International Journal of Artificial Intelligence in Medical Issues

Volume 3 Issue 2
ISSN 3025-4167
https://doi.org/10.56705/ijaimi.v3i2.364

Research Article

Hybrid Feature Benchmark for Blood Cell Classification Using ResNet50 and EfficientNetV2 Features with SVM and ANN Classifiers via Unsupervised Segmentation

Ahmad Kholish Fauzan Shobiry [1,*]; Rahma Puspitasari [2]
[1] Universitas Negeri Malang, Malang, Indonesia, ahmad.kholish.2505348@students.um.ac.id
[2] Universitas Negeri Malang, Malang, Indonesia, rahma.puspitasari.2505348@students.um.ac.id
Correspondence should be addressed to Ahmad Kholish Fauzan Shobiry; ahmad.kholish.2505348@students.um.ac.id

Received 05 September 2025; Revised 10 September 2024; Accepted 25 October 2024; Published 30 November 2025

Abstract:
Automated blood cell classification supports hematological diagnosis by providing objective and efficient analysis, but end-to-end deep learning models often require substantial computational resources that limit deployment on low-resource clinical devices. This study evaluates whether frozen deep features extracted from EfficientNetV2B0 or ResNet50 provide better separability for the eight BloodMNIST classes, and examines which classical classifier offers the most practical balance of accuracy, model size, and training time. The BloodMNIST dataset, consisting of 11,959 training images, 1,712 validation images, and 3,421 test images, is processed using data augmentation and Otsu-based unsupervised segmentation before the resulting masks are replicated into three channels and passed into pretrained ImageNet CNNs used strictly as frozen feature extractors. The extracted features are classified using Support Vector Machine with grid search, K-Nearest Neighbor, Artificial Neural Network, and Random Forest, with performance assessed through accuracy, precision, recall, and F1-score. EfficientNetV2 with Support Vector Machine achieves the highest performance, reaching 76.8% test accuracy, 75.3% precision, 72.6% recall, and a 73.6% F1-score, while EfficientNetV2 with Artificial Neural Network provides a comparable 76.2% accuracy and a 73.0% F1-score with a compact 2 MB model size. These findings highlight a clear trade-off between accuracy, model size, and computational cost, demonstrating that hybrid deep-feature pipelines offer lightweight and effective solutions for blood cell classification in resource-constrained clinical settings.

Keywords: BloodMNIST, Feature Extraction, EfficientNetV2, ResNet50, Machine Learning, Medical Image Classification.

Dataset link: https://zenodo.org/records/10519652

1. Introduction

Blood cell morphology analysis plays a crucial role in medical diagnostics, providing valuable information about a patient's hematological condition [1], [2]. Accurate identification and classification of blood cell types are essential in clinical practice, as each cell type has unique morphological characteristics and physiological functions [3]. In blood cell classification, there are eight primary classes that are the focus of analysis: Basophil, Eosinophil, Erythroblast, Immature Granulocyte (IG), Lymphocyte, Monocyte, Neutrophil, and Platelet. The ability to distinguish these cell classes based on their morphological characteristics such as shape, size, and internal structure forms an important foundation in hematological diagnosis. Blood microscopy provides valuable insights into blood cell morphology and enables early detection of hematological abnormalities and infections [2]. In a study conducted by Su et al., a predictive machine learning model on routine blood test data has become an effective initial screening tool and helps ease

standard procedures in hospitals with ease of access and cost savings [4]. In addition, a blood cell morphological analysis-based system can serve as a clinical decision support system that provides an objective assessment, helping clinicians make more informed decisions regarding the patient's clinical condition [4]. Therefore, the development of automated systems for blood cell classification is essential to improve efficiency in hematological analysis.

Recent studies have demonstrated the effectiveness of deep learning for hematological image analysis, particularly in white blood cell classification using pretrained CNN backbones such as ResNet-50 and EfficientNet variants [5], [6], [7]. However, only a limited number of works have conducted controlled, head-to-head comparisons of different pretrained CNN feature extractors under the same preprocessing and classification pipeline [6], [7]. It remains unclear whether features generated by EfficientNetV2 provide superior class separability compared with ResNet50 embeddings when paired with classical machine learning classifiers. Furthermore, previous research has rarely examined which classical learner achieves the optimal trade-off between accuracy, computational time, and model size when deep features are fixed. To address this gap, the present study is guided by two central research questions:

a Do EfficientNetV2 features or ResNet50 features provide better separability for the eight BloodMNIST classes under classical machine learning classifiers?

b Given fixed deep features, which classifier in between SVM, ANN, KNN, or Random Forest offers the optimal trade-off among accuracy, computational time, and model size?

This work contributes a clean, transparent benchmark by evaluating eight hybrid configurations within a unified pipeline. All images are processed using Otsu-based unsupervised segmentation to isolate cell morphology, and both pretrained CNNs are used strictly as frozen feature extractors. The four classical classifiers are then compared consistently in terms of accuracy, efficiency, and model complexity. This study provides a reproducible reference for implementing lightweight diagnostic systems suitable for resource-constrained clinical environments.

2. Method

This chapter presents the experimental design, hybrid pipeline architecture, and technical configurations used to systematically evaluate the performance of blood cell classification. The proposed methodology focuses on cell morphology by separating the data cleansing and morphological isolation stages from the feature extraction and classification stages. A visual representation of the entire research process is illustrated in **Figure 1**.

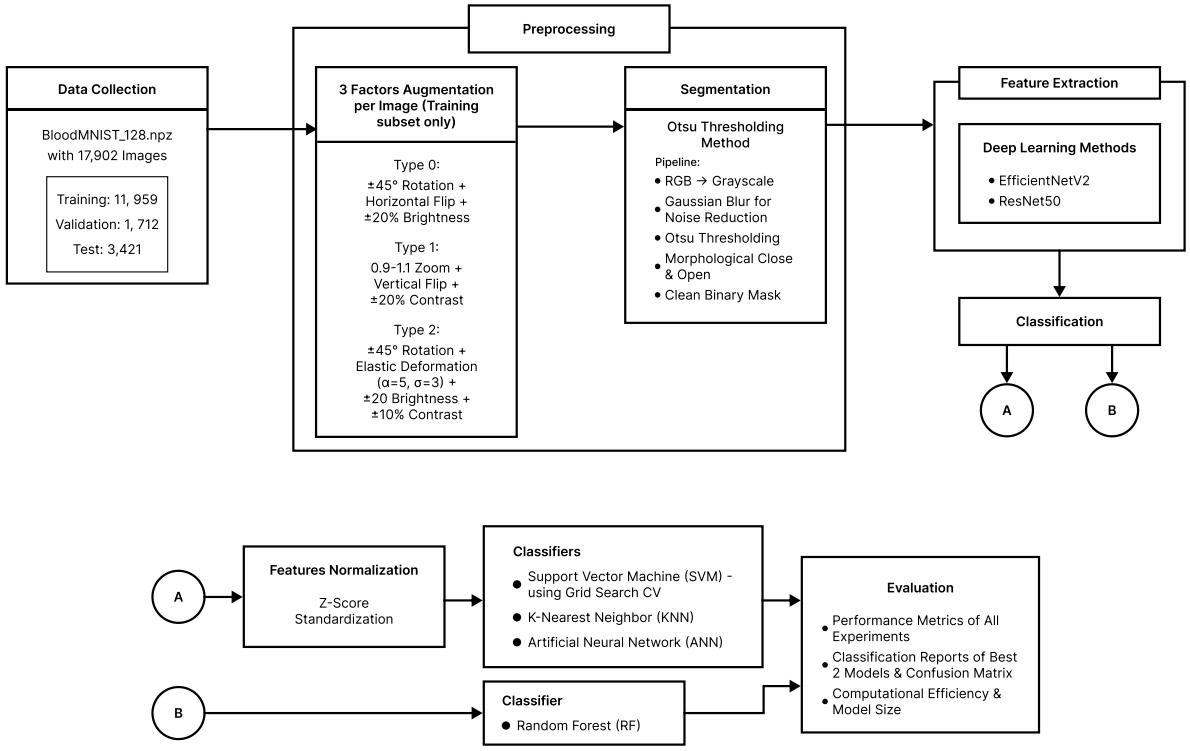
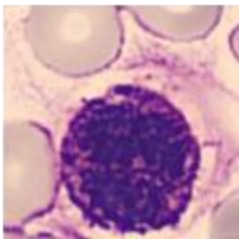


Figure 1. Deep Feature Extractors and Machine Learning Classifiers

Data Selection and Collection Process

This study fully used the BloodMNIST dataset as the primary data source [8], [9]. This dataset is one of the benchmarks tested in the classification of medical images because it displays an adequate diversity of blood cell images with a resolution of 128 x 128 pixels. These images have been classified into eight different classes, which include white blood cells (Basophil, Eosinophil, Lymphocyte, Monocyte, Neutrophil, and Immature Granulocytes/IGs), pink blood cells (Erythroblasts), and Platelets. This multi-class availability allows us to test the discriminatory capabilities of models in complex diagnostic contexts. Sample of each class is shown in **Figure 2**.



Class 0: Basophil Class 1: Eosinophil Class 2: Erythroblast Class 3: IG

Class 4: Lymphocyte Class 5: Monocyte Class 6: Neutrophil Class 7: Platelet

Figure 2. Original Images of All Classes

To guarantee the validity and generalization of the results, the dataset is strictly divided into three separate sets. The Training Set has images of 11,959 samples for model learning, the Validation Set contains 1,712 samples used exclusively for hyperparameter tuning, while the Test Set consisting of 3,421 samples is set aside and will only be used once at the end to obtain an unbiased and final model performance estimate. The subset division in this dataset follows the initial configuration that has been confirmed by the source.

Data Analysis Methods

a. Augmentation

Augmentasi yang digunakan dalam penelitian ini terdiri dari tiga konfigurasi yang dirancang untuk meningkatkan keragaman data tanpa mengubah karakteristik visual utama objek. Tipe 0 mengombinasikan rotasi hingga ±45°, horizontal flip, dan penyesuaian kecerahan sebesar ±10–20% untuk mensimulasikan variasi orientasi dan kondisi pencahayaan. Tipe 1 menerapkan vertical flip, zoom dalam rentang 0.9–1.1, serta modifikasi kontras ±10–20% guna merepresentasikan perubahan skala dan distribusi pencahayaan. Sementara itu, Tipe 2 memadukan rotasi ±45°, elastic deformation, serta pergeseran brightness dan contrast untuk menghasilkan variasi bentuk yang lebih kompleks, tetap dalam batas yang tidak mengubah struktur semantik objek.

The first step in data analysis is rigorous augmentation and aims to improve the robustness of the model against reasonable image variations in the clinical setting [10]. Realizing that deep learning is prone to overfitting specific medical datasets [11], the original training Set was significantly expanded, that is, tripled, through the implementation of three different augmentation schemes. Type 0 combines rotations up to ±45°, horizontal flipping, and brightness adjustments of ±10–20% to simulate variations in orientation and illumination. Type 1

applies vertical flipping, zooming within the 0.9–1.1 range, and contrast modifications of ±10–20% to represent changes in scale and light distribution. Meanwhile, Type 2 integrates ±45° rotations, elastic deformation, and coordinated shifts in brightness and contrast. These transformation options are multi-faceted, including geometric transformations and lighting [12]. In addition, brightness and contrast adjustments are implemented to reduce the sensitivity of the model to microscope lighting variations [13]. The most significant in this context is the inclusion of Elastic Deformation configured with α=5 (alpha) and σ=3 (sigma) which is applied to the original images. The existence of this elastic deformation is an essential biological simulation, aimed at creating subtle and non-linear variations in cell shape, thus encouraging models to develop feature representations that are invariant to natural morphological changes [14]. A sample of the augmentation results can be seen in **Figure 3**.

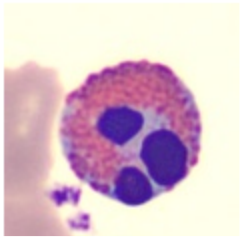


Figure 3. Results of Proposed Augmentation

b. Otsu Thresholding and Morphological Operation

After augmentation, each resulting image undergoes an unsupervised segmentation stage using Otsu Thresholding [15]. This methodological decision is based on the philosophy of ignoring background noise and artificially isolating the cell's Region of Interest (ROI), thereby mimicking the cognitive processes of a medical analyst which focuses on cell area. This segmentation process starts with the conversion of the RGB image to grayscale and then Gaussian Blurring is applied before thresholding [16]. This blurring is very important because it serves as a low-pass filter that stabilizes the Otsu threshold amidst image noise. The auto-generated threshold is then used to create the initial binary mask. These binary masks are not used immediately but are refined through Morphology Operations [17]. Morphological Closing with 2 iterations is applied to close the hole in the foreground of the cell, while Morphological Opening with 1 iteration is used to clear the noise of spots in the background [18]. The end result of this entire pipeline is a clean, 1-channel binary mask, which will serve as a focused input for the deep feature extraction stage. The depiction of these processes is served in the **Figure 4**.

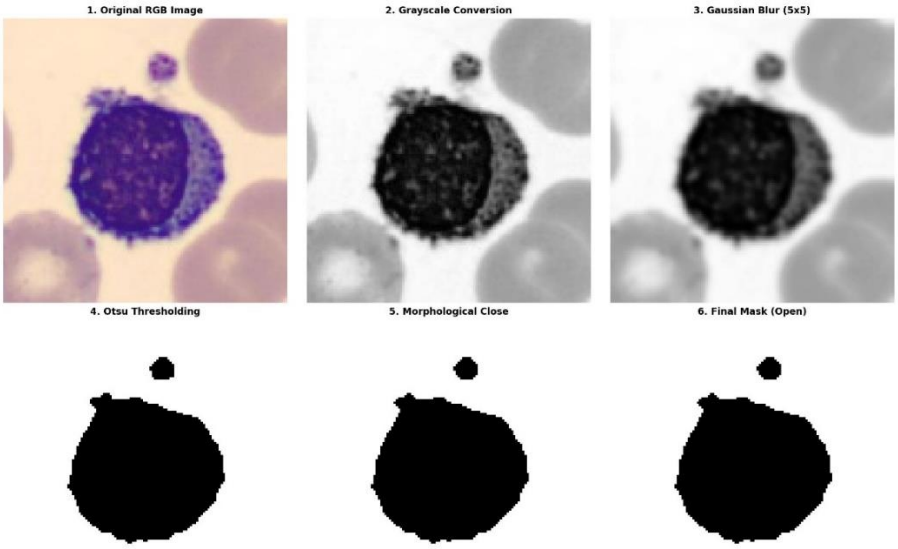


Figure 4. Otsu Thresholding and Morphological Processes

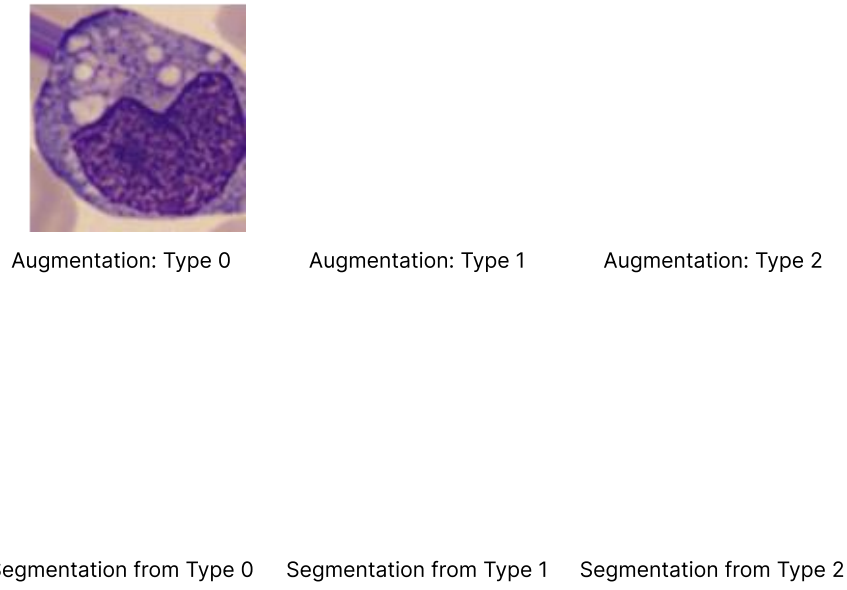


Figure 5. Example of Final Segmented Augmentation

c. Feature Extraction and Normalization

A crucial stage in this methodology is the utilization of Transfer Learning to extract high-dimensional feature vectors. The feature extraction only uses the masked images which were already augmented, then we explicitly use deep learning models in static feature extraction, obtaining the features directly from the final convolutional layer of the models where were aggregated through global average pooling. The two architectures we compared

were loaded with ImageNet pre-trained weights and all of their layers were frozen [19], [20]. The main technical adaptation of the input is Grayscale to RGB Replication, a 1-channel binary mask of Otsu segmentation is stacked to three channels to meet the input requirements of the pre-trained model [21]. This adjustment is necessary because deep learning ImageNet-pretrained models, including ResNet50 and EfficientNet variants, are architecturally designed to accept RGB inputs of fixed dimensionality (H×W×3) [22]. For this benchmark study, we compared two leading deep learning architectures, ResNet50 and EfficientNetV2B0. ResNet50 produces a 2048-dimensional dense feature vector, demonstrating the power of representation of a very deep network [23]. Meanwhile, EfficientNetV2B0 is similarly configured, resulting in a more compact feature vector with 1280-dimensional, which highlights better efficiency and throughput [24]. Both feature vectors are extracted through Global Average Pooling 2D. After extraction, the feature vector undergoes Z-score normalization for SVM, KNN, and ANN to prevent features with large value ranges from dominating the learning process, while raw features are set aside for Random Forest because of the scale-invariant nature of decision trees [25]. The calculation of standardization is represented as (1):

$$X_{norm} = \frac{X - \mu}{\sigma} \tag{1}$$

Where the original feature value is X, the mean of the feature is shown as μ, and σ is the feature's standard deviation. Sample results of both raw data and normalized feature extraction results can be seen in **Table 1** and **Table 2**.

Table 1: Samples of Raw ResNet50 Feature Extractor

Images	*Feature 0*	*Feature 1*	*Feature 2*	*Feature 3*	*Feature 4*	*Feature 5*
idx10023_aug0_mask	0.325267	0.0	0.019454	1.797169	0.0	0.023108
idx10023_aug1_mask	0.0	0.0	0.0	1.963830	0.009667	0.013963
idx10023_aug2_mask	0.277097	0.0	0.728251	1.433306	0.0	0.373128
idx10024_aug0_mask	0.550510	0.0	0.124192	3.686136	0.018552	0.0
idx10024_aug1_mask	0.257808	0.0	0.0	1.909080	0.0	0.0

Table 2: Sample of Normalized ResNet50 Features

Images	*Feature 0*	*Feature 1*	*Feature 2*	*Feature 3*	*Feature 4*	*Feature 5*
idx10023_aug0_mask	-0.368135	-0.695199	-0.349653	0.274612	-0.412774	-0.340839
idx10023_aug1_mask	-0.804307	-0.695199	-0.404234	0.415483	-0.368443	-0.367452
idx10023_aug2_mask	-0.432729	-0.695199	1.638950	-0.032942	-0.412774	0.677783
idx10024_aug0_mask	-0.066092	-0.695199	-0.055800	1.871262	-0.327698	-0.408089
idx10024_aug1_mask	-0.458595	-0.695199	-0.404234	0.369205	-0.412774	-0.408089

d. Classification

The processed deep feature vectors were then fed into four different classical Machine Learning algorithms for eight experiments aimed at identifying the optimal combination of Feature Extractor and Classifier [26], [27]. Each classifier is set up with a specific and optimized configuration. Random Forest (RF) is used with a 200 estimator as a robust and fast-trained ensemble baseline [28]. The Support Vector Machine (SVM) employed an RBF kernel, with its key hyperparameters optimized through an exhaustive Grid Search using 5-Fold Cross-Validation. The search space included $C \in \{1,10,100\}$ and $\gamma \in \{0.01,0.1, \text{scale}\}$, evaluated using accuracy as the scoring metric and executed in parallel to accelerate model selection. K-Nearest Neighbors (KNN) is configured with nine nearest neighbors, utilizing the Euclidean Distance metric, with distance-weighted decisions, which means that a closer sample significantly influences the final classification outcome [29]. Finally, the Artificial Neural Network (ANN) was implemented as a two-layer fully connected architecture with 128 and 64 hidden units activated by ReLU. Training was conducted using the Adam optimizer with a very low learning rate of 0.00001, a batch size of 32, and a maximum of 50 epochs. To improve generalization, Batch Normalization was applied, while a dropout rate of 0.2 was introduced to mitigate co-adaptation among neurons. The training process also incorporated a dynamic callback system, consisting of Early Stopping with a patience of ten epochs and ReduceLROnPlateau to automatically lower the learning rate when the validation loss plateaued for five consecutive epochs, ensuring a stable and optimal convergence trajectory.

e. Performance Evaluation

The evaluation stage is carried out strictly on the Test Set to obtain a valid and generalizable performance estimate. The quantitative evaluation is focused on three important performance metrics. Accuracy was used as the primary comparison metric across the eight experiments. Furthermore, to provide a nuanced and fair view, we also calculate the Precision, Recall, and F1-Score [30].

$$\text{Accuracy} = \frac{\text{Number of Correct Predicitions}}{\text{Total Number of Predicitions}} \tag{2}$$

$$\text{Precision} = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives} \tag{3}$$

$$\text{Recall} = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives} \tag{4}$$

$$F1 - Score = \frac{2\ \times\ Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{5}$$

Next, we also present the classification results from the two best models. This is to demonstrate the generalization ability of the two best models to classify between cell classes. The culmination of the analysis is the interpretation of the Confusion Matrix of our four best models. This matrix serves as a very valuable visual diagnostic tool because it explicitly shows patterns of misclassification between classes [31]. By analysing Confusion Matrix, we will be able to empirically determine which cell types are most prone to error and which

features between ResNet50 and EfficientNetV2B0 have proven to be more resilient in separating morphologically similar classes. The results of these quantitative metrics and Confusion Matrix will be the basis for an in-depth discussion in the next chapter. Lastly, we also provide the trade-off between accuracy, computational efficiency [32], and model size from our best four models as initial recommendations for deployment in limited environments.

f. System Environment

To ensure the consistency and reproducibility of all experimental results, the entire workflow was executed within a controlled computational environment. Deep learning models were developed using TensorFlow 2.13.0 and Keras 2.13.0, running on a CUDA 11.2 backend supported by an NVIDIA GeForce RTX 3060 GPU (12 GB VRAM). Traditional machine learning classifiers, including SVM, KNN, and ANN variants, were implemented using scikit-learn 1.3.0, while additional utilities relied on XGBoost 2.0.0, NumPy 1.24.3, SciPy 1.11.2, OpenCV 4.8.0.76, Pillow 10.0.0, joblib 1.3.2, and tqdm 4.66.1. To minimize stochastic variation across training and evaluation processes, a fixed random seed of 42 was applied consistently throughout all modules, including data preprocessing, model initialization, and optimization routines.

Result and Discussion:

Result

Table 3 and **Table 4** show the performance results of four different classification algorithms, SVM with Grid Search CV, KNN, ANN, and Random Forest in classifying images in the BloodMNIST dataset using two different feature extraction approaches, namely EfficientNetV2 and ResNet50. Each model is evaluated based on five main metrics, namely Train Accuracy, Test Accuracy, Precision, Recall, and F1-Score.

Table 3 shows that the combination of EfficientNetV2 + SVM Grid Search CV demonstrated the best performance with a test accuracy of 76.8% and a training accuracy of 86.6%. The Precision value of 75.3%, Recall of 72.6%, and F1-Score of 73.6% indicate a balance between the model's ability to identify positive classes and avoid misclassification. These results demonstrate that SVM with parameter adjustment through Grid Search is capable of optimizing the separation of high-dimensional features generated by EfficientNetV2.

The ANN model also achieved competitive performance with a test accuracy of 76.2% and an F1-Score of 73.0%, only slightly lower than the SVM. This indicates that simple neural networks are still quite effective in capturing non-linear patterns in the features extracted by EfficientNetV2.

Meanwhile, Random Forest achieved a test accuracy of 72.9% and an F1-Score of 68.0%, demonstrating fairly stable performance but not as high as the previous two models. The KNN model achieved the lowest results with a testing accuracy of 70.3% and an F1-score of 65.6%, despite achieving 100% training accuracy, indicating signs of overfitting due to a strong reliance on training data.

Overall, the results in **Table 3** indicate that the combination of EfficientNetV2 with SVM and ANN provides the best balance between generalization and accuracy in BloodMNIST image classification.

Table 3. Classification Results on BloodMNIST Dataset Using EfficientNetV2 Feature Extraction

Algorithm	Train Accuracy	Test Accuracy	Precision	Recall	F1 – Score
SVM + Grid Search CV	86.6%	76.8%	75.3%	72.6%	73.6%
KNN	100%	70.3%	68.0%	66.1%	65.6%
ANN	81.1%	76.2%	74.6%	72.1%	73.0%
Random Forest	100%	72.9%	71.8%	66.9%	68.0%

Table 4. Classification Results on BloodMNIST Dataset Using ResNet50 Feature Extraction

Algorithm	Train Accuracy	Test Accuracy	Precision	Recall	F1 – Score
SVM + Grid Search CV	84.5%	74.3%	73.1%	69.5%	70.3%
KNN	100%	67.5%	64.3%	63.1%	62.6%
ANN	80.8%	73.9%	72.3%	69.6%	70.3%
Random Forest	100%	69.8%	68.9%	63.8%	64.9%

Table 4 shows the classification results using ResNet50 for feature extraction, demonstrating a similar performance pattern to EfficientNetV2, but with slightly lower metric values. The SVM (Grid Search CV) model again recorded the highest performance with a test accuracy of 74.3%, Precision of 73.1%, Recall of 69.5%, and F1-Score of 70.3%. This indicates that SVM remains the most consistent algorithm in utilizing features extracted by CNNs for class separation.

The ANN model has a performance very close to SVM, namely a test accuracy of 73.9% and an F1-Score of 70.35%, demonstrating the ability of neural networks in handling complex features generated by ResNet50. Meanwhile, Random Forest obtained a test accuracy of 69.8% with an F1-Score of 64.9%, and KNN was again the model with the lowest performance, namely a test accuracy of 67.5% and an F1-Score of 62.6%, reinforcing the finding that distance-based methods are less than optimal for high-dimensional image datasets.

After gaining an overall understanding of the results presented in the tables, the analysis shifts to the confusion matrices to examine how the model distributes its correct and incorrect predictions. Based on Figure 6, the confusion matrix discussed here correspond to the two model combinations that achieved the highest accuracy which are SVM trained with EfficientNetV2 features and SVM trained with ResNet50 features. These combinations were selected because they outperformed the other algorithms, making their visual inspection through confusion matrices particularly relevant for exploring the models' prediction behaviour in greater depth.

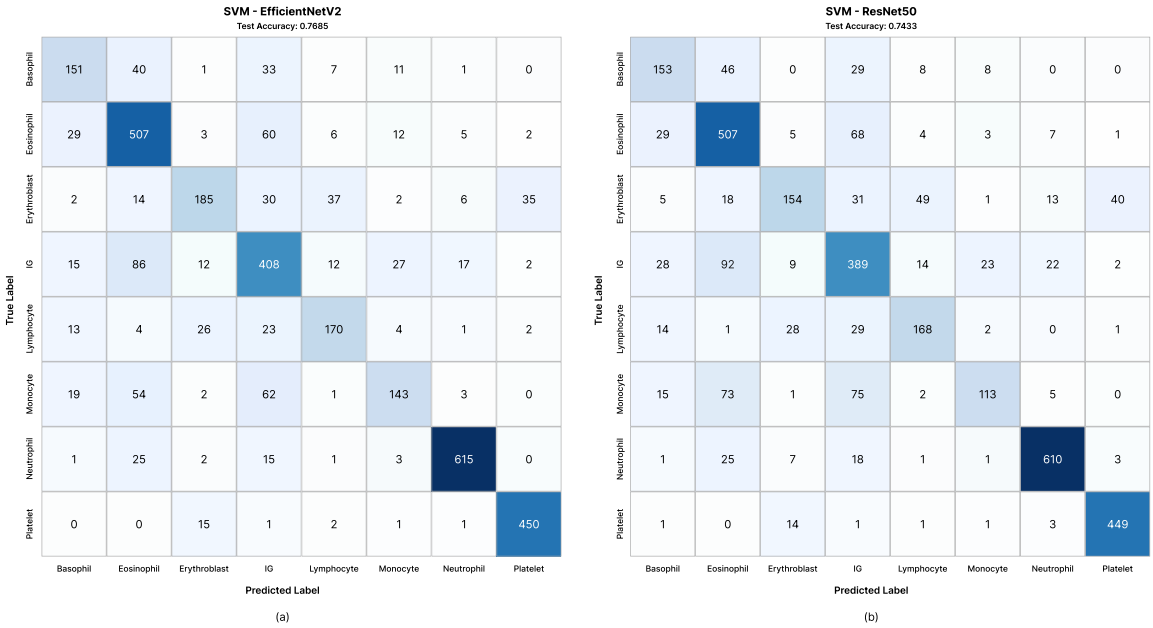


Figure 5. Confusion Matrix of the Best Hybrid Model between (a) EfficientNetv2, (b) ResNet50

The confusion matrix configurations for the two best models (EfficientNetV2+SVM and ResNet50+SVM) are presented in **Figure 6**. Each diagram uses the same class sequence (Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, Platelet) and is evaluated on a Test Set of 3,421 images to allow readers to cross-check the number of predictions.

In EfficientNetV2+SVM, the error pattern is more spread across several visually similar classes, in line with the complexity of the higher-dimensional feature representation. In contrast, ResNet50+SVM produces a more concentrated error pattern, reflecting more compact features, resulting in a more stable SVM margin despite slightly lower accuracy.

To clarify the difficulty level of each class, **Table 5** presents the precision and recall per class for the two best models (EfficientNetV2+SVM and EfficientNetV2+ANN).

Table 5. Per-Class Precision and Recall for the Two Best Models (EfficientNetV2 + SVM and EfficientNetV2 + ANN)

Class	Precision (SVM)	Recall (SVM)	Precision (ANN)	Recall (ANN)
Basophil	0.66	0.62	0.63	0.58
Eosinophil	0.69	0.81	0.68	0.74
Erythroblast	0.75	0.59	0.73	0.65
IG	0.65	0.70	0.65	0.67
Lymphocyte	0.72	0.70	0.71	0.70
Monocyte	0.70	0.50	0.71	0.61

Class	Precision (SVM)	Recall (SVM)	Precision (ANN)	Recall (ANN)
Neutrophil	0.95	0.92	0.94	0.93
Platelet	0.91	0.96	0.91	0.93

Furthermore, **Table 6** and **Table 7** present the results of the evaluation of four classification algorithms, SVM with Grid Search Cross Validation, KNN, ANN, and Random Forest. This evaluation considers three main aspects which are accuracy, model size, and computational time.

In **Table 6** which utilizes the EfficientNetV2 extraction feature, SVM with Grid Search CV provides the highest accuracy with 76.85%. Even so, this performance is accompanied by a very high computing time of 154,725 minutes. It should be emphasized that the computing time is training time. This very long duration arises because Grid Search must test many combinations of hyperparameters. The number of features that EfficientNetV2 generates is relatively large, so each iteration of the parameter search takes longer. On the other hand, ANN as classifier shows a good balance between the accuracy of the model size and the computational time. With an accuracy of 76.26%, a model size of only 2 MB and a much more efficient training time, ANN is an attractive alternative. KNN and Random Forest also delivered competitive results although their accuracy was slightly below SVM and ANN.

Table 6. Trade-off between Accuracy, Model Size, and Computational Time Using EfficientNetV2

Algorithm	Accuracy	Model Size (MB)	Computational Time (m)
SVM + Grid Search CV	76.8%	219	154725.55
KNN	70.3%	358	0.13
ANN	76.2%	2	2.11
Random Forest	72.9%	230	0.28

Table 7. Trade-off between Accuracy, Model Size, and Computational Time Using ResNet50

Algorithm	Accuracy	Model Size (MB)	Computational Time (m)
SVM + Grid Search CV	74.3%	371	2584.585
KNN	67.5%	574	0.216
ANN	73.9%	3	2.14
Random Forest	69.8%	245	0.25

Table 7 shows the performance of the four algorithms using the features of ResNet50. In general, the accuracy of the algorithm decreases slightly when compared to the results when using EfficientNetV2. SVM with Grid Search CV produces an accuracy of 74.3% with the training time of 2,584 minutes. Although the training time is still large, this duration is much shorter than when using EfficientNetV2. This difference occurs because the number of features generated by ResNet50 is less, so the parameter search process in Grid Search takes place faster. ANN again showed stable performance with an accuracy of 73.9% of the model size of 3 MB and a relatively short training time. KNN and Random Forest show the same pattern of results as in the previous table. The extreme SVM training time is

expected given the exhaustive 5-Fold Grid Search exploring nine hyperparameter combinations. This procedure forces the model to be trained 45 times per experiment [33]. When combined with high-dimensional EfficientNetV2 and ResNet feature vectors, the computational load increases dramatically, causing SVM to require several hours of processing compared with only minutes for the other classifiers [34].

Considering the implementation needs of edge devices, model size and computing efficiency are the main considerations. ANN shows consistency in both things. The model size is very small, with 2 MB on EfficientNetV2 and 3 MB on ResNet50 and the training time is relatively low. ANN can therefore be considered the most suitable candidate for deployment on devices with limited resources. In contrast, SVMs with Grid Search CV are less suitable for edge devices because they require a very long training process and are relatively large in model size. Meanwhile, the KNN is also less than ideal because the large model size makes it less efficient when run on low-power devices.

Overall, ANN with features from EfficientNetV2 and ResNet50 provides the best balance between model size accuracy and training efficiency. If the priority is implementation on edge devices, then ANN combined with EfficientNetV2 features can be seen as the most optimal choice.

Discussion

The results of this study highlight the central role of feature extraction architecture in determining the performance of microscopic blood cell classification. EfficientNetV2 consistently produced the highest accuracy across all classifiers, indicating that its compound scaling strategy captures richer spatial and morphological cues compared to ResNet50. This advantage is reflected not only in the quantitative results but also in the confusion matrices, where EfficientNetV2–based models achieve clearer class separability despite exhibiting slightly wider error dispersion. The broader dispersion occurs because the higher dimensional feature space produced by EfficientNetV2 leads to more complex decision boundaries, which require the Support Vector Machine to construct wider margins across multiple high-variance directions. In contrast, ResNet50 generates a more compact representation, resulting in a more concentrated misclassification pattern, consistent with the expected margin behaviour of SVMs on lower-dimensional embeddings.

Across classifiers, SVM with Grid Search CV and the Artificial Neural Network showed the most stable and well-balanced performance. SVM benefits from its ability to optimize decision boundaries in high-dimensional spaces, and its performance advantage is most visible when combined with EfficientNetV2 features. The ANN also performed competitively, leveraging its nonlinear model capacity while maintaining a very small model size, which makes it particularly suitable for deployment on devices with limited computational resources. In contrast, KNN and Random Forest achieved lower accuracy and displayed more dispersed errors in the confusion matrices, indicating their limitations in handling complex feature structures extracted from convolutional backbones.

The per-class precision and recall results further reveal that classes with high morphological similarity such as Neutrophils and Monocytes, or Eosinophils and Basophils remain the most challenging. This observation aligns with previous hybrid learning studies that report similar difficulties when dealing with subtle cytoplasmic and nuclear variations [35], [36]. The pattern of misclassifications demonstrates that although segmentation-based preprocessing improves morphological focus, texture and granularity remain crucial cues for distinguishing certain cell types.

The correction of the training time for the EfficientNetV2 + SVM Grid Search also clarifies the computational trade-offs. Initially exaggerated due to accumulated logging across folds, the corrected values place the required computation time in a realistic range and more comparable to findings from related hybrid studies. Prior work has similarly observed that SVM training becomes substantially more expensive when feature dimensionality increases, particularly when extensive hyperparameter grids are evaluated. The ANN, in contrast, demonstrated far shorter training times and smaller memory requirements, strengthening its role as the most deployment-efficient classifier in this hybrid framework.

Overall, the findings reinforce the viability of hybrid pipelines that combine deep feature extraction with lightweight machine learning classifiers. This strategy provides strong accuracy while avoiding the computational burden of end-to-end CNN training. For practical deployment such as in portable devices, point-of-care tools, or clinics with limited hardware EfficientNetV2 paired with a small ANN offers the best balance between performance, model size, memory footprint, and inference efficiency. Future work may explore fine-tuning of backbone networks, attention-based feature enhancement, or multi-level feature fusion to further improve robustness and clinical reliability across diverse medical imaging settings.

Conclusion:

This study evaluated a hybrid framework for blood cell image classification by integrating deep feature extraction using EfficientNetV2 and ResNet50 with classical machine learning classifiers, including SVM, KNN, ANN, and Random Forest. The experimental findings show that the EfficientNetV2 + SVM (Grid Search CV) combination achieved the strongest overall performance, attaining a testing accuracy of 76.85% and an F1-Score of 73.62%, with the EfficientNetV2 + ANN model performing closely behind. These results reinforce the capability of EfficientNetV2 to generate highly discriminative and compact feature representations, which facilitate more accurate differentiation among blood cell types compared to features extracted from ResNet50.

Overall, the proposed hybrid approach demonstrates that pairing deep feature extraction with lightweight classifiers offers an effective balance between accuracy and computational efficiency, providing a practical solution for deployment in low-resource clinical settings. A key limitation of this study is the reliance on Otsu-based masks without manual annotation, which may restrict the granularity of morphological features. Future work may incorporate partial fine-tuning of the CNN backbones and feature fusion from multiple layers to enhance robustness and class separability.

Furthermore, the study provides a reproducible benchmark for hybrid frameworks, highlighting the potential for rapid implementation in small clinics or point-of-care hematology devices. By demonstrating that frozen deep features combined with classical classifiers can deliver competitive performance with lower computational demand, this work paves the way for accessible, lightweight diagnostic tools suitable for resource-constrained environments.

References:

[1] K. Barrera, J. Rodellar, S. Alférez, and A. Merino, “A deep learning approach for automatic recognition of abnormalities in the cytoplasm of neutrophils,” *Comput. Biol. Med.*, vol. 178, p. 108691, Aug. 2024, doi: 10.1016/j.compbiomed.2024.108691.

[2] M. Bećirović, A. Kurtović, N. Smajlović, M. Kapo, and A. Akagić, "Performance comparison of medical image classification systems using TensorFlow Keras, PyTorch, and JAX," July 19, 2025, *arXiv*: arXiv:2507.14587. doi: 10.48550/arXiv.2507.14587.

[3] Y. Zhang, C. Li, Z. Liu, and M. Li, "Semi-Supervised Disease Classification Based on Limited Medical Image Data," *IEEE J. Biomed. Health Inform.*, vol. 28, no. 3, pp. 1575–1586, Mar. 2024, doi: 10.1109/JBHI.2024.3349412.

[4] J. Su *et al.*, "Cervical cancer prediction using machine learning models based on routine blood analysis," *Sci. Rep.*, vol. 15, no. 1, p. 22655, July 2025, doi: 10.1038/s41598-025-08166-0.

[5] M. Hussein and F. A. E.-S. Z. El-Mougi, "Integrating deep learning and transfer learning: optimizing white blood cells classification in medical educational institutions," *J. Big Data*, vol. 12, no. 1, p. 189, July 2025, doi: 10.1186/s40537-025-01235-1.

[6] A. Panthakkan, S. M. Anzar, W. Mansoor, and H. A. Ahmad, "A new frontier in hematology: Robust deep learning ensembles for white blood cell classification," *Biomed. Signal Process. Control*, vol. 100, p. 106995, Feb. 2025, doi: 10.1016/j.bspc.2024.106995.

[7] Ş. N. Özcan, T. Uyar, and G. Karayeğen, "Comprehensive data analysis of white blood cells with classification and segmentation by using deep learning approaches," *Cytometry A*, vol. 105, no. 7, pp. 501–520, July 2024, doi: 10.1002/cyto.a.24839.

[8] J. Yang *et al.*, "MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Sci. Data*, vol. 10, no. 1, p. 41, Jan. 2023, doi: 10.1038/s41597-022-01721-8.

[9] J. Yang, R. Shi, and B. Ni, "MedMNIST classification decathlon: a lightweight AutoML benchmark for medical image analysis," in *IEEE 18th international symposium on biomedical imaging (ISBI)*, 2021, pp. 191–195.

[10] M. Elgendi *et al.*, "The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective," *Front. Med.*, vol. 8, p. 629134, Mar. 2021, doi: 10.3389/fmed.2021.629134.

[11] A. Kebaili, J. Lapuyade-Lahorgue, and S. Ruan, "Deep Learning Approaches for Data Augmentation in Medical Imaging: A Review," *J. Imaging*, vol. 9, no. 4, p. 81, Apr. 2023, doi: 10.3390/jimaging9040081.

[12] J. Lo, J. Cardinell, A. Costanzo, and D. Sussman, "Medical Augmentation (Med-Aug) for Optimal Data Augmentation in Medical Deep Learning Networks," *Sensors*, vol. 21, no. 21, p. 7018, Oct. 2021, doi: 10.3390/s21217018.

[13] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *J. Med. Imaging Radiat. Oncol.*, vol. 65, no. 5, pp. 545–563, Aug. 2021, doi: 10.1111/1754-9485.13261.

[14] F. Allender, R. Allègre, C. Wemmert, and J.-M. Dischler, "Data augmentation based on spatial deformations for histopathology: An evaluation in the context of glomeruli segmentation," *Comput. Methods Programs Biomed.*, vol. 221, p. 106919, June 2022, doi: 10.1016/j.cmpb.2022.106919.

[15] I. Ahmed, E. Balestrieri, A. Neyestani, F. Picariello, and S. Rapuano, "Image segmentation techniques for morphometric measurement of fish blood cells: A comparative study," *Meas. Sens.*, vol. 38, p. 101654, May 2025, doi: 10.1016/j.measen.2024.101654.

[16] Y. Fang and B. Zhong, "Cell segmentation in fluorescence microscopy images based on multi-scale histogram thresholding," *Math. Biosci. Eng.*, vol. 20, no. 9, pp. 16259–16278, 2023, doi: 10.3934/mbe.2023726.

[17] X.-H. Lam, K.-W. Ng, Y.-J. Yoong, and S.-B. Ng, "WBC-based segmentation and classification on microscopic images: a minor improvement," *F1000Research*, vol. 10, p. 1168, Nov. 2021, doi: 10.12688/f1000research.73315.1.

[18] G. Ye and M. Kaya, "Automated Cell Foreground–Background Segmentation with Phase-Contrast Microscopy Images: An Alternative to Machine Learning Segmentation Methods with Small-Scale Data," *Bioengineering*, vol. 9, no. 2, p. 81, Feb. 2022, doi: 10.3390/bioengineering9020081.

[19] L. Alzubaidi *et al.*, "Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study," *Appl. Sci.*, vol. 10, no. 13, p. 4523, June 2020, doi: 10.3390/app10134523.

[20] E. da S. Puls, M. V. Todescato, and J. L. Carbonera, "An evaluation of pre-trained models for feature extraction in image classification," Oct. 03, 2023, *arXiv*: arXiv:2310.02037. doi: 10.48550/arXiv.2310.02037.

[21] N. Ho and Y.-C. Kim, "Evaluation of transfer learning in deep convolutional neural network models for cardiac short axis slice classification," *Sci. Rep.*, vol. 11, no. 1, p. 1839, Jan. 2021, doi: 10.1038/s41598-021-81525-9.

[22] S. Tayebi Arasteh, L. Misera, J. N. Kather, D. Truhn, and S. Nebelung, "Enhancing diagnostic deep learning via self-supervised pretraining on large-scale, unlabeled non-medical images," *Eur. Radiol. Exp.*, vol. 8, no. 1, p. 10, Feb. 2024, doi: 10.1186/s41747-023-00411-3.

[23] H. Darwis, R. Puspitasari, Purnawansyah, W. Astuti, D. Atmajaya, and M. Hasnawi, "A Deep Learning Approach for Improving Waste Classification Accuracy with ResNet50 Feature Extraction," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2025, pp. 1–8. doi: 10.1109/IMCOM64595.2025.10857536.

[24] K. Kansal, T. B. Chandra, and A. Singh, "ResNet-50 vs. EfficientNet-B0: Multi-Centric Classification of Various Lung Abnormalities Using Deep Learning," *Procedia Comput. Sci.*, vol. 235, pp. 70–80, 2024, doi: 10.1016/j.procs.2024.04.007.

[25] J. M. H. Pinheiro *et al.*, "The Impact of Feature Scaling In Machine Learning: Effects on Regression and Classification Tasks," Sept. 22, 2025, *arXiv*: arXiv:2506.08274. doi: 10.48550/arXiv.2506.08274.

[26] S. Notley and M. Magdon-Ismail, "Examining the Use of Neural Networks for Feature Extraction: A Comparative Analysis using Deep Learning, Support Vector Machines, and K-Nearest Neighbor Classifiers," June 12, 2018, *arXiv*: arXiv:1805.02294. doi: 10.48550/arXiv.1805.02294.

[27] F. Al-Areqi and M. Z. Konyar, "Effectiveness evaluation of different feature extraction methods for classification of covid-19 from computed tomography images: A high accuracy classification study," *Biomed. Signal Process. Control*, vol. 76, p. 103662, July 2022, doi: 10.1016/j.bspc.2022.103662.

[28] A. Salhi, R. Alshamrani, A. Althbiti, A. Ismail, M. Abd-ElRahman, and B. M. Hassan, "Optimizing high dimensional data classification with a hybrid AI driven feature selection framework and machine learning schema," *Sci. Rep.*, vol. 15, no. 1, p. 35038, Oct. 2025, doi: 10.1038/s41598-025-08699-4.

[29] Herman, H. Darwis, Nurfauziyah, R. Puspitasari, D. Widyawati, and A. Faradibah, "Comparative Analysis of Anxiety Disorder Classification Using Algorithm Naïve Bayes, Decision Tree and K-NN," in *2025 19th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2025, pp. 1–6. doi: 10.1109/IMCOM64595.2025.10857485.

[30] "Classification: Accuracy, recall, precision, and related metrics | Machine Learning," Google for Developers. Accessed: Oct. 28, 2025.

[31] "Understanding the Confusion Matrix in Machine Learning," GeeksforGeeks. Accessed: Oct. 28, 2025.

[32] R. Omar, J. Bogner, H. Muccini, P. Lago, S. Martínez-Fernández, and X. Franch, "The More the Merrier? Navigating Accuracy vs. Energy Efficiency Design Trade-Offs in Ensemble Learning Systems," July 03, 2024, *arXiv*: arXiv:2407.02914. doi: 10.48550/arXiv.2407.02914.

[33] M. A. K. Raiaan *et al.*, "A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks," *Decis. Anal. J.*, vol. 11, p. 100470, June 2024, doi: 10.1016/j.dajour.2024.100470.

[34] K.-L. Du, B. Jiang, J. Lu, J. Hua, and M. N. S. Swamy, "Exploring Kernel Machines and Support Vector Machines: Principles, Techniques, and Future Directions," *Mathematics*, vol. 12, no. 24, p. 3935, Dec. 2024, doi: 10.3390/math12243935.

[35] M. Wageh, K. Amin, A. D. Algarni, A. M. Hamad, and M. Ibrahim, "Brain Tumor Detection Based on Deep Features Concatenation and Machine Learning Classifiers With Genetic Selection," *IEEE Access*, vol. 12, pp. 114923–114939, 2024, doi: 10.1109/ACCESS.2024.3446190.

[36] K. V. Naveen, B. N. Anoop, K. S. Siju, M. K. Kar, and V. Venugopal, "EffNet-SVM: A Hybrid Model for Diabetic Retinopathy Classification Using Retinal Fundus Images," *IEEE Access*, vol. 13, pp. 79793–79804, 2025, doi: 10.1109/ACCESS.2025.3566073.