RESEARCH PAPER

A Comparative Performance Evaluation of Deep-Learning Architectures for Eight-Class Blood Cell Image Classification

Rajalakshmi Sankaran[1*], Lakshmanan M[2], R. Aiyshwariya Devi[3], R. Helen[4]

[1]Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Pennalur, Tamil Nadu, India.

[2]Department of Computer Science and Engineering (AI&ML), School of Engineering, Dayananda Sagar University, Bengaluru, Karnataka, India.

[3]Department of Artificial Intelligence and Data Science, R.M.K. College of Engineering and Technology, Puduvoyal, Tamil Nadu, India.

[4]Department of Biomedical Engineering, Saveetha Engineering College, Chennai, Tamil Nadu, India.

***Corresponding Author:** Rajalakshmi Sankaran, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Pennalur, Tamil Nadu, India.| Email: rajiaslakshmi@gmail.com

Abstract

Automated blood-cell image classification offers a practical test case for assessing how convolutional neural networks learn visual differences among closely related biological categories. This study compares five CNN-based models on BloodMNIST: a custom Simple CNN, ResNet-18, EfficientNet, DenseNet, and a lightweight MobileNet-style network. The experiment follows the standard MedMNIST data split, comprising 11,959 images for training, 1,712 for validation, and 3,421 for testing. Images are represented as 28 × 28 RGB arrays, pixel values are scaled to [0, 1], labels are one-hot encoded, and training-time augmentation and class weighting are applied. Performance is assessed using test accuracy, macro-averaged precision, recall and F1-score, multi-class ROC-AUC, loss, parameter count, and class-wise recall. Among the evaluated architectures, ResNet-18 delivered the strongest results, achieving 95.18% test accuracy, a macro F1-score of 0.9465, and an AUC of 0.9970. EfficientNet attained 93.66% accuracy, followed by DenseNet at 91.99% and the Simple CNN at 88.31%. In comparison, the MobileNet-style model showed substantially weaker performance, with an accuracy of 18.24% and a macro F1-score of 0.0386. The comparison indicates that residual learning provided the strongest balance of class-level and aggregate performance in the recorded run.

Keywords: BloodMNIST; blood-cell image classification; convolutional neural network; ResNet-18; EfficientNet; DenseNet; MobileNet; comparative evaluation; macro F1-score; ROC-AUC

How to cite this article: Sankaran R, Lakshmanan M, Devi RA, Helen R., A Comparative Performance Evaluation of Deep-Learning Architectures for Eight-Class Blood Cell Image Classification. Int J Drug Deliv Technol. 2026;16(90s): 2114-2126; DOI: 10.25258/ijddt.16.90s.211

1. Introduction

Blood-cell classification from microscopic images is a well-established image-recognition problem with practical relevance to digital microscopy and computational haematology. A cell image contains several visual cues, including shape, colour distribution, texture, nucleus morphology, and cytoplasmic appearance. These cues can overlap across categories, so a classifier must learn combinations of features rather than rely on a single visual characteristic. Computer-vision methods offer a consistent way to study this task, provided that the evaluation is transparent and that performance is examined at both aggregate and class levels.

CNNs perform image classification by learning basic visual features through convolutional operations and progressively combining these features to form more complex representations. However, CNN architectures differ in their connectivity, parameter efficiency, and optimisation behaviour. A compact custom network may be easier to train and deploy, whereas deeper architectures may capture richer representations. Residual connections, dense feature reuse, and depthwise separable convolutions represent different design choices that can influence the accuracy–complexity trade-off.

Comparisons are most useful when the candidate models are evaluated on the same data partitions and with the same metrics. Reporting only accuracy can conceal poor recognition of a less frequent class, while a single aggregate score cannot explain where errors occur. For this reason, the present study considers accuracy alongside macro precision, macro recall, macro F1-score, ROC-AUC, loss, model parameter count, and class-wise recall. This combination offers a broader view of both predictive quality and model size.

The study uses BloodMNIST, a small-image benchmark within the MedMNIST collection. The experiment loads eight cell categories and evaluates five models: Simple CNN, ResNet-18, EfficientNet, DenseNet, and MobileNet-style CNN. The purpose is comparative rather than algorithmic: no new classification algorithm is proposed. The contribution is a consolidated account of the implemented experiment, a consistent comparison of the five architectures, and an interpretation of their observed class-wise behaviour.

1.1 Research objectives

This work focuses on four aspects: recording the dataset and preprocessing procedures, evaluating the five architectures using multiple test metrics, assessing whether overall scores reflect performance across all eight blood-cell classes, and investigating the association between model complexity and the results obtained in the experiment.

1.2 Scope of the Research

This is a computational benchmark using an existing public dataset. It does not involve patient recruitment, new specimen collection, or clinical deployment. The results reflect the models' performance on the benchmark test dataset and do not establish their suitability for clinical use.

2. Literature Review

2.1 Benchmark datasets for biomedical image classification

A benchmark provides a shared dataset and evaluation framework through which competing methods can be compared under defined conditions. Yang et al. developed MedMNIST v2, which provides a standardised set of two-dimensional and three-dimensional biomedical image datasets with predefined training, validation, and testing splits [1]. Within this collection, BloodMNIST comprises 17,092 images of individual normal blood cells distributed across eight categories, with each image scaled to 28 × 28 pixels. The consistent dataset structure facilitates reproducible comparisons between classification models; however, the limited image resolution may restrict the fine-grained cellular features that models can capture [1], [7]. Previous studies in deep-learning-based medical image analysis have likewise emphasised the need for suitable datasets, rigorous validation procedures, and careful interpretation of evaluation results [8], [9].

2.2 Convolutional neural networks

CNNs extract visual features from images by applying convolutional operations, introducing nonlinear activation functions, and reducing spatial dimensions through downsampling. Early layers generally learn local patterns, while deeper layers combine these into more complex features. CNNs have become a common approach in medical image analysis because they can learn discriminative representations directly from images [8], [9]. A compact custom CNN is useful as a baseline in this study because it provides a reference against which more elaborate architectures can be compared.

2.3 Residual networks

He et al. proposed residual learning, in which shortcut connections allow a block to learn a residual mapping relative to its input [2]. These connections facilitate optimisation in deeper networks and made ResNet a widely used image-classification backbone. In the evaluation, ResNet-18 achieved the best test-set accuracy of 95.18%, with a corresponding macro F1-score of 0.9465. This result indicates that the residual architecture performed well under the specific training configuration; it does not establish universal superiority across datasets.

2.4 EfficientNet

EfficientNet was introduced as a family of networks that jointly scales depth, width, and input resolution to balance accuracy and computational cost [3]. The EfficientNet-based architecture achieved 93.66% accuracy and a macro F1-score of 0.9290 on the test set. Its results were comparable to those of ResNet-18, indicating that it is a competitive option within this benchmark. However, these findings should be considered in relation to the low-resolution input images and the particular architectural configuration and training procedure adopted in this research.

2.5 DenseNet

DenseNet connects each layer to subsequent layers within a dense block, encouraging feature reuse and information flow through the network [4]. The DenseNet-style implementation in this study achieved 91.99% test accuracy with a reported parameter count of 105,536. This result is relevant when considering model size alongside classification performance. However, parameter count alone does not provide a complete measure of computational efficiency; inference latency, memory consumption, and hardware requirements would also be needed for a practical deployment comparison.

2.6 MobileNet and efficient convolution

MobileNet reduces computational requirements by employing depthwise separable convolutions instead of standard convolution operations [5]. A similar architectural approach is adopted by Xception, where depthwise separable convolution forms a fundamental component of the network design [6]. These approaches are relevant to applications with limited computing resources. The MobileNet-style model in the present study produced a substantially lower score than the other candidates, with class-wise outputs suggesting a prediction collapse. This result is treated as an implementation-specific anomaly rather than evidence that the MobileNet family is inherently unsuitable.

2.7 Evaluation beyond accuracy

Accuracy is easy to interpret but may conceal poor performance on individual classes. Precision and recall provide complementary information, while the F1-score combines precision and recall. Macro-averaging calculates each class's metric independently and assigns equal importance to all classes, making it suitable for evaluating multiclass classification tasks. ROC-AUC measures discrimination across decision thresholds, while confusion matrices show which classes are being confused. Literature reviews on deep learning for medical image analysis stress the importance of selecting evaluation methods appropriate to the intended task and distinguishing benchmark results from evidence of clinical validity [8], [9]. Accordingly, this study considers accuracy, macro F1-score, AUC, class-wise recall, confusion matrices, and learning curves.

2.8 Positioning of the present study

Research on blood-cell and other biomedical image classification illustrates the use of deep architectures for tasks ranging from segmentation to subtype classification. Berlin Shaheema et al. proposed a hybrid Dense-Gated 3D U-Net for diabetic-retinopathy lesion severity identification [10]. Although that work addresses retinal lesions rather than blood-cell classification, it illustrates how hybrid deep-learning designs can be tailored to a specific medical-image task.

Mohan et al. reported an Inception U-Net approach for brain-tumour segmentation [11], providing another example of adapting CNN components to biomedical image analysis. In a related blood-cell application, Mohan et al. investigated efficient CNNs for acute lymphoblastic leukaemia prediction [12]. This study is closely associated with blood-cell image analysis, but its specific classification task and dataset should be distinguished from those of BloodMNIST. Other established

literature includes CNN architectures for image recognition and medical-image analysis, including residual learning, dense connectivity, efficient scaling, and separable convolutions [2]–[9]. Taken together, these studies motivate a controlled comparison of established CNN families on a shared benchmark. The present study does not propose a new architecture; its contribution is a transparent comparison of five candidates using common data partitions and complementary evaluation metrics.

3. Research Design and Methodology

3.1 Study design and data source

The research is a comparative computational study based on BloodMNIST as accessed through MedMNIST. The research work partitions and uses the same held-out test split for the final evaluation of all five models. The dataset is divided into eight categories: Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, and Platelet.

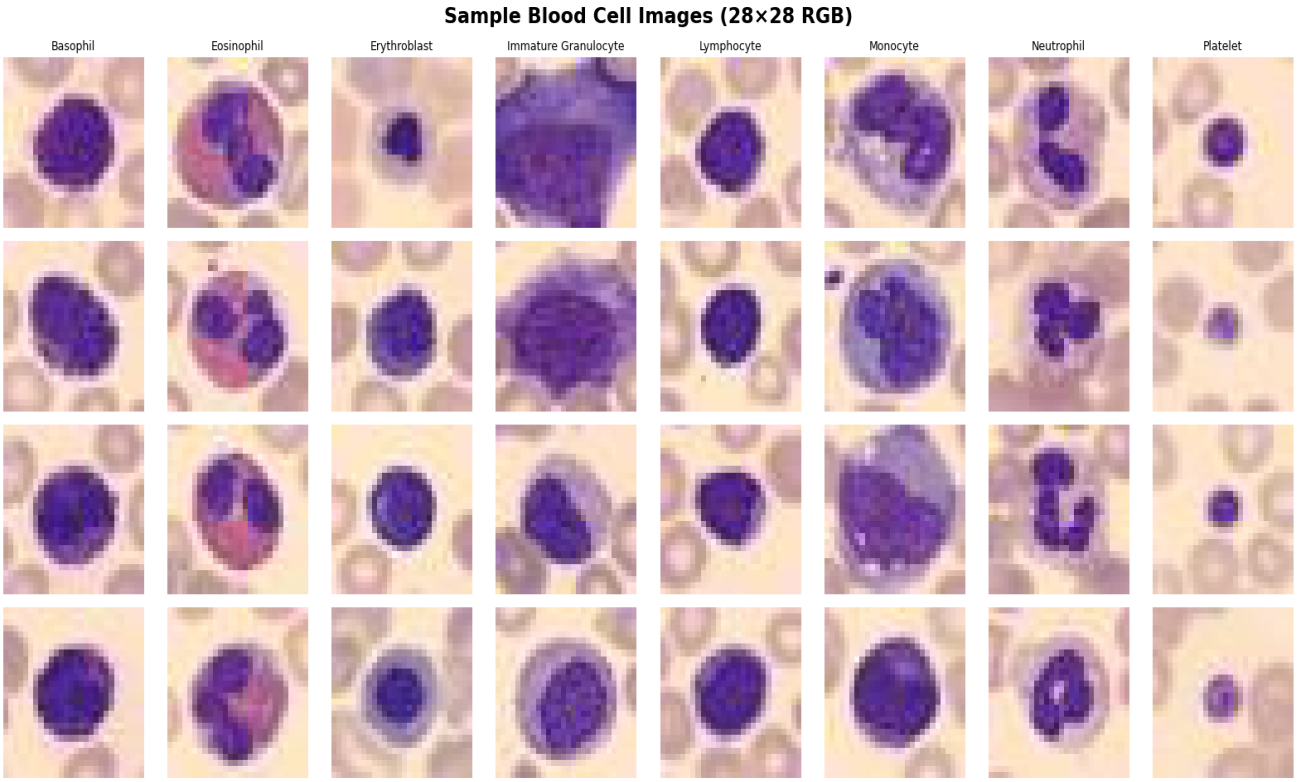

Figure 1. Representative BloodMNIST images illustrating the eight blood-cell categories.

Table 1. Distribution of BloodMNIST Images Across Training, Validation, and Test Partitions

Partition	Number of images	Role in experiment
Training	11,959	Parameter fitting and training-time augmentation
Validation	1,712	Model selection and checkpoint monitoring
Test	3,421	Final comparative evaluation
Total	17,092	All predefined partitions

Table 1 summarizes the allocation of BloodMNIST images among the three data subsets used in this experiment. Of the 17,092 images in the dataset, 11,959 are assigned to the training set, 1,712 to the validation set, and the remaining 3,421 to the test set. The training partition is used for model parameter fitting and training-time data augmentation, while the validation partition supports model selection and checkpoint monitoring.

The test partition is reserved for the final comparative evaluation of the five CNN architectures. This predefined partitioning provides a common evaluation setting for all models and supports a consistent comparison of their classification performance.

The use of fixed partitions supports a more direct comparison than independently splitting the data for each model. The validation split is used during training to monitor performance and select checkpoints; the test split is reserved for the final reported metrics.

3.2 Data preparation and label encoding

The research work converts image arrays to floating-point values and divides pixel intensities by 255, mapping the original range to [0, 1]. This scaling gives the networks a consistent input range. Integer class labels are transformed into eight-dimensional one-hot vectors, matching the softmax output layer and categorical cross-entropy objective. The same normalisation and label representation are used for training, validation, and test data.

3.3 Class distribution and class weighting

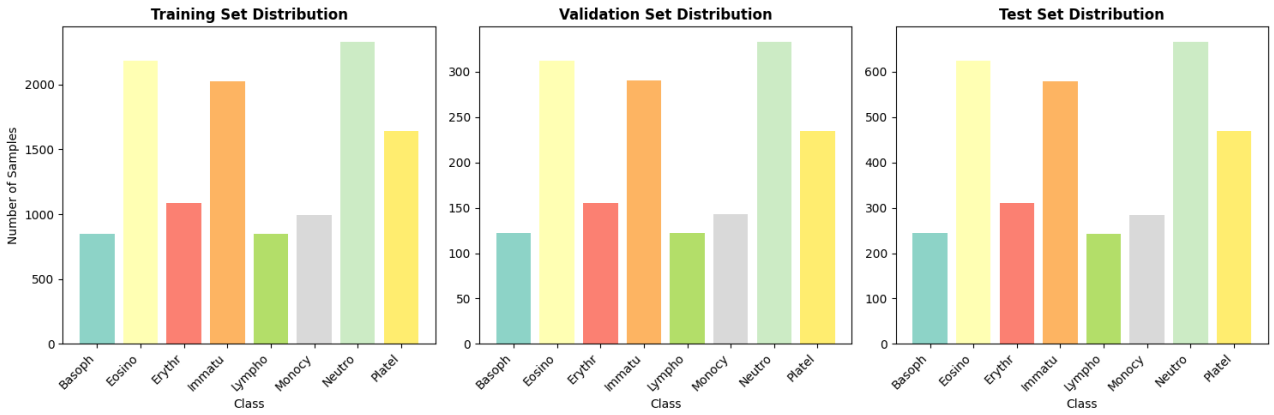

Figure 2. Image distribution among the training, validation, and testing sets.

Figure 2 presents the image distribution across the training, validation, and test subsets for the eight BloodMNIST categories: Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, and Platelet. Neutrophils have the highest number of samples in all three partitions, followed by Eosinophils and Immature granulocytes. Lymphocytes and Basophils have comparatively fewer samples. The proportions of the eight classes remain generally similar across the training, validation, and test subsets, suggesting a consistent class distribution among the three data partitions. This distribution provides an overview of class representation for evaluating the CNN models on the BloodMNIST dataset. The research work examines the distribution of classes in the available partitions and computes class weights from the training labels. These weights are intended to increase the contribution of less represented classes during optimisation. Class weighting can be useful when a model otherwise favours common categories, but it does not guarantee balanced performance. The resulting class-wise recall values are therefore reported alongside macro-averaged metrics.

3.4 Training-time augmentation

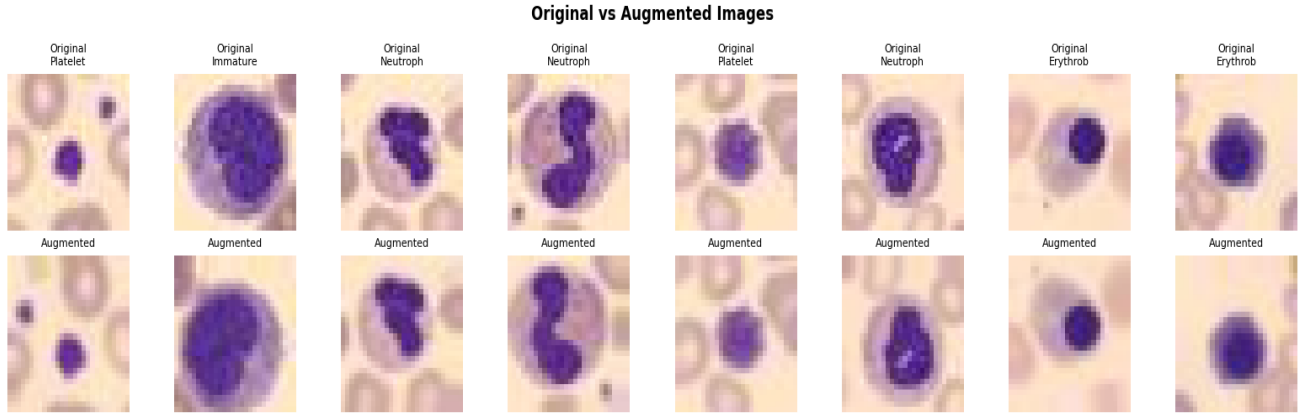

Figure 3. Examples of original BloodMNIST images and their augmented counterparts.

Figure 3 shows representative BloodMNIST blood-cell images before and after data augmentation. The top row displays the original images, while the bottom row shows their corresponding augmented versions. The examples include Platelet, Immature granulocyte, Neutrophil, and Erythroblast cells. The augmented images exhibit variations in visual appearance while retaining the main characteristics of the corresponding cell types. This illustrates the use of data augmentation to introduce diversity into the training images, potentially helping convolutional neural network (CNN) models learn more robust features and improve generalization to unseen samples. Training augmentation includes random rotation up to ±15 degrees, horizontal and vertical flips, and width and height shifts up to ±10%. The aim is to expose the models to modest variations in orientation and position so that they do not depend too strongly on the precise placement of a cell in an image. Augmentation is applied to training data only; validation and test images are evaluated without random transformations. The transformations are relatively simple and reproducible, although their biological suitability should be considered when transferring the method to another image source.

3.5 Architectures compared

Five architectures are included in the comparison. The Simple CNN acts as a compact baseline. ResNet-18 provides residual connections; EfficientNet-style layers represent an efficiency-oriented design; DenseNet-style blocks encourage feature reuse; and the MobileNet-style model uses lightweight convolutional operations. The names identify the architectural families represented in the dataset; because the implementations are adapted for small images.

Table 2 summarizes the five CNN architectures evaluated in this study, highlighting their respective roles in the comparative

analysis and their recorded trainable parameter counts. The Simple CNN serves as a compact baseline, while ResNet-18 represents residual feature learning. The EfficientNet-style model focuses on computational efficiency, the DenseNet-style model emphasizes dense feature reuse, and the MobileNet-style model represents a lightweight convolutional design. Among the listed models, ResNet-18 has the highest trainable parameter count (11,187,336), whereas the DenseNet-style model has the lowest (105,536). This comparison provides an overview of the differences in model complexity and architectural design considered in the BloodMNIST classification experiments.

Table 2. Comparison of CNN Architectures and Their Trainable Parameter Counts

Model	Role in comparison	Trainable parameter count recorded
Simple CNN	Compact baseline	391,368
ResNet-18	Residual feature learning	11,187,336
EfficientNet-style	Efficiency-oriented architecture	945,776
DenseNet-style	Dense feature reuse	105,536
MobileNet-style	Lightweight convolutional design	2,268,232

3.6 Training configuration and checkpoint selection

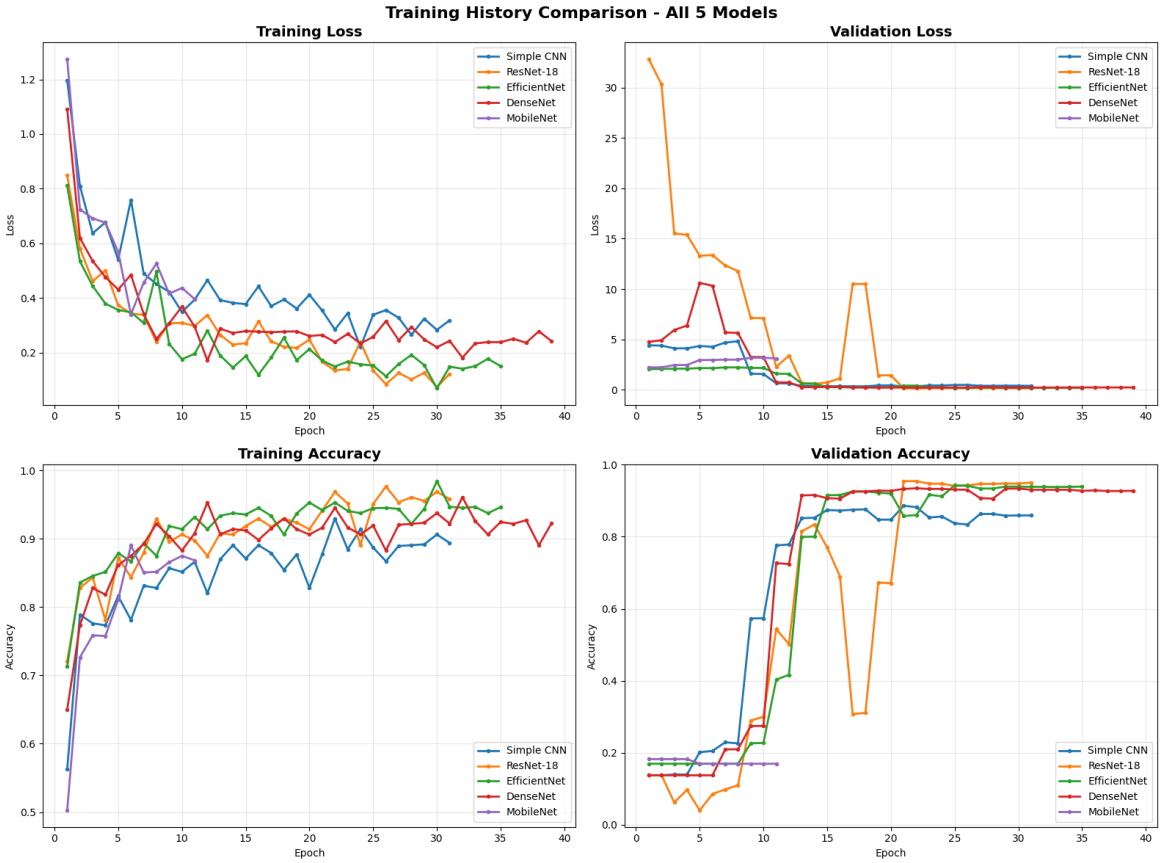

Figure 4. Training and validation loss and accuracy curves for the five assessed models.

Figure 4 depicts the changes in training and validation accuracy and loss over successive epochs for the five CNN architectures: Simple CNN, ResNet-18, EfficientNet, DenseNet, and MobileNet. The training loss generally decreases as the epochs progress, while training accuracy increases, indicating that the models learn meaningful features from the BloodMNIST dataset. The EfficientNet and ResNet-18 models achieve relatively high training accuracy, while the Simple CNN shows comparatively lower accuracy. The validation curves reveal differences in convergence behaviour among the models. ResNet-18 shows noticeable variations in validation accuracy and loss during the initial and intermediate training stages, followed by greater stability. In the later epochs, EfficientNet and DenseNet attain validation accuracy levels of approximately 93%–94%. The Simple CNN stabilizes at a comparatively lower validation accuracy of around 0.86. Overall, the figure illustrates the learning progress, convergence patterns, and differences in training stability among the five CNN architectures.

The experimental study uses TensorFlow 2.19.0 and MedMNIST 3.0.2. It defines shared training settings, including an epoch limit of up to 50, an Adam optimiser, categorical cross-entropy, and accuracy as a training metric. Callbacks are configured to monitor validation behaviour, including early stopping, learning-rate reduction, and checkpoint saving. The

best checkpoint is restored for test evaluation. This strategy reduces the chance of reporting the final epoch simply because it was the last one, although the exact selected epoch and final hyperparameters should be exported from a clean run for full reproducibility.

3.7 Evaluation metrics

The comparison uses the following measures:

Accuracy: Proportion of test images assigned the correct class.

Macro precision: Arithmetic mean of per-class precision values.

Macro recall: Arithmetic mean of per-class recall values.

Macro F1-score: Arithmetic mean of per-class F1-scores.

Multi-class ROC-AUC: One-vs-rest AUC aggregated across classes.

Test loss: Categorical cross-entropy evaluated on the test partition.

Parameter count: Number of model parameters reported by the model implementation.

Class-wise recall: Proportion of test examples from each class correctly identified.

All five models are evaluated on the same test images using a shared evaluation function. Predicted class probabilities are converted to class labels for metrics that require discrete predictions, while the probability scores are used for ROC analysis. Confusion matrices and classification reports provide a class-level view.

4. Results and Quantitative Evaluation

4.1 Overall test-set performance

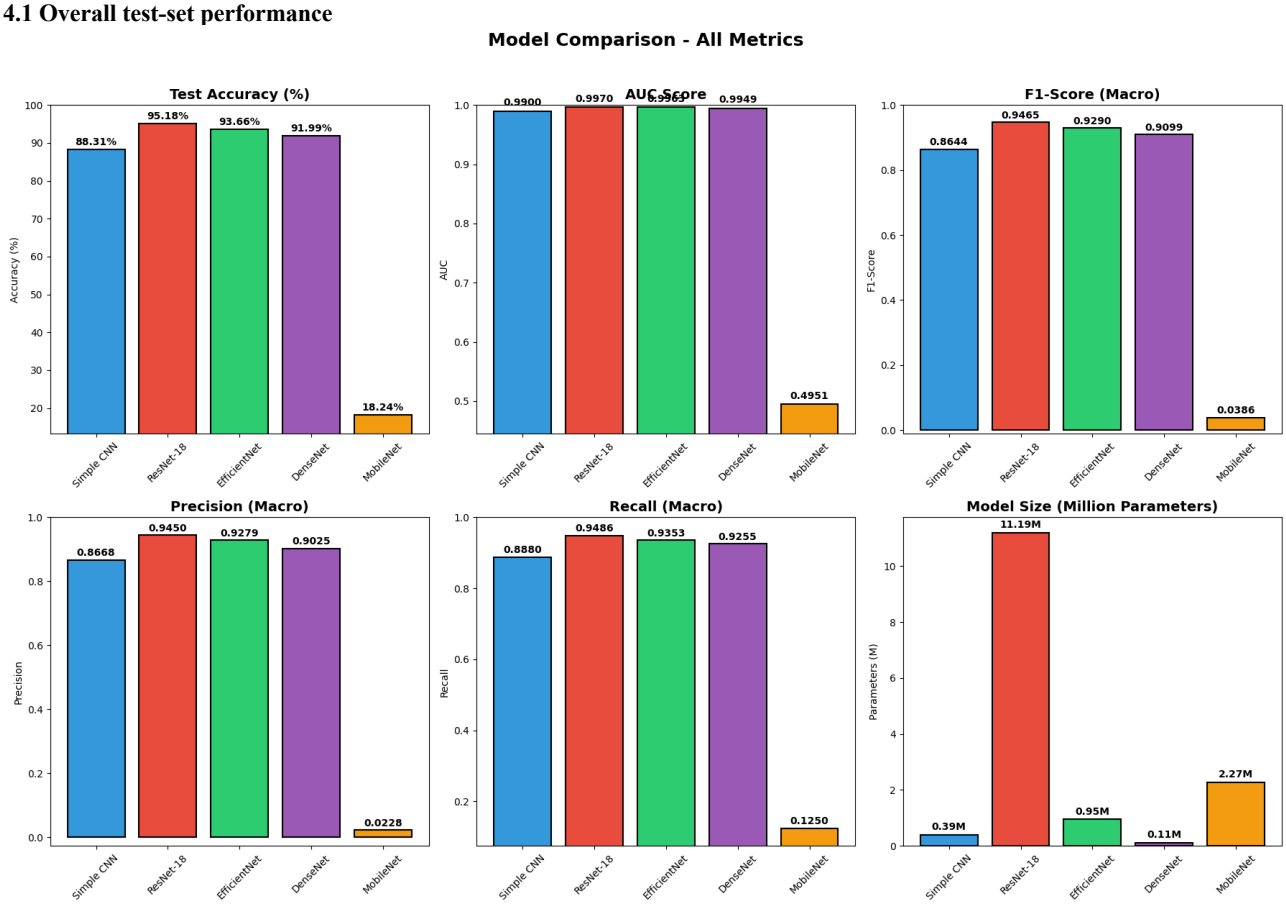

Figure 5. Comparative test metrics for the evaluated architectures, including accuracy, AUC, F1-score, precision, recall, and model parameters.

Table 3 summarizes the five models using the values recorded in the experiment. ResNet-18 ranks first on test accuracy, macro precision, macro recall, macro F1-score, and AUC. EfficientNet is the second strongest model by accuracy and macro F1-score. DenseNet exceeds the Simple CNN on the aggregate metrics, while MobileNet has markedly lower scores than all other models. The five CNN architectures are compared using several test-set measures, including accuracy, area under the ROC curve (AUC), macro-precision, macro-recall, macro F1-score, test loss, and the number of trainable parameters. ResNet-18 achieves the highest classification accuracy (95.18%), AUC (0.9970), macro-precision (0.9450), macro-recall (0.9486), and macro-F1 score (0.9465), along with the lowest test loss (0.1493). EfficientNet and DenseNet also demonstrate strong classification performance, with accuracies of 93.66% and 91.99%, respectively. The Simple CNN achieves an accuracy of 88.31%, providing a baseline for comparison. In contrast, MobileNet records substantially lower performance, with an accuracy of 18.24%, an AUC of 0.4951, and a test loss of 2.2134. These results indicate that ResNet-18 performs best among the evaluated models under the reported experimental settings.

Table 3. Comparative Performance Evaluation of CNN Models on the BloodMNIST Test Set

Model	Accuracy (%)	AUC	Macro precision	Macro recall	Macro F1	Test loss	Parameters
Simple CNN	88.31	0.9900	0.8668	0.8880	0.8644	0.3471	391,368
ResNet-18	95.18	0.9970	0.9450	0.9486	0.9465	0.1493	11,187,336
EfficientNet	93.66	0.9963	0.9279	0.9353	0.9290	0.1789	945,776
DenseNet	91.99	0.9949	0.9025	0.9255	0.9099	0.2256	105,536
MobileNet	18.24	0.4951	0.0228	0.1250	0.0386	2.2134	2,268,232

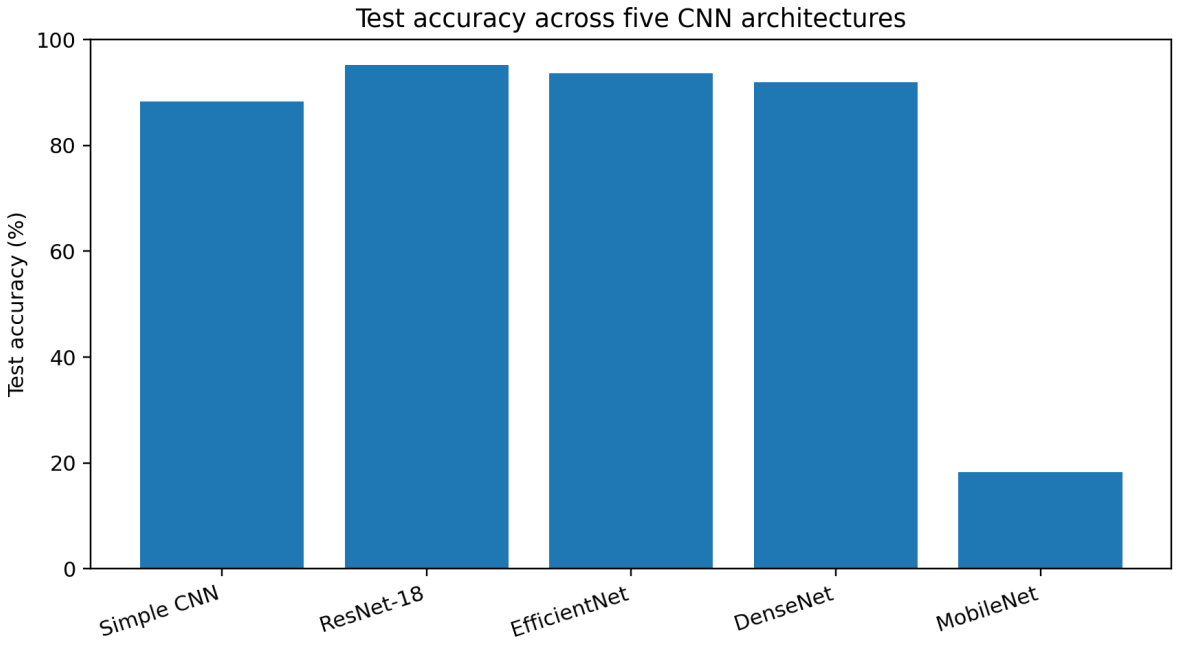

Figure 6. Test accuracy reported for the five models.

The macro-averaged results help clarify whether the strongest model performs consistently across categories. ResNet-18 records macro precision of 0.9450, macro recall of 0.9486, and macro F1-score of 0.9465. EfficientNet's corresponding values are 0.9279, 0.9353, and 0.9290. DenseNet records 0.9025, 0.9255, and 0.9099, while the Simple CNN records 0.8668, 0.8880, and 0.8644. In this run, the ranking is broadly consistent across all three macro measures. MobileNet's macro precision of 0.0228, macro recall of 0.1250, and macro F1-score of 0.0386 indicate that its low accuracy is accompanied by very poor class-level performance. This pattern is important because it shows that the low result is not simply a modest trade-off between model size and accuracy.

4.2 Precision, recall, and F1-score

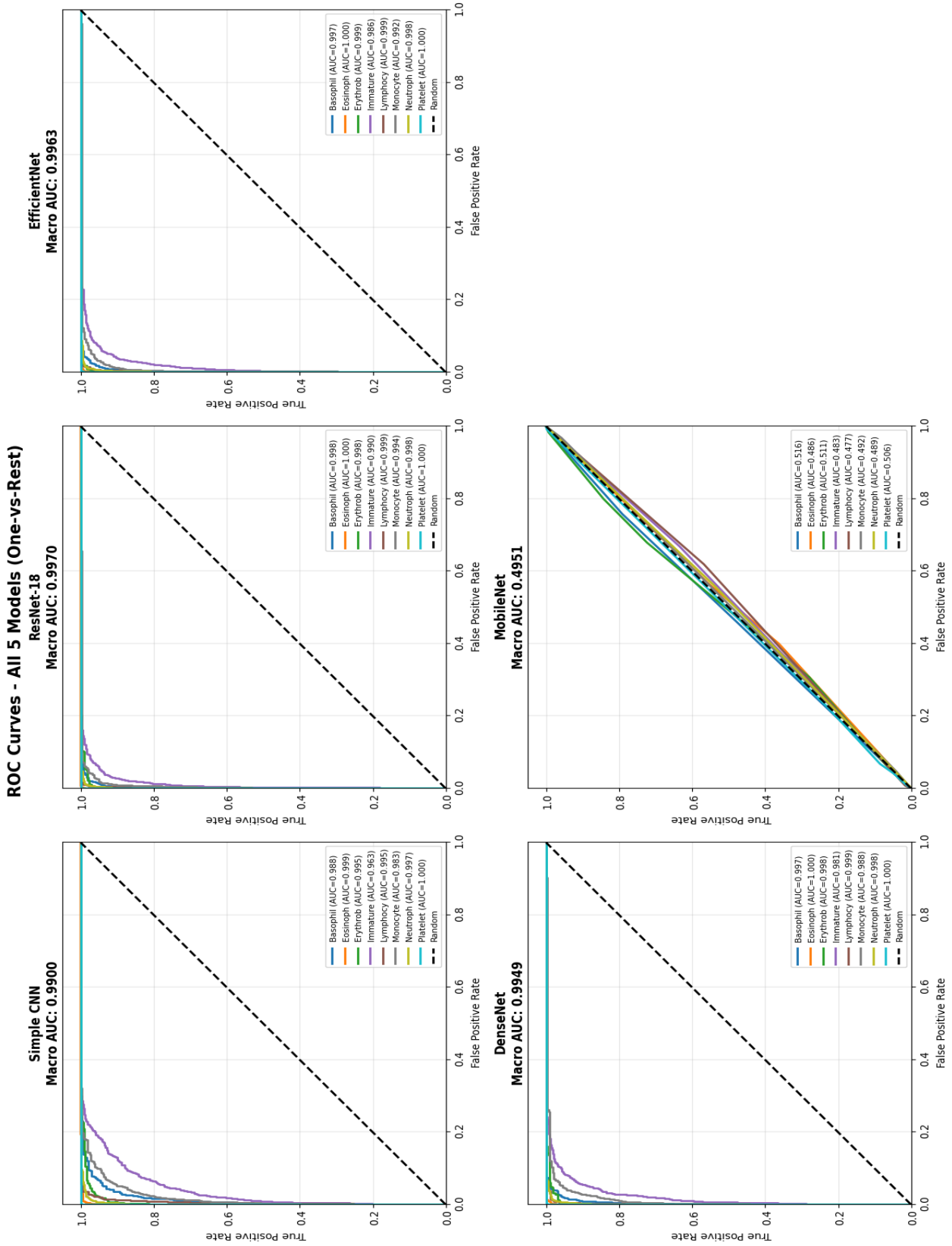

Figure 7. One-vs-rest ROC curves for the five evaluated architectures.

4.3 Class-wise recall

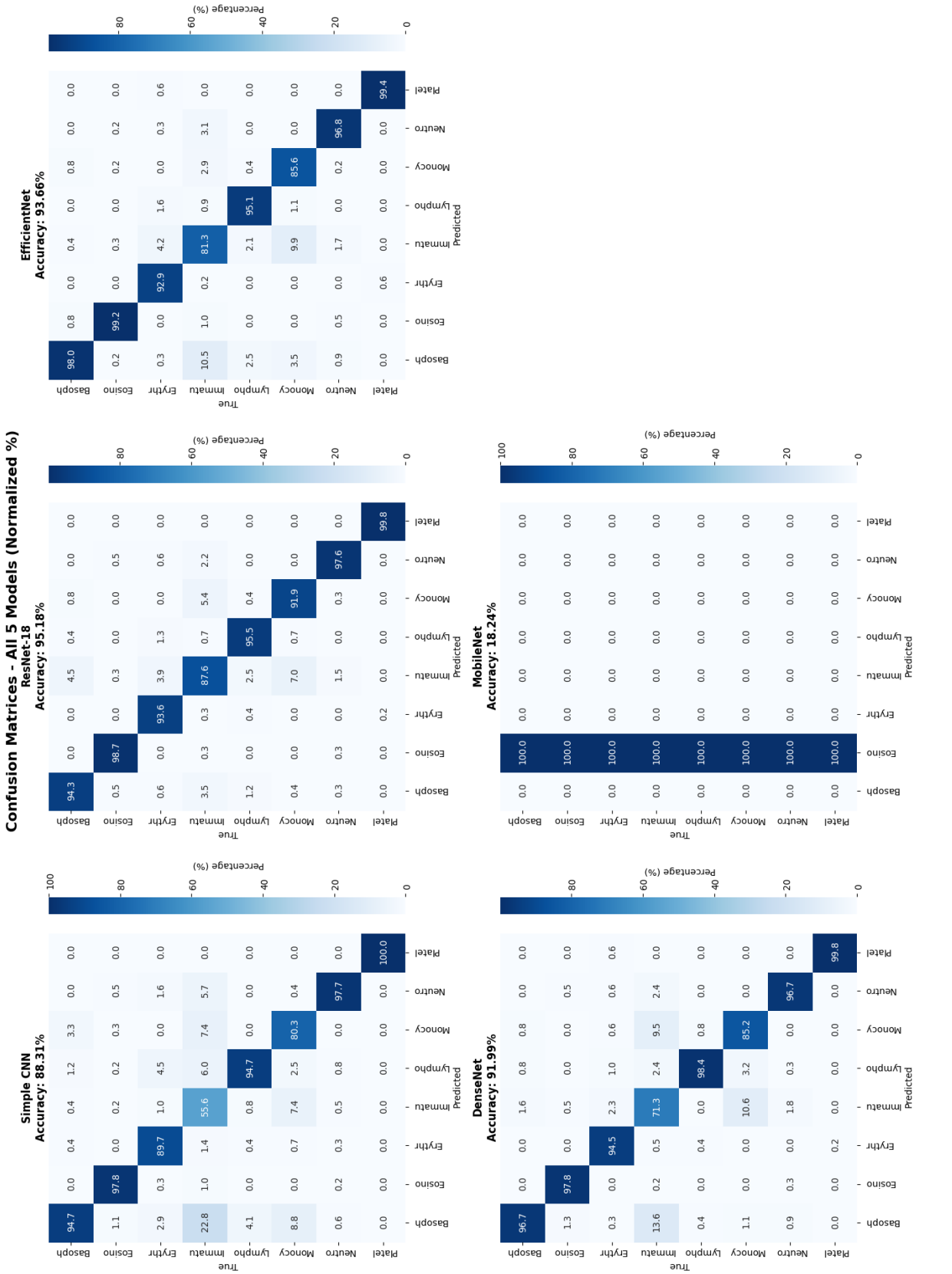

Figure 8. Confusion matrices for the evaluated models, showing class-level prediction patterns on the test set.

Figure 7 displays the ROC curves of the five CNN architectures—Simple CNN, ResNet-18, EfficientNet, DenseNet, and MobileNet—for the eight BloodMNIST classes using a one-versus-rest classification strategy. These curves show how the true-positive rate varies with the false-positive rate as the classification threshold changes. Simple CNN, ResNet-18, EfficientNet, and DenseNet show curves approaching the upper-left corner, indicating strong class discrimination, with macro AUC values of 0.9900, 0.9970, 0.9963, and 0.9949, respectively. Among these models, ResNet-18 achieves the highest macro AUC of 0.9970.

In contrast, MobileNet records a macro AUC of 0.4951, and its ROC curves lie close to the diagonal reference line, indicating discrimination close to chance level. Overall, the ROC analysis supports the strong discriminative performance of four models, while MobileNet's substantially lower AUC.

Figure 8 compares the normalized confusion matrices of five deep-learning models—Simple CNN, ResNet-18, EfficientNet, DenseNet, and MobileNet—for the eight BloodMNIST categories: Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, and Platelet. Each matrix shows the percentage of samples belonging to an actual class that are assigned to each predicted class. Darker blue cells along the main diagonal represent correct predictions, whereas off-diagonal cells indicate classification errors. Among the evaluated models, EfficientNet achieved the highest classification accuracy of 93.66%, followed by DenseNet at 91.99%, ResNet-18 at 95.18%, Simple CNN at 88.31%, and MobileNet at 18.24%. ResNet-18's reported accuracy of 95.18% is higher than EfficientNet's 93.66%, making ResNet-18 the best-performing model according to the displayed accuracy values.

ResNet-18 demonstrates strong classification performance across most classes, while EfficientNet and DenseNet also show high diagonal values for several cell types. Simple CNN exhibits comparatively greater confusion for the Immature Granulocyte and Monocyte classes. MobileNet predominantly predicts a single class, resulting in poor overall classification performance. Overall, the figure highlights differences in class-wise prediction capability and indicates that ResNet-18 provides the highest reported overall accuracy among the five models.

Figure 9 shows the Class-wise recall where each model succeeds and where errors remain. ResNet-18 records 87.56% recall for Immature Granulocyte, compared with 55.61% for the Simple CNN, 81.35% for EfficientNet, and 71.33% for DenseNet. This is the largest and most notable separation among the listed class-wise results. For the other categories, the four functioning models generally record high recall, although there are still differences between architectures. Platelet recall is near 100% for all four, while Monocyte recall varies more substantially. The class-wise results support the use of macro metrics in this comparison. A model may obtain a strong overall score while still struggling with a particular category. The Simple CNN's 55.61% recall for Immature Granulocyte is one example. ResNet-18 has a substantially higher recall for that class and maintains high recall across the remaining categories, which helps explain its macro F1 advantage.

Figure 10 presents the radar chart where the selected metrics on a common visual summary. Because metrics have different interpretations and scales, it is best used as a compact overview rather than as a substitute for the numerical results.

Table 4. Comparison of Class-Wise Classification Accuracy (%) Across Five Deep-Learning Models for Blood Cell Images.

Class	Simple CNN	ResNet-18	EfficientNet	DenseNet	MobileNet
Basophil	94.67	94.26	97.95	96.72	0.00
Eosinophil	97.76	98.72	99.20	97.76	100.00
Erythroblast	89.71	93.57	92.93	94.53	0.00
Immature granulocyte	55.61	87.56	81.35	71.33	0.00
Lymphocyte	94.65	95.47	95.06	98.35	0.00
Monocyte	80.28	91.90	85.56	85.21	0.00
Neutrophil	97.75	97.60	96.85	96.70	0.00
Platelet	100.00	99.79	99.36	99.79	0.00

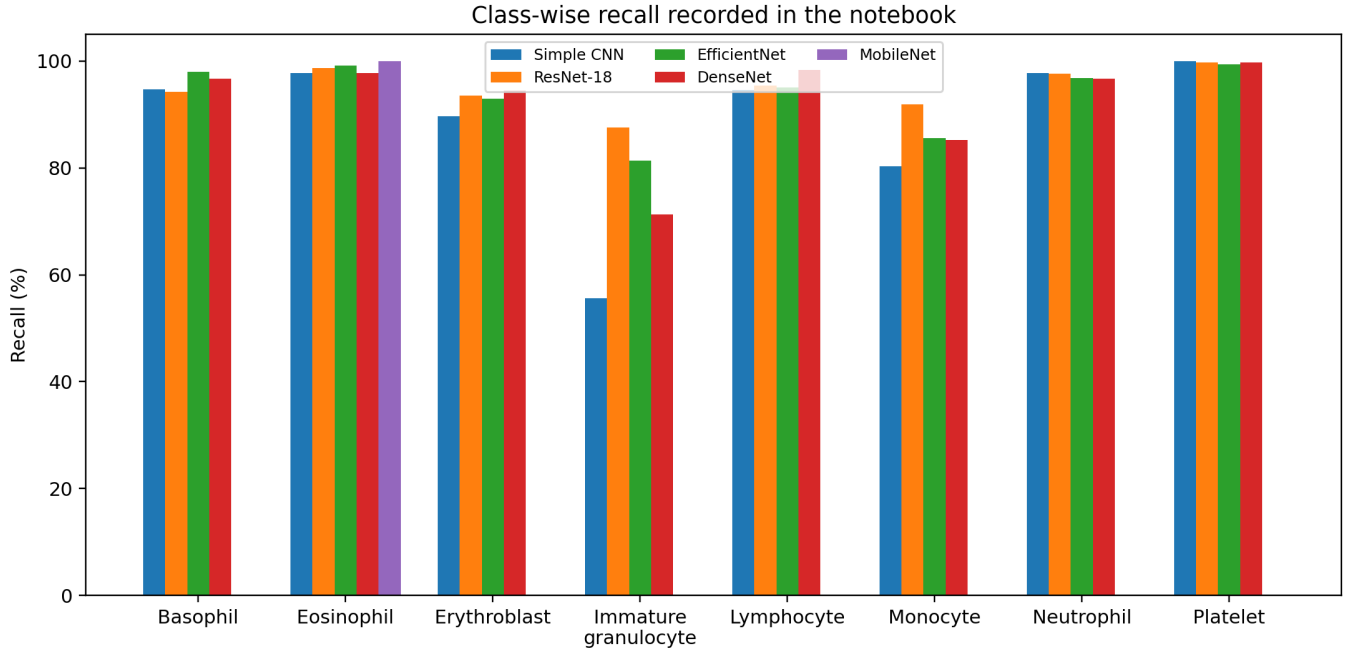

Figure 9. Class-wise recall values

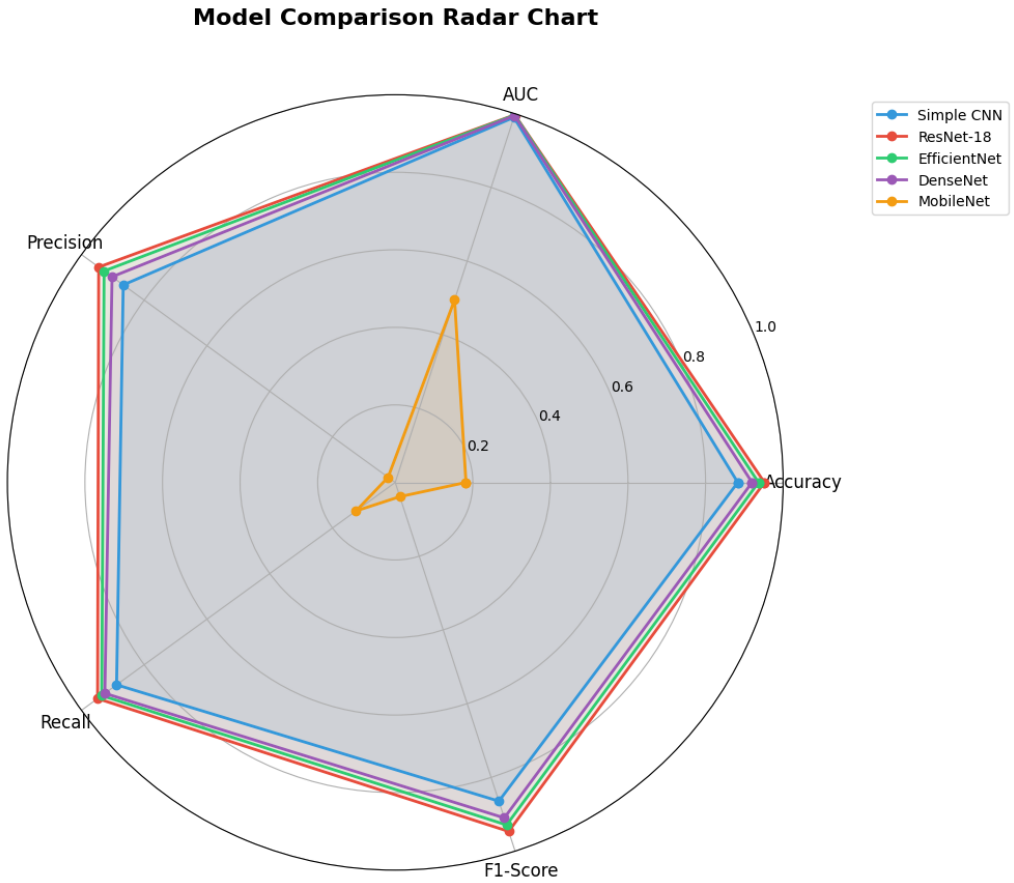

Figure 10. Radar-chart comparison of model metrics.

4.4 Model size and performance trade-off

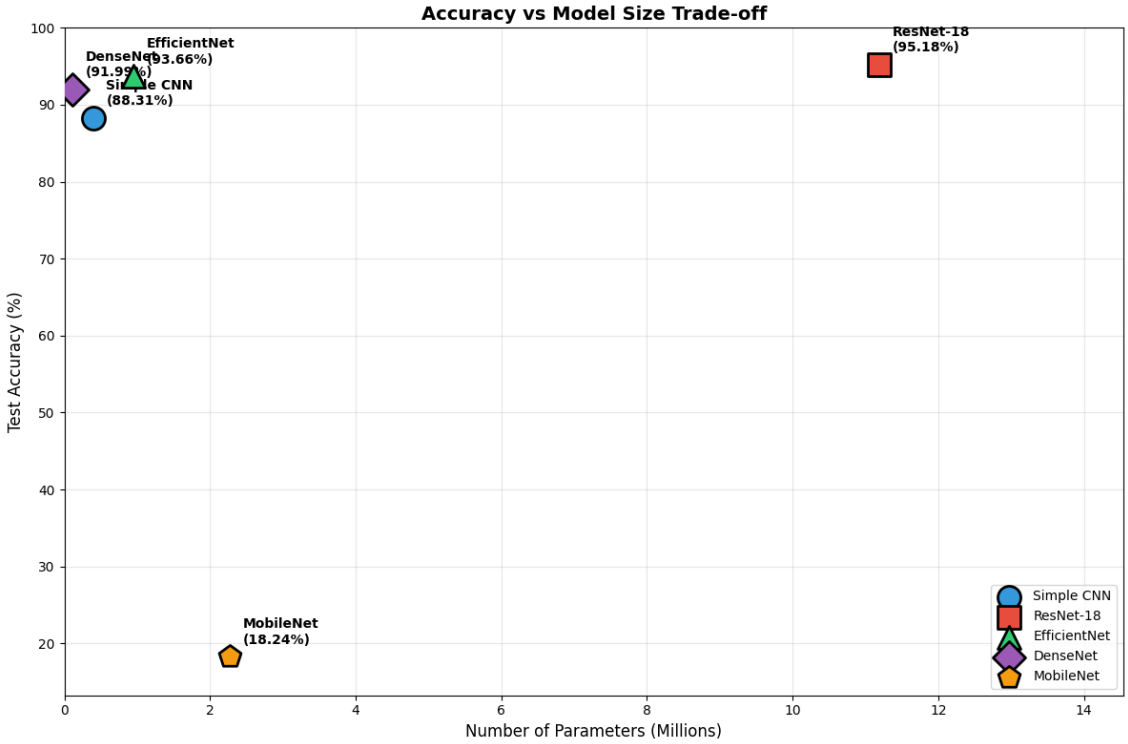

Figure 11. Relationship between model parameter count and test accuracy

Figure 11 shows that the ResNet-18 achieves the strongest recorded predictive performance but also has the largest parameter count, at 11.19 million. EfficientNet has fewer than one million parameters and achieves 93.66% accuracy, making it an attractive candidate when parameter count matters. DenseNet has 105,536 parameters and reaches 91.99% accuracy, the most favourable accuracy result among the smallest models in this table. The Simple CNN has 391,368 parameters and 88.31% accuracy. These comparisons illustrate that model size and accuracy do not move in a simple linear relationship.

Parameter count is only a proxy for computational cost. It does not directly measure runtime, peak memory, energy use, or latency on a particular device. Those measurements would be needed to make a strong deployment-oriented claim. Based on the metrics available here, ResNet-18 is the accuracy-oriented choice, while EfficientNet and DenseNet are worth further evaluation where model size is an important constraint.

4.5 Interpretation of the MobileNet result

MobileNet's recorded accuracy is 18.24%, its AUC is 0.4951, and its macro F1-score is 0.0386. The class-wise recall table reports 100% recall for Eosinophil and 0% for each of the other seven categories. This pattern is consistent with a model that predicts a single class for nearly all test images. The result must therefore be treated as a diagnostic finding rather than a fair estimate of the best achievable performance of MobileNet-like networks.

5. Discussion

The recorded comparison places ResNet-18 first on the principal aggregate metrics. Its 95.18% test accuracy and 0.9465 macro F1-score suggest that the residual-style model learned features that generalised well to the held-out BloodMNIST images under the experiment's training configuration. The advantage is also visible in class-wise recall, particularly for Immature Granulocyte and Monocyte. These results make ResNet-18 the strongest candidate in the present experiment, rather than establishing it as the universally best architecture for blood-cell classification.

EfficientNet provides a close alternative. Its accuracy is 93.66%, and its macro F1-score is 0.9290, while its reported parameter count is substantially lower than ResNet-18's. This balance may be useful where memory or model size is a concern. DenseNet produces 91.99% accuracy with only 105,536 parameters, suggesting that dense feature reuse can offer a practical compromise in this particular implementation. A future comparison should add measured training time, inference latency, and memory use to quantify these trade-offs more directly.

The Simple CNN remains a useful baseline. It achieves 88.31% accuracy and an AUC of 0.9900, showing that a compact network can capture substantial discriminative information from the low-resolution images. At the same time, its recall for Immature Granulocyte is lower than that of the other functioning models, illustrating why accuracy alone is not enough. The gap between the Simple CNN and ResNet-18 is therefore meaningful both in aggregate performance and in the treatment of a difficult class.

6. Conclusion and Future Work

This study compared five CNN-based architectures for eight-class blood-cell image classification using the recorded BloodMNIST experiment. ResNet-18 delivered the strongest overall results, recording 95.18% test accuracy, a macro F1-score of 0.9465, and an AUC of 0.9970. EfficientNet ranked next, with 93.66% accuracy and a macro F1-score of 0.9290.

DenseNet attained 91.99% accuracy while maintaining a comparatively low number of model parameters, whereas the Simple CNN achieved 88.31% accuracy and served as a baseline for comparison. Class-wise recall showed that ResNet-18 handled Immature Granulocyte more effectively than the other functioning models in this run. The MobileNet-style model recorded a much lower result, with evidence of single-class prediction behaviour. The future work can extend the comparison through repeated runs with fixed random seeds, confidence intervals, measured inference time and memory use, and external evaluation on a separate dataset.

7. References

[1] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "MedMNIST v2—A large-scale lightweight benchmark for 2D and 3D biomedical image classification," Scientific Data, vol. 10, art. no. 41, 2023, doi: 10.1038/s41597-022-01721-8.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning, PMLR, vol. 97, 2019, pp. 6105–6114.

[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708, doi: 10.1109/CVPR.2017.243.

[5] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.

[7] J. Yang, R. Shi, and B. Ni, "MedMNIST classification decathlon: A lightweight AutoML benchmark for medical image analysis," in Proc. IEEE Int. Symp. Biomedical Imaging (ISBI), 2021, pp. 191–195, doi: 10.1109/ISBI48211.2021.9434062.

[8] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," Annual Review of Biomedical Engineering, vol. 19, pp. 221–248, 2017, doi: 10.1146/annurev-bioeng-071516-044442.

[9] G. Litjens et al., "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017, doi: 10.1016/j.media.2017.07.005.

[10] S. Berlin Shaheema, V. P. Kolanchinathan, S. Rajalakshmi, G. Gurumoorthy, S. B. Mohan, and J. Jasper, "Diabetic retinopathy lesions severity identification using a hybrid Dense-Gated 3D-UNet," in Proc. 2024 Int. Conf. Data Science and Network Security (ICDSNS), 2024, doi: 10.1109/ICDSNS62112.2024.10691178.

[11] S. B. Mohan, B. Ramesh, G. Gurumoorthy, and B. Kannan, "Brain tumour segmentation using deep learning method—Inception U-Net," International Journal of Medical Engineering and Informatics, 2026, doi: 10.1504/IJMEI.2026.153152.

[12] S. B. Mohan, S. Sathya, S. Rajalakshmi, G. Gurumoorthy, and R. Sivanraju, "Efficient convolutional neural networks for acute lymphoblastic leukaemia prediction in computer vision," Scientific Reports, 2025, doi: 10.1038/s41598-025-30777-w.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[14] C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[17] A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, 2009.

[18] T.-Y. Lin et al., "Focal loss for dense object detection," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.324.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), 2015.

[20] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009, doi: 10.1016/j.ipm.2009.03.002.