Blood Cell Classification Using Deep Convolutional Neural Network Architectures

Isikay Karakus†

***Abstract*—Accurate classification of peripheral blood cells is fundamental in supporting diagnostic processes within clinical haematology. With the increasing integration of artificial intelligence in biomedical workflows, deep learning has emerged as a promising approach to automate this task. This paper presents a comparative evaluation of several Convolutional Neural Network (CNN) architectures applied to the BloodMNIST dataset, which comprises over 17,000 annotated RGB microscopy images across eight distinct blood cell types. The study aims to assess the effectiveness of compact, fully trainable models that do not rely on pre-trained backbones. A total of five core architectures are implemented under consistent training conditions: a baseline CNN, a U-Net-inspired encoder, an Inception-style model, a ResNet-based Skip Connection CNN, and a hybrid Inception-Skip configuration. Each model is evaluated based on classification accuracy, number of parameters, memory footprint, and training and inference times. Among the candidates, the Skip Connection CNN achieves the highest test accuracy of 96.78%, while maintaining relatively low complexity and computational cost. The U-Net-inspired model also demonstrates a favourable trade-off between performance and resource efficiency. In contrast, hybrid configurations and lower-resolution baselines exhibit limitations in generalisation and scalability. The results underline the importance of architectural design in balancing predictive performance and computational demands, with implications for the deployment of CNN models in resource-constrained medical imaging environments.**

***Index Terms*—Blood Cell Classification, Medical Image Analysis, Convolutional Neural Networks, Skip Connections, Model Efficiency, U-Net.**

I. Introduction

Deep learning has significantly impacted biomedical image analysis, particularly in the classification of microscopy images in fields such as histopathology and haematology. One important application is the automated classification of peripheral blood cells, which plays a vital role in disease diagnosis and monitoring, enabling efficient clinical workflows and supporting early detection of abnormalities.

Despite recent progress, a key challenge remains: achieving high classification accuracy across visually similar blood cell types while ensuring computational efficiency. While convolutional neural networks (CNNs) are widely adopted for such tasks, the effect of architectural design variations on model performance in datasets like BloodMNIST [1] has not been systematically studied. Prior work often relies on computationally intensive or pretrained models [2], [3], which may fail to generalise well or be impractical for use in constrained settings.

†Department of Mathematics, University of Padova, email: isikay.karakus@studenti.unipd.it

This study addresses these gaps by implementing and evaluating five CNN-based architectures on the BloodMNIST dataset: a baseline CNN, a U-Net-inspired encoder, an Inception-based model, a ResNet-style skip connection model, and a hybrid Inception-ResNet architecture. The models are trained under uniform conditions, using data augmentation and class weighting to counteract imbalance. Standard evaluation metrics such as accuracy, precision, recall, and F1-score are employed to assess classification performance. The designs are deliberately kept modular and lightweight to ensure interpretability and end-to-end trainability, with a view toward real-world deployment.

- Implementation and evaluation of five principal CNN architectures on the BloodMNIST dataset, covering standard, encoder-based, and hybrid designs.
- Systematic comparison of model complexity, quantified by parameter count and duration of training against classification performance to evaluate efficiency and scalability.
- Consistent training configuration using class weighting and data augmentation ensures fair benchmarking and reproducibility.
- The Skip Connection CNN achieves the best performance with 96.78% accuracy, demonstrating strong generalisation across all blood cell types.
- The architectural insights are discussed in terms of their applicability to real-world biomedical image analysis systems.

All models were trained using Google Colab Pro with an L4 GPU and 22.5 GB of memory.

This paper is structured as follows. Section II reviews prior research in CNN-based biomedical imaging. Section III introduces the dataset and outlines the data processing pipeline. Section IV describes preprocessing steps and training configuration. Section V presents the architectural designs. Experimental findings and performance analysis are discussed in Section VI. Finally, Section VII concludes the work and outlines future directions.

II. Related Work

Deep learning has become the prevailing approach for biomedical image classification, providing superior performance compared to traditional feature engineering methods. In particular, Convolutional Neural Networks (CNNs) have demonstrated remarkable success in tasks such as cell classification, lesion detection, and histopathological image analysis [2].

A widely used architecture in biomedical imaging is the U-Net, originally proposed for semantic segmentation [4]. U-Net leverages an encoder-decoder structure with skip connections, enabling both local and contextual information capture. Although primarily designed for segmentation, recent work has adapted its encoder path for classification tasks to exploit its hierarchical feature extraction capability while reducing architectural complexity [5].

The Inception architecture introduced by Szegedy et al. [6], [7] proposed a novel block structure with multiple parallel convolutional filters of varying sizes to extract features at different scales. In biomedical applications, this multiscale approach has proven effective in handling variability in cell morphology and staining conditions.

Residual Networks (ResNets) introduced by He et al. [8], [9] addressed the vanishing gradient problem in deep networks by using identity-based skip connections. These residual connections help preserve low-level features and improve gradient flow during training, resulting in faster convergence and improved generalisation.

Hybrid approaches that combine Inception blocks with skip connections have also emerged, aiming to unify multi-scale representation with training stability. Such architectures have shown improved performance in medical image classification tasks leveraging the complementary strengths of both designs [10].

In this study, several deep learning models were evaluated in the BloodMNIST dataset [1], including a baseline convolutional neural network, a U-Net inspired encoder model, an Inception-based convolutional network, a convolutional network with ResNet-style skip connections, and a hybrid model that combines Inception modules with residual connections. The objective of the evaluation was to examine how different architectural design choices affect classification performance in the task of blood cell image analysis.

III. Processing Pipeline

The processing pipeline developed for this work is structured to enable a systematic evaluation of multiple deep learning architectures for blood cell image classification. A high-level overview of the workflow is provided in Fig. 1.

The pipeline begins with image acquisition and formatting, followed by standardised pre-processing steps as described in Sect. IV. After preprocessing, the dataset is divided into training, validation, and test sets, and data augmentation is applied during training to improve generalisation.

A range of convolutional neural network (CNN) architectures are then instantiated, including a baseline CNN, an augmented CNN, a U-Net inspired encoder, an Inception-based model, a skip connection model based on residual learning, and a hybrid model combining Inception modules with skip connections. All models are designed and trained using a consistent optimisation and evaluation protocol to ensure fair comparison.

Once training is completed, the performance of each model is assessed using standard classification metrics, including accuracy, precision, recall, and F1 score. This modular pipeline enables consistent benchmarking and isolates architectural choices as the key factor influencing model performance.

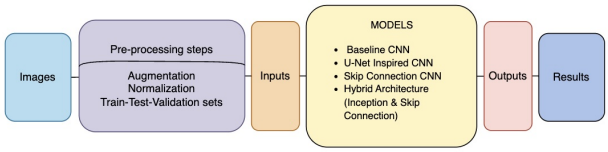

Fig. 1: Overview of the proposed processing pipeline.

IV. Images and Features

This study addresses the problem of classifying peripheral blood cell types using colour microscopy images. The dataset used is BloodMNIST, part of the MedMNIST v2 collection [1], which contains 17,092 RGB images annotated into eight distinct cell types. Each sample is originally sized at 128×128 pixels and stored as a three-channel image, where each channel corresponds to one of the RGB colour intensities.

A. *Input Images and Formatting*

The input to the system consists of fixed-size RGB image tensors with a shape of $128 \times 128 \times 3$. As an initial baseline, a downsampled version with dimensions $64 \times 64 \times 3$ was utilised. Subsequently, more advanced architectures were trained on the resolution 128×128 dataset to explore their performance in greater detail. The 256×256 version of the dataset was not used due to computational resource constraints.

B. *Class Structure and Distribution*

The dataset includes eight distinct blood cell types, each associated with a fixed class label: *Basophil (Class 0), Eosinophil (Class 1), Erythroblast (Class 2), Immature Granulocyte (Class 3), Lymphocyte (Class 4), Monocyte (Class 5), Neutrophil (Class 6), and Platelet (Class 7).* A detailed distribution of the number of samples per class is provided in Table 1. As evident from the table, the dataset is imbalanced, with Neutrophil, Monocyte, Basophil, and Immature Granulocyte being relatively underrepresented. To mitigate the impact of class imbalance during training, inverse-frequency class weighting is applied, assigning greater importance to minority classes.

Class Name	Number of Samples	Percentage (%)
Neutrophil	1218	7.13
Eosinophil	3117	18.24
Basophil	1551	9.07
Lymphocyte	2895	16.94
Monocyte	1214	7.10
Immature Granulocyte	1420	8.31
Erythroblast	3329	19.48
Platelet	2348	13.74

TABLE 1: Class distribution in the BloodMNIST dataset.

C. *Preprocessing*

Prior to training, all image pixel values were normalised to the $[0, 1]$ range by dividing each channel by 255. The dataset was provided in .npz format with predefined training, validation, and test splits, and the original class distribution was preserved to ensure consistency with the benchmark protocol. Data augmentation was applied exclusively to the training set in order to enhance generalisation. augmentation techniques included random rotations ($0°$, $90°$, $180°$, or $270°$) , horizontal flipping, and zoom shift transformations implemented through random cropping followed by resizing to the original input dimensions.

D. *Train/Validation/Test Splitting*

The dataset is provided with predefined training, validation, and test splits. For the 128×128 version, these splits consist of 11,959 training samples (70%), 1,712 validation samples (10%), and 3,421 test samples (20%). The partitions are stratified to preserve the original class distribution within each subset. These predefined splits are used consistently across all models to ensure fair and comparable performance evaluation.

V. Learning Framework

This section presents the overall learning framework used to classify blood cell images from the BloodMNIST dataset. A supervised learning approach is adopted, where various convolutional neural network (CNN) architectures are trained using consistent training protocols to ensure fair performance comparison. The framework includes architectural designs, training configuration, class imbalance handling, and evaluation metrics.

A. *Training Configuration*

All models were trained using the Adam optimiser with a learning rate of 1×10^{-4}. The loss function used is sparse categorical cross-entropy, appropriate for multi-class classification with integer labels. Training was conducted for a maximum of 40 epochs using a batch size of 64. To avoid overfitting and reduce computational cost, **early stopping** was implemented with a patience of 5 epochs terminating training if the validation loss did not improve over five consecutive epochs. A *ReduceLROnPlateau* **learning rate scheduler** was employed to reduce the learning rate by a factor of 0.5 after three stagnant epochs without validation loss improvement. Training times were monitored, and the best model weights were restored automatically.

B. *Class Weighting for Imbalance Compensation*

The dataset exhibits significant class imbalance (see Tab. 1). To address this, class weights were computed using the *'compute_class_weight'* function from Scikit-learn library in 'balanced' mode. These weights were passed to the loss function during training, ensuring that underrepresented classes were penalised less and received greater attention during optimisation.

C. *Model Architectures*

1) ***Baseline CNN Model****:* The baseline CNN is composed of three convolutional blocks with progressively increasing filter sizes (32, 64, and 128), each followed by Batch Normalisation, Max Pooling, and Dropout layers. After the convolutional blocks, the feature maps are flattened into a 1D vector and passed through a dense layer with 128 neurons and ReLU activation, followed by a high dropout rate (0.5) for further regularisation. Finally, a Dense output layer with 8 units and softmax activation produces a probability distribution over the 8 classes. This model was trained on both 64×64 and 128×128 images.

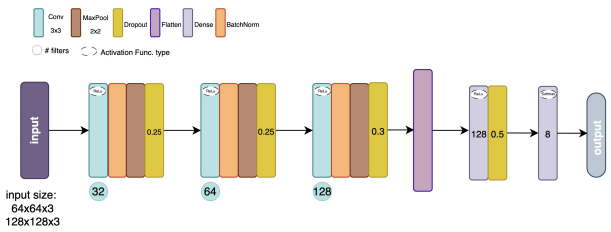

Fig. 2: Architecture of Baseline CNN Model

2) ***Enhanced Baseline CNN Model****:* This variant of the baseline CNN retains the same architecture but includes real-time data augmentation on the 128×128 images. The training process was enhanced by random rotations of 90 degrees (via *tf.image.rot90*), horizontal flips, and zoom shift transformations using random cropping (retaining 90-100% of the original area) followed by resizing. These augmentations introduced greater variability into the training data, better simulating real-world imaging conditions and improving generalisation, particularly for underrepresented or visually similar classes.

3) ***Simplified U-Net-Inspired CNN****:* While the full U-Net includes a symmetric decoder path and skips connections for pixel-wise predictions [4], [5], this version retains only the contracting (encoder) path, which enables deep hierarchical feature extraction at multiple resolutions. The model processes 128×128 RGB images through three convolutional blocks with increasing filter sizes (32, 64, and 128). Each block includes a convolutional layer, batch normalization, max pooling, and progressively increasing dropout rates (0.1, 0.15, 0.2) to control overfitting. The extracted features are flattened and passed through a fully connected layer with 64 units and a final softmax output for 8-class classification. Compared to the baseline CNN, this model offers reduced complexity and improved feature abstraction. The architecture is presented in Fig. 3.

4) ***Inception-based CNN Model****:* To enhance multi-scale feature extraction in blood cell classification, an Inception-based CNN was implemented, inspired by the original Inception (GoogLeNet) architecture [6], [7]. This model processes 128×128 RGB images through an initial convolution and pooling layer, followed by two stacked Inception blocks. Each

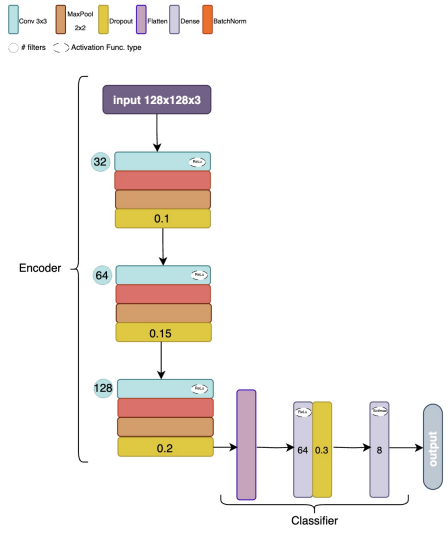

Fig. 3: Architecture of the U-Net-Inspired CNN model.

block combines parallel convolutional pathways, including 1×1, 3×3, and 5×5 filters, as well as max pooling with projection, allowing the network to capture both fine-grained and spatially contextual features simultaneously. Outputs from these branches are concatenated, preserving resolution-specific representations. After spatial downsampling, a global average pooling layer is used to reduce the risk of overfitting and replace fully flattened layers. The classification head consists of a dense layer with 64 units and ReLU activation, followed by dropout(0.3), and a final softmax layer for 8-class prediction.

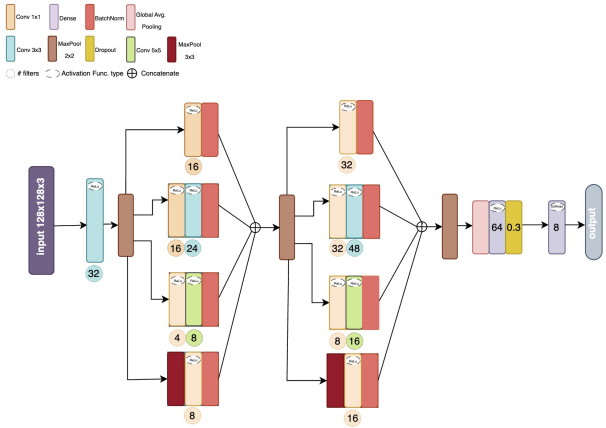

Fig. 4: Architecture of Inception-based CNN Model

5) ***Skip Connection CNN Model****:* This ResNet-inspired model introduces skip connections in the second and third convolutional blocks to facilitate residual learning and improve gradient flow. Each residual block has two convolutional layers with batch normalisation and a 1×1 projection shortcut for dimension matching. Following the convolutional layers, global average pooling, a dense layer with 64 units, dropout, and a softmax output layer complete the model. The Skip Connection CNN model architecture is shown in Fig. 5.

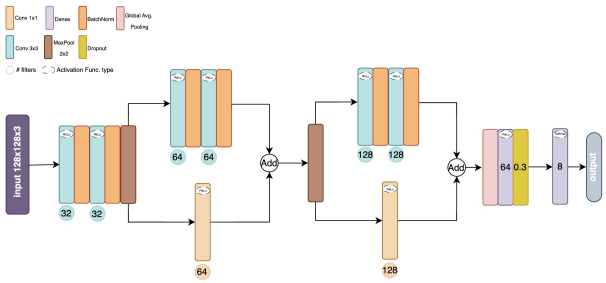

Fig. 5: Architecture of Skip Connection CNN Model

6) ***Inception + Skip Connection Hybrid Model****:* To combine the strengths of multi-scale feature extraction and stable gradient flow, a hybrid CNN architecture integrating Inception blocks with residual (skip) connections was implemented. This model, inspired by Inception [6] and ResNet [8], processes 128×128 RGB images through an initial convolution and pooling layer, followed by two Inception blocks. Each block consists of four parallel branches 1×1, 3×3, and 5×5 convolutions, and 3×3 max pooling with 1×1 projection, designed to capture features at different spatial scales. The outputs are concatenated and combined with a skip connection from the block input, projected via a 1×1 convolution to match the output tensor shape. This element-wise addition facilitates residual learning and improves information flow. After downsampling via max pooling, a classification head comprising global average pooling, a dense layer with 64 ReLU-activated units, dropout regularisation, and a softmax output layer is used for 8-class prediction.

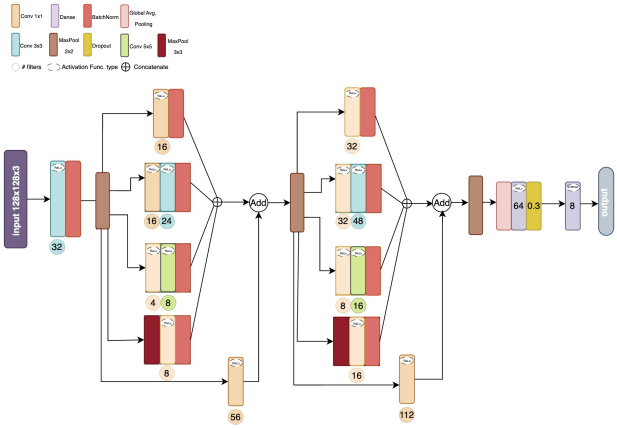

Fig. 6: Architecture of Inception & Skip Connection CNN Model

D. *Evaluation Metrics*

Performance of the trained models was assessed using standard classification metrics: accuracy, precision, recall, and F1-score. These were calculated globally and per class using the following formulations:

- **Accuracy**:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precision**:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall**:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-score**:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Confusion matrices and classification reports were also generated to visualise performance differences and detect patterns in misclassifications.

VI. Results

This section presents a comparative evaluation of all implemented CNN architectures on the BloodMNIST dataset. Models were assessed based on test accuracy, macro-averaged F1-score, class-wise performance, and computational cost. All experiments were conducted using a Google Colab Pro environment equipped with an L4 GPU and 22.5 GB of memory. The BloodMNIST dataset, comprising 17,092 images, was split into training, validation, and test sets with ratios of 70%, 10%, and 20%, respectively.

A. *Overall Performance Comparison*

The classification performance of each model, along with the corresponding training and inference times, is summarised in Table 2. As the complexity of the architectures increases, particularly with the inclusion of skip connections, clear improvements in accuracy and generalisation performance are observed. The *Skip Connection CNN* achieved the highest test accuracy of 96.78%, with a training time of 235 seconds and an inference time of 6.01 seconds. While this model required the longest training duration, its test-time latency was shorter than that of both the Inception and Inception + Skip CNN models, indicating a strong balance between predictive performance and computational efficiency.

The U-Net Inspired CNN followed with 96.40% accuracy and demonstrated a faster training time of 153 seconds and lower test latency (4.38 seconds). The Baseline CNN with augmentation achieved 95.88% accuracy, showing the benefits of augmentation despite requiring slightly more training time than the unaugmented Baseline CNN (128×128).

Inception and Inception + Skip CNNs, though more architecturally complex, reached slightly lower accuracies of 95.03% and 94.53%, respectively, and also suffered from the longest inference times, exceeding 6.4 seconds. The Baseline CNN (64×64) was the fastest to train (31 seconds) and test (3.89 seconds), but had the lowest test accuracy of 88.66%, illustrating the trade-off between speed and accuracy.

Model	Accuracy (%)	Epochs	Train Time (s)	Test Time (s)
Baseline CNN (64$\times$64)	88.66	9	31.13	3.89
Baseline CNN (128$\times$128)	94.45	25	154.59	4.42
Baseline CNN + Augmentation	95.88	29	176.90	4.52
U-Net Inspired CNN	96.40	25	153.26	4.38
Inception CNN	95.03	30	181.25	6.49
Skip Connection CNN	**96.78**	**20**	**235.07**	**6.01**
Inception + Skip CNN	94.53	32	227.08	6.73

TABLE 2: Test accuracy, training epochs, and timing metrics for each CNN architecture.

B. *Evaluation Metrics*

Table 3 shows the macro-averaged precision, recall, and F1-score values for all evaluated CNN models. These metrics provide a balanced view of model performance across all blood cell classes, mitigating the influence of class imbalance.

The *Skip Connection CNN* and the *Inception CNN* obtained the highest macro-averaged precision, recall, and F1-score (0.97 for each), indicating highly generalisable and consistent classification capability. The U-Net Inspired CNN also demonstrated strong performance, achieving macro scores of 0.95. Notably, the Baseline CNN with augmentation improved upon the unaugmented version, confirming the benefits of data augmentation. Although the Inception + Skip CNN was architecturally more sophisticated, its performance on the macro-averaged metrics was worse than simpler models. This may be due to the combination of features not working effectively together or because the model was too complex and overfitted to the training data.

Model	Precision	Recall	F1-score
Baseline CNN (64×4)	0.89	0.87	0.88
Baseline CNN (128×28)	0.94	0.94	0.94
Baseline CNN + Augmentation	0.96	0.96	0.96
U-Net Inspired CNN	0.95	0.95	0.95
Inception CNN	**0.97**	**0.97**	**0.97**
Skip Connection CNN	**0.97**	**0.97**	**0.97**
Inception + Skip CNN	0.89	0.87	0.88

TABLE 3: Macro-averaged precision, recall, and F1-score for all CNN architectures.

C. *Memory and Model Size*

Table 4 compares the models in terms of parameter count and estimated memory consumption. This analysis aids in understanding the resource trade-offs associated with different architectures. While the Baseline CNN (128×128) and its augmented variant required the most memory (16.36 MB), the Inception and Inception + Skip models were significantly lighter, consuming under 1 MB. The *Skip Connection CNN* again demonstrated a balanced profile, offering high accuracy and macro-averaged metrics while requiring only 1.17 MB of memory.

D. *Class-wise Performance Analysis*

Basophil. Basophil classification varied notably across models. The *Skip Connection CNN* achieved the highest

Model	Parameters (M)	Memory Usage (MB)
Baseline CNN (64×4)	4.29	4.36
Baseline CNN (128×28)	4.29	16.36
Baseline CNN + Augmentation	4.29	16.36
U-Net Inspired CNN	2.19	8.36
Inception CNN	0.04	0.14
Skip Connection CNN	0.31	1.17
Inception + Skip CNN	0.05	0.17

TABLE 4: Parameter count and estimated memory usage of CNN architectures.

accuracy (99.59%) and perfect recall (1.00), with a strong F1-score of 0.93. The *U-Net Inspired CNN* yielded the best F1-score (0.97) with balanced precision (0.95) and recall (0.99). Data augmentation improved baseline performance from 0.92 to 0.96. The *Inception CNN* also performed well (F1: 0.93), while the *Baseline CNN (64×64)* underperformed (F1: 0.82, Acc: 85.25%). The *Inception + Skip CNN* achieved high recall (0.99) but a lower F1-score (0.87), suggesting reduced precision. Overall, the U-Net Inspired and Skip Connection CNNs were most effective models for this class.

Eosinophil. Eosinophil classification showed consistently strong results across all models. The Baseline CNN (64×64) had the weakest performance (Recall: 0.90, Accuracy: 90.38%), but increasing resolution to 128×128 significantly improved accuracy to 99.36% and F1-score to 1.00. Data augmentation further boosted results. The U-Net Inspired, Inception, Skip Connection, and Inception + Skip CNNs all achieved perfect metrics (Precision, Recall, F1-score: 1.00), with the last two reaching the highest accuracy (99.84%). These results indicate that Eosinophils are easily distinguishable, especially with deeper or hybrid architectures.

Erythroblast. Model performance for Erythroblasts varied notably with input resolution. The Baseline CNN (64×64) showed the weakest results (Accuracy: 84.24%, Recall: 0.84) despite perfect precision, indicating frequent false negatives. Raising the resolution to 128×128 improved accuracy to 92.28%, and augmentation further boosted performance (F1-score: 0.97, Accuracy: 96.78%). Advanced models like U-Net Inspired, Inception, and Skip Connection CNNs all achieved F1-scores of 0.98 and accuracies above 96%. The Inception + Skip CNN also performed well (Precision: 1.00, Recall: 0.96). These results highlight the importance of resolution, depth, and augmentation for robust Erythroblast detection.

Immature Granulocyte. Performance varied considerably across models for Immature Granulocytes. The Baseline CNN (64×64) underperformed (Accuracy: 81.00%, F1-score: 0.77), while increasing resolution to 128×128 improved results (F1-score: 0.87). Augmentation further raised the F1-score to 0.90. The U-Net Inspired CNN achieved 91.02% accuracy and a 0.91 F1-score, highlighting the benefit of encoder-style designs. The *Skip Connection CNN* reached the best F1-score (0.92) and strong accuracy (90.33%). Inception-based models yielded decent but lower scores, and the Inception + Skip CNN performed unexpectedly poorly. Overall, models with skip connections or encoder mechanisms proved most effective for this class.

Lymphocyte. Lymphocyte classification improved consistently with more advanced architectures. The Baseline CNN (64×64) achieved an F1-score of 0.90, which increased to 0.93 with 128×128 input and reached 0.99 after augmentation. The U-Net Inspired CNN scored 0.97, while both Inception variants reached 0.99. The *Skip Connection CNN* delivered the best performance with 99.59% accuracy, perfect recall, and a 0.99 F1-score. These results highlight the effectiveness of deeper, regularised models, especially those with skip connections, for lymphocyte classification.

Monocyte. Monocyte classification showed wide performance differences across models. The Baseline CNN (64×64) performed poorly (F1-score: 0.75, accuracy: 71.13%), but resolution increase and augmentation improved results to 0.92 F1-score and 96.13% accuracy. The U-Net Inspired CNN scored 0.93, while the *Skip Connection CNN* outperformed all with 96.83% accuracy and a 0.98 F1-score. Although the Inception CNN had perfect precision, its low recall (0.79) reduced its F1-score to 0.88. The Inception + Skip CNN also underperformed, confirming that skip-based models are more effective for Monocyte detection.

Neutrophil. Neutrophil classification was consistently strong, with all models achieving F1-scores above 0.92. The Baseline CNN (64×64) surprisingly reached 97.60% accuracy, though with lower precision (0.87). Higher-resolution models, including the augmented baseline, U-Net, and Inception variants, achieved balanced precision and recall (F1-score: 0.95-0.96). The Inception CNN achieved the highest accuracy (98.20%), while the *Skip Connection* and Inception + Skip models also generalised well. Overall, Neutrophils were reliably classified across architectures.

Platelet. Platelet classification was consistently near-perfect across all models. Even the Baseline CNN (64×64) achieved 99.36% accuracy and a 0.99 F1-score. All other models, including the U-Net, Inception, and Skip Connection variants, reached 100.00% accuracy with F1-scores of 0.99-1.00. The *Skip Connection CNN* achieved perfect precision and recall, confirming that Platelets are easily distinguishable and well-handled by both simple and advanced architectures.

Overall, the class-wise analysis reveals that model performance varied substantially depending on the blood cell type and architectural complexity. Classes like Platelet and Eosinophil were consistently classified with near-perfect accuracy, even by simpler models, indicating their distinctive visual features. In contrast, more challenging classes such as Monocyte, Immature Granulocyte, and Erythroblast benefited greatly from deeper architectures and data augmentation, with skip connections and encoder-style designs proving particularly effective. The *Skip Connection CNN* and *U-Net Inspired CNN* consistently emerged as top performers across most classes, balancing accuracy, recall, and F1-scores. These findings emphasise the importance of model architecture, resolution, and augmentation strategies in achieving robust, generalisable performance across diverse blood cell types.

VII. Concluding Remarks

This study evaluated a series of CNN architectures for peripheral blood cell classification using the BloodMNIST dataset. Among the tested models, the Skip Connection CNN emerged as the most effective, combining high test accuracy, generalisation ability, and resource efficiency. The U-Net Inspired CNN also demonstrated strong performance with lower memory usage. The Baseline CNN benefited notably from data augmentation, while models trained on low-resolution inputs were consistently outperformed by their high-resolution counterparts.

The proposed models, particularly the Skip Connection and U-Net variants, show potential for applications such as clinical diagnostics and edge-based deployment, where both accuracy and computational efficiency are important. Their strong class-wise performance further highlights their suitability for automated blood analysis pipelines.

Future work could address several limitations identified in this study. First, systematic hyperparameter tuning may yield further performance improvements. Incorporating attention mechanisms into the architectural design is another promising direction that could enhance feature learning and interpretability. Employing higher-resolution input images (e.g., 256×256) may improve fine-grained classification but would require greater computational resources. Moreover, extending the evaluation to include multiple biomedical imaging datasets would help assess the robustness and generalisability of the proposed models.

From a *personal perspective*, this project offered a valuable learning experience in the design, implementation and evaluation of convolutional neural networks for biomedical image classification. I explored a range of architectural paradigms from simple baseline CNNs to deeper models incorporating skip connections and inception modules and observed how these choices affected class-wise performance and overall generalisation. A major challenge was managing GPU memory constraints, particularly when training high-resolution or computationally intensive models. To address this, I switched to the Google Colab Pro environment, which provided improved memory capacity and stability. I also encountered difficulties in identifying effective augmentation strategies that improved performance without hindering convergence. Despite these challenges, the project significantly deepened my understanding of CNN design and evaluation in a biomedical imaging context.

References

[1] J. Yang, Z. Shi, L. Liu, Q. Chen, Y. Liu, and Q. Yao, "Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8141–8150, 2021.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sanchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[3] S. K. Zhou, H. Greenspan, and D. Shen, "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," *Physics in Medicine & Biology*, vol. 66, no. 12, p. 12TR01, 2021.

[4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, Springer, 2015.

[5] T. Falk, D. Mai, R. Bensch, O. Cicek, A. Abdulkadir, Y. Marrakchi, A. Bohm, J. Deubner, Z. Jackel, K. Seiwald, *et al.*, "U-net: deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, 2019.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *European Conference on Computer Vision (ECCV)*, pp. 630–645, 2016.

[10] S. H. Shuvo, S. Mahmud, A. H. Shamim, *et al.*, "Inception-resnet based deep learning model for disease classification from chest x-ray images," *Informatics in Medicine Unlocked*, vol. 28, p. 100815, 2022.