5e-4	16	21	Dominated	0.2076
5e-4	24	18	Non-dominated	0.6734
5e-4	28	27	Dominated	0.5044
5e-4	32	22	Dominated	0.5179
1e-3	12	17	Non-dominated	0.4705
1e-3	16	21	Non-dominated	0.6828
1e-3	24	15	Dominated	0.1526
1e-3	28	17	Dominated	0.3538
1e-3	32	19	Non-dominated	0.5351

Table S4. MobileNetV2 + Adam—20 configurations tested.

LR	BS	Epoch	Pareto status	Ci
1e-4	12	18	Non-dominated	0.9856
1e-4	16	18	Dominated	0.1633
1e-4	24	16	Non-dominated	0.4254
1e-4	28	17	Non-dominated	0.5817
1e-4	32	23	Non-dominated	0.4036
3e-4	12	17	Dominated	0.4455
3e-4	16	18	Non-dominated	0.6467
3e-4	24	16	Non-dominated	0.7962
3e-4	28	18	Dominated	0.0069
3e-4	32	16	Non-dominated	0.9202
5e-4	12	14	Dominated	0.2896
5e-4	16	15	Non-dominated	0.7569
5e-4	24	23	Dominated	0.6163
5e-4	28	15	Non-dominated	0.3652
5e-4	32	16	Non-dominated	0.7536
1e-3	12	16	Non-dominated	0.4842
1e-3	16	14	Dominated	0.2089
1e-3	24	18	Non-dominated	0.8255
1e-3	28	17	Dominated	0.2947
1e-3	32	15	Non-dominated	0.7700