Blood Cell Type Prediction - Feasible Task for Deep Neural Networks?

Aleksandra Maslova†

***Abstract*—Automated blood cell classification is crucial as it delivers rapid, precise, and consistent diagnostic results that are vital for identifying and treating various medical conditions. Additionally, it eliminates the tedious and cognitively demanding process of manually examining countless microscopic images, enabling healthcare professionals to dedicate their time and expertise to more essential patient care activities. This research evaluates six different architectures: five CNN variants (including attention mechanisms CBAM and SE, skip connections, and Inception-inspired designs) and a Vision Transformer. All models were trained from scratch on the BloodMNIST dataset for 8-class blood cell classification. Models were tested on both original imbalanced data and an augmented balanced dataset to assess the impact of data augmentation. Results demonstrate that Inception achieved the highest accuracy of 98.25% on the augmented dataset, closely followed by CNN with CBAM at 97.84%. Data augmentation consistently improved all models' performance by 0.46-4.48%, confirming the importance of balanced training data. Computational resource analysis revealed the high cost of the best-performing Inception model, while CNN with CBAM achieved comparable accuracy with significantly lower resource requirements, providing valuable insights for model selection in deployment scenarios.**

***Index Terms*—Blood Cell Classification, Medical Image Analysis, Convolution Neural Networks, Visual Transformer, Attention Mechanism, Skip Connections**

I. Introduction

The development of deep learning has revolutionized medical image analysis by enabling automated feature extraction and pattern recognition that can distinguish subtle morphological differences between cell types with remarkable precision. However, the optimal architectural design for blood cell classification remains an active area of research, with various approaches showing different strengths in handling the inherent challenges of microscopic cellular imagery.

Current state-of-the-art approaches in medical image classification predominantly rely on transfer learning from pre-trained models or fine-tuning of existing architectures [1], [2], with recent advances exploring ensemble methods and hybrid architectures with attention mechanisms across various medical imaging modalities [3], [4].

In this work, I systematically evaluate six different deep learning architectures for blood cell classification on the BloodMNIST [5] dataset, progressing from simple Convolutional Neural Networks (CNNs) to more sophisticated designs including skip-connection networks, CNNs with attention mechanisms, Inception-inspired architecture, and Vision Transformers. All models were trained from scratch without transfer learning to ensure a fair comparison of architectural capabilities. This comprehensive comparison aims to identify the most effective approaches for automated blood cell recognition and provide insights into architectural choices that best capture the discriminative features necessary for accurate hematological analysis, considering also computational efficiency including training time, memory consumption, and CPU usage. All experiments were conducted on an Apple M1 Pro processor.

This report is structured as follows. In Section II I describe the current state of research in the field, the system and the dataset are respectively presented in Sections III and IV. The proposed model architectures are detailed in Section V and their performance evaluation is carried out in Section VI. Concluding remarks are provided in Section VII.

†Department of Mathematics, University of Padova,
e-mail: aleksandra.maslova@studenti.unipd.it

II. Related Work

Numerous researchers have dedicated their efforts to evaluating various CNN architectures for medical image analysis, consistently demonstrating their remarkable performance and capabilities in clinical diagnostic tasks [6]–[8].

Neural Networks with skip-connections were initially proposed by He K. and al. [9] and helped to overcome the vanishing gradient problem through identity skip connections that preserve low-level features and enhance gradient propagation, leading to accelerated training convergence and better generalization performance. This approach was applied to medical images showing high performance results [10].

Inception network, originally proposed by Szegedy et al. [11], introduced a new type of blocks consisting of multiple parallel branches that exploit different convolution strategies allowing the model to capture features at multiple scales and depths. This approach was also found to be beneficial for medical images [1], [12]. It is worth stating that this architecture, due to its high complexity, is usually used as a pre-trained model [1].

Visual Transformers (ViT), initially proposed by Dosovitskiy et al. [13] revolutionized the image processing architectures. Unlike convolutional neural networks (CNNs), Vision Transformers (ViTs) leverage self-attention mechanisms to capture both local and global features from image data, subsequently passing these features through residual connections into a fully connected multilayer perceptron (MLP) head. ViTs were proven to be effective in application to the medical image analysis [14], [15].

III. Processing Pipeline

The computational framework developed for this study enables systematic evaluation of multiple deep learning architectures on identical datasets for blood cell image classification, ensuring reproducible and unbiased results that facilitate fair model comparison. Figure 1 illustrates the overall methodology and workflow structure.

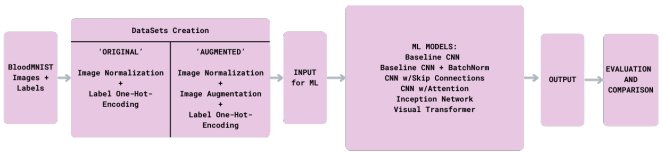

Fig. 1: Overview of the study pipeline

For evaluating the proposed architectures, two datasets were used: the "Original" and the "Augmented" (dataset creation is described in Section IV in detail). This approach allowed to evaluate the importance of a balanced training set for model performance.

The models' architecture progressively evolved from simple CNN variants to more sophisticated designs, including skip-connections, two different attention mechanisms, Inception-inspired blocks, and Vision Transformers.

IV. Images and Features

This study utilizes the BloodMNIST dataset, which comprises 17,092 microscopic RGB (three-channel) images of 8 types of peripheral blood cells and the corresponding labels, assigned to each image to differentiate the distinct classes. Figure 2 shows sample images from each class of the BloodMNIST dataset, illustrating the eight different types of peripheral blood cells.

For this study, out of 3 available image sizes, the smallest size of 64x64 pixels was chosen due to computational resource limitations.

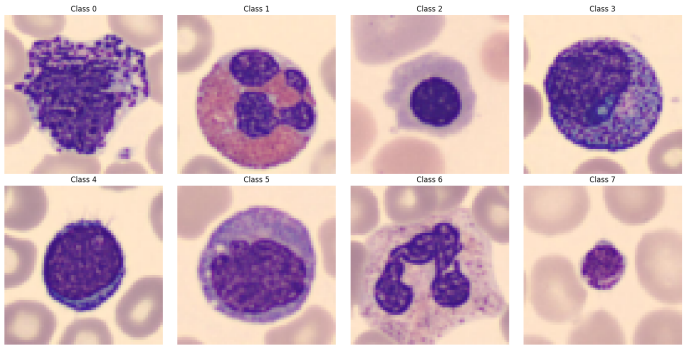

Fig. 2: Representative sample images for each of the eight blood cell classes in the BloodMNIST dataset

A. Class Distribution

Once downloaded, the dataset was first inspected for possible imbalance in the class representation. The class distribution for each subset (train, validation, and test) is presented in Table 1. It can be clearly seen that classes 0, 2, 4, and 5 are highly underrepresented. To address this issue, a data augmentation technique was applied, which is described in more detail in the following section. As a result of the augmentation procedure, all classes had 2,330 samples, which corresponds to the number of samples in the initially most represented class (class 6).

TABLE 1: Class distribution across training, validation, and test sets

Class	Train		Validation		Test	
	Count	%	Count	%	Count	%
0	852	7.1	122	7.1	244	7.1
1	2181	18.2	312	18.2	624	18.2
2	1085	9.1	155	9.1	311	9.1
3	2026	16.9	290	16.9	579	16.9
4	849	7.1	122	7.1	243	7.1
5	993	8.3	143	8.4	284	8.3
6	2330	19.5	333	19.5	666	19.5
7	1643	13.7	235	13.7	470	13.7

B. Train, Validation and Test Splitting

The BloodMNIST dataset is provided with predefined training, validation, and testing subsets. These sets consisted of 11,959 images with corresponding labels for training (70%), 1,712 samples for validation (10%), and 3,421 samples for testing the models (20%). The dataset splits are stratified to maintain consistent class proportions across training, validation, and test subsets.

C. Datasets Creation

Two datasets were prepared for this study, which differed in their training split composition. Both datasets maintained identical validation and test splits with the same sizes as originally provided by BloodMNIST, while the training splits varied due to augmentation technique, applied exclusively to one dataset.

- The first dataset, named "Original", maintained the original class imbalance in the training split. Images in all splits were normalized by dividing pixel values by 255; labels in all splits were one-hot-encoded. "Original" dataset's training split consisted of 11,959 images with corresponding labels, as provided by BloodMNIST.
- The second dataset, named "Augmented," incorporated data augmentation technique applied exclusively to the training images subset using vertical and horizontal flipping to account for the various orientations of blood cells during microscopic image capture. For the augmented dataset, validation and test sets received only normalization to preserve the real-world distribution, while labels were one-hot-encoded. "Augmented" dataset's training split consisted of 18,640 images with corresponding labels.

V. Learning Framework

In this study, for the classification task on the BloodMNIST dataset all the following models were trained from scratch, no pre-trained model was used. In this section, all the evaluated models are described, as well as the main parameters of the training and evaluation approach.

A. Training Configuration

All models were trained on RGB images of size $64 \times 64 \times 3$ over 25 epochs with a batch size of 64. Unless otherwise specified, the majority of models employed the Adam optimizer with learning rate of 0.0001, accompanied by *EarlyStopping* with a patience of 5 epochs. All models were compiled using the categorical cross-entropy loss function and evaluated using accuracy as the primary metric, suitable for multi-class classification tasks.

The *ReduceLROnPlateau* callback was applied consistently across all models, reducing the learning rate by a factor of 0.5 if the validation loss plateaued for 3 consecutive epochs, with a minimum learning rate capped at 1×10^{-6}. This ensured adaptive control over learning rates and contributed to more stable convergence.

For select models, variations such as alternative learning rate or extended *EarlyStopping* patience (e.g., 10 epochs) were explored. These exceptions are described in the corresponding model-specific subsections.

B. Model Architectures

1) Baseline CNN: This is the simplest model tested, it consists of two convolution layers (16 and 32 filters, respectively), each followed by ReLU activation and MaxPooling layer. The output of two blocks is Flattened and passed through two dense layers, the last one with 8 neurons and softmax activation produces a probability distribution over the 8 classes. The architecture of the baseline model is shown in Figure 3.

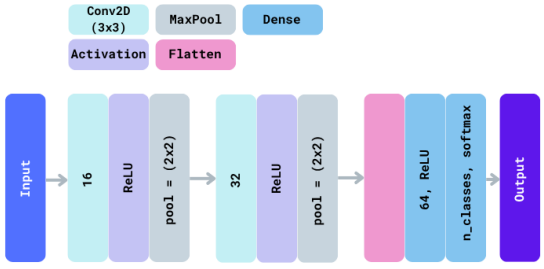

Fig. 3: Baseline CNN Model

2) Baseline CNN with Batch Normalization: In this model, the influence of the batch normalization layer was investigated. To each convolution block of the baseline architecture, the batch normalizaiton layer was added before the activation layer. The updated architecture is shown in Figure 4.

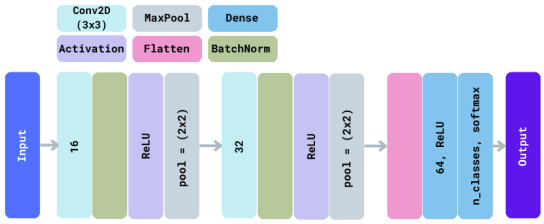

Fig. 4: Baseline CNN Model with Batch Normalization Layers

3) CNN with Skip Connections: This model was inspired by the ResNet architecture; the lighter version was implemented to be tested on the blood cell dataset. The proposed architecture can be investigated in Figure 5.

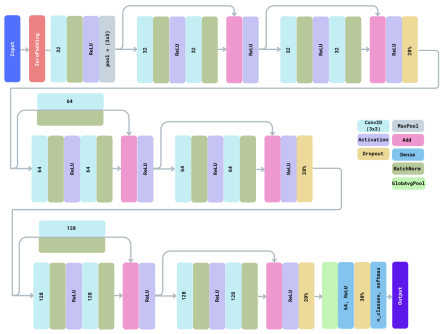

Fig. 5: CNN Model with Skip Connections

After zero padding and the initial convolution, the input is passed through three *Skip Blocks*, each composed of residual units with skip connection. The core layers in each residual block consist of two consecutive convolutional layers, each followed by batch normalization, and a ReLU activation.

In each block, the original input (skip path) is either forwarded unchanged or passed through a 1×1 convolution with optional downsampling (stride of 2) to match the spatial dimensions and depth of the main path. The output of the main path is then added to the skip path, and a final ReLU activation is applied to the combined output.

Dropout layers are included after every pair of residual blocks to prevent overfitting. Finally, a global average pooling layer and a dense ReLU layer are applied before the softmax output layer, enabling multi-class classification.

4) CNN with Attention Mechanism: CNNs are highly effective at capturing local spatial patterns through hierarchical feature extraction. However, they inherently suffer from limited global context awareness due to their local receptive fields and fixed kernel operations.

To address this limitation, attention mechanisms can be integrated within CNNs. Attention enables the model to adaptively focus on the most relevant regions of an image, regardless of spatial distance, thereby improving the capture of long-range dependencies and semantic relationships.

In this part of the study the two commonly used Attention Mechanisms for images were tested: Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM).

For CNN with CBAM and SE blocks, the Adam optimizer was configured with a learning rate of 0.0005. In addition to this, Early Stopping callback was set to 10 epochs patience.

- SE is a channel attention mechanism that enhances feature representations by modeling inter-channel dependencies. It compresses spatial information via global average pooling, then uses two fully connected layers with a sigmoid activation to generate channel-wise weights, which are applied to the feature maps. SE blocks are lightweight and can improve CNN performance with minimal overhead.
- CBAM combines channel and spatial attention to refine features. It first computes channel attention using average and max pooling, then applies spatial attention based on pooled channel descriptors. This sequential attention helps CNNs focus on both informative features and relevant spatial locations, improving overall effectiveness.

Each Attention Block was incorporated into the same architecture to evaluate its impact on the model's performance. The scheme of the models with Attention Blocks is shown in Figure 6.

Fig. 6: CNN Model with Attention Blocks

5) Inception CNN: This model was inspired by the original Inception (GoogleNet) architecture. This model incorporates custom *Inception-like blocks* to enhance representational power by enabling the network to capture features at multiple scales and depths.

Each *Inception Block* consists of four parallel branches with different convolutional or pooling operations, all contributing equally to the final output, which represents the concatenation of the outputs of each branch. The structure is summarized in Figure 7.

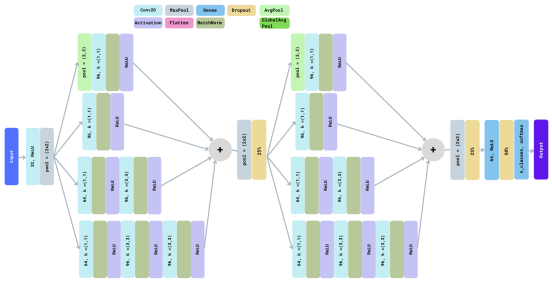

Fig. 7: Inception CNN Model

Two Inception Blocks are placed consecutively, each of which is followed with maxpooling and dropout layers. Lastly, the data is processed through two dense layers: first consists of 64 neurons and is followed by dropout layer (50% dropout rate), the second represents the prediction layer with 8 neurons and softmax activation.

6) Visual Transformer: The custom *patch_and_position_embedding* function extracts non-overlapping image patches using a *Conv2D* layer and adds learnable positional embeddings to each patch representation. This mechanism is inspired by Vision Transformers (ViTs) and is used to encode both spatial and positional information before feeding data into subsequent transformer-based or attention-based layers.

The workflow of creation positional patch embeddings is the following:

- Patch Extraction: The input image is divided into fixed-size patches via a convolution operation with kernel size and stride equal to the patch size.
- Flattening: The resulting patch grid is reshaped into a sequence of patch embeddings.
- Positional Embedding: A learnable embedding is added to each patch to retain spatial order information.

This approach enables the model to treat images as sequences of patch tokens, suitable for transformer-like architectures.

The transformer block uses *MultiHeadAttention* layer as a self-attention mechanism applied to the normalized sequence of patches. The output of attention layer is added to the embedding and processed through a multilayer perceptron (MLP), which basically consists of dense layer with GeLU activation and dropout. In the current work, a two-layer perceptron was implemented. The output of MLP is added to the tensor passed to the MLP and normalized. All the outputs of the Transformer block are reshaped with *Flatten* layer and used as the image representation input to the classifier head, which is represented with another two-layer MLP (512 and 256 units respectively) followed by final dense layer with 8 neurons and softmax funciton, which produces the final predictions.

In this work, the transformed block was used only once for computational reasons, while the number of such blocks can be increased.

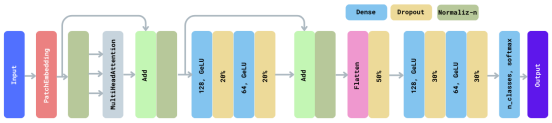

Fig. 8: Visual Transformer

C. Evaluation Metrics

To comprehensively assess model performance, both classification quality and computational efficiency were evaluated using the following metrics:

- **Test Accuracy (%)**: The percentage of correctly classified instances in the test dataset. It is calculated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100;$$

- **Weighted F1 Score**: The harmonic mean of precision and recall, weighted by the number of instances in each class. It is given by:

$$\text{Weighted F1} = \sum_{i=1}^{K} w_i \cdot \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i},$$

where $w_i = \frac{n_i}{\sum_{j=1}^{K} n_j}$ is the weight for class i, n_i is the number of true instances in class i, and K is the number of classes.

- **Weighted Recall**: The average recall across all classes, weighted by support:

$$\text{Weighted Recall} = \sum_{i=1}^{K} w_i \cdot \text{Recall}_i;$$

- **Weighted Precision**: The average precision across all classes, weighted by support:

$$\text{Weighted Precision} = \sum_{i=1}^{K} w_i \cdot \text{Precision}_i;$$

- **Training Time (sec)**: The total duration required to train the model, measured in seconds.
- **CPU Usage (%)**: The average percentage of CPU resources utilized during the training process.
- **Memory Usage (MB)**: The memory consumption during model training, measured in megabytes (MB), required for total parameters of the model.

VI. Results

A. Performance Results

Since each model was evaluated using two datasets, the results are organized in two separate tables. The performance of the models that were tested on the "Original" dataset is summarized in Table 2. Analogous results for the "Augmented" dataset are presented in Table 3.

TABLE 2: Performance metrics on the Original dataset

Model	Accuracy(%)	Precision	Recall	F1-score
Baseline CNN	90.35	0.90	0.90	0.90
Baseline + BatchNorm	94.88	0.95	0.95	0.95
CNN w/Skip Conn-s	94.59	0.95	0.95	0.95
CNN w/CBAM	**96.87**	**0.97**	**0.97**	**0.97**
CNN w/SE	90.47	0.91	0.91	0.92
Inception	93.77	0.94	0.94	0.94
ViT	94.86	0.95	0.95	0.95

Key findings from results of models trained on "Original" dataset (analyzing independently from the "Augmented" dataset):

- **CBAM is the best-performing model among tested**: CNN with CBAM achieves the best performance across all metrics (96.87% accuracy), demonstrating the effectiveness of combined channel and spatial attention mechanisms for this task.
- **Attention Mechanism Disparity**: There's a significant performance gap between CBAM (96.87%) and SE (90.47%), which is surprising since both are attention mechanisms. This suggests that spatial attention (which CBAM includes but SE lacks) may be particularly important for blood cell classification.
- **Batch Normalization Impact**: The dramatic improvement from Baseline CNN (90.35%) to Baseline with Batch Normalization (94.88%) shows that proper normalization is crucial for this dataset.

TABLE 3: Performance metrics on the Augmented dataset

Model	Accuracy (%)	Precision	Recall	F1-score
Baseline CNN	92.81	0.93	0.93	0.93
Baseline + BatchNorm	95.53	0.96	0.96	0.96
CNN w/Skip Conn-s	95.76	0.96	0.96	0.96
CNN w/CBAM	97.84	0.98	0.98	0.98
CNN w/SE	90.97	0.91	0.91	0.93
Inception	**98.25**	**0.98**	**0.98**	**0.98**
ViT	95.32	0.95	0.95	0.95

Key findings from results of models trained on "Augmented" dataset (analyzing independently from the "Original" dataset):

- **Inception is the best-performing model among tested**: Inception achieves the highest performance (98.25% accuracy) with excellent precision, recall, and F1-score (all 0.98), demonstrating that its multi-scale feature extraction is highly effective for blood cell classification.
- **Attention Mechanism Disparity**: Trained with augmented datatset, models with different attention mechanisms show persistent disparity between CBAM (97.84%) and SE (90.47%).

Analyzing performance of the models on two different training sets jointly, the following conclusions emerge:

- **Inception is the overall best-performing model**: Inception (98.25%) trained on augmented dataset achieves the best performance, surpassing CBAM (97.84%). This suggests that Inception's multi-scale feature extraction capabilities become more effective when trained on balanced augmented data.
- **Data augmentation universally improves performance**: All models show improved performance on the augmented dataset, ranging from 0.46% to 4.48% improvement, confirming that addressing class imbalance through augmentation significantly benefits model training across all architectures.
- **SE attention shows limitations**: CNN with SE attention (90.97%) shows the least improvement and remains the worst performer, reinforcing that spatial attention (missing in SE) is crucial for this task.
- **CBAM demonstrates consistent effectiveness**: CBAM continues to perform excellently (97.84%), maintaining its position as one of the top-performing models and showing consistent effectiveness across both datasets.
- **Consistent Metrics**: The alignment between accuracy, precision, recall, and F1-score across all models suggests balanced performance without significant bias toward specific classes.

B. Resource Utilization

Beyond classification performance, understanding the computational requirements of different architectures is crucial for practical deployment considerations. This section presents the resource utilization metrics for all evaluated models, including training time, CPU usage, and memory consumption.

Table 4 summarizes these computational performance indicators across both the "Original" and "Augmented" datasets, highlighting the computational trade-offs associated with increased model complexity.

TABLE 4: Runtime and resource usage per model

Model	Dataset	Time (sec)	CPU Usage (%)	Memory (MB)
Baseline CNN	Original	131	12.1	2.02
	Augmented	194	10.9	2.02
Baseline +BatchNorm	Original	126	7.6	2.02
	Augmented	250	10.5	2.02
CNN w/Skip Conn-s	Original	1234	10.4	2.7
	Augmented	1955	17.0	2.7
CNN w/CBAM	Original	810	11.0	0.47
	Augmented	1269	28.1	0.47
CNN w/SE	Original	409	17.9	0.48
	Augmented	861	26.9	0.48
Inception	Original	2532	26.8	2.11
	Augmented	3773	9.4	2.11
ViT	Original	923	9.8	32.65
	Augmented	1407	19.5	32.65

Looking at the resource utilization results, some observations can be stated:

- **Consistent Increases**: All models show longer training times on augmented datasets (48-58% increases), which is expected due to the larger dataset size (18,640 samples for "Augmented" dataset vs 11,959 samples for "Original" dataset).
- **CBAM's Cost**: Despite being the second-best performer (97.84%), CBAM uses minimal memory and moderate training time, offering excellent efficiency.
- **Inception's Cost**: The best performer (98.25%) requires the highest computational cost, representing a classic performance-efficiency trade-off.
- **ViT's Cost**: ViT shows moderate resource usage despite underperforming, suggesting potential for optimization.

C. Overall Results Considerations

Based on the comprehensive evaluation across both Original and Augmented datasets, Inception emerged as the best-performing model with 98.25% accuracy on the augmented dataset, closely followed by CNN with CBAM at 97.84%.

Data augmentation consistently improved all models' performance by 0.46-4.48%, confirming that addressing class imbalance is crucial for effective blood cell classification.

From a computational perspective, attention-based models (CBAM, SE) demonstrated superior memory efficiency (0.47-0.48 MB) compared to other architectures, though Inception required the highest training time (2532-3773 seconds) despite achieving top performance. CBAM offers the best balance between performance and computational efficiency, while Inception provides the highest accuracy at increased computational cost.

Overall, the choice of model should be made considering specific deployment conditions and computational constraints, with CBAM offering the best balance between performance and efficiency, while Inception provides maximum accuracy at higher computational cost. Nevertheless, there remains substantial room for improvement in the proposed architectures through further optimization and refinement.

VII. Concluding Remarks

In the current report several deep learning architectures were developed and evaluated, proving that automated blood cell type prediction is a feasible task for modern machine learning approaches.

Several directions for *future research* that could enhance the performance and robustness of the proposed models:

- **Hyperparameter Optimization:** Further hyperparameter tuning may be beneficial for improving model performance, particularly for architectures incorporating attention mechanisms.
- **Higher Resolution Dataset Evaluation:** The BloodMNIST dataset provides images at higher resolution compared to the current dataset. Training sophisticated models such as Inception-inspired architecture and Vision Transformer on these higher-resolution images could be particularly advantageous, as these models are designed to capture fine-grained features that may be more apparent at increased spatial resolutions.
- **Vision Transformer Enhancement:** The Vision Transformer architecture shows promising potential despite its current underperformance compared to CNN-based models. Adding additional transformer blocks or implementing more sophisticated attention mechanisms could unlock the full potential of this architecture for medical image classification, particularly when combined with higher resolution images.
- **Advanced Validation Techniques:** Implementing additional validation methodologies beyond the current Hold Out approach would provide a more comprehensive and robust assessment of model performance. Technique such as k-fold cross-validation could offer deeper insights into model generalization capabilities and reduce the risk of overfitting to specific dataset characteristics.

From a *personal perspective*, working with this task allowed me to deepen my knowledge of CNN architectures while providing me with hands-on experience in designing, tuning, and evaluating models for image classification. One of the initial challenges was the search for the right augmentation technique, since harsh augmentation led to poor performance due to excessive deformation and smudging of the resulting images. For example, aggressive rotation as part of the augmentation pipeline caused significant image distortion that negatively impacted model training. Another significant difficulty was the limitation of computational resources, which

constrained the depth and complexity of the models that could be practically implemented and trained. Overall, it was a valuable experience full of new knowledge that I intend to further broaden and deepen in future research endeavors.

REFERENCES

[1] H. Kim, A. Cosa-Linan, N. Santhanam, and et al., "Transfer learning for medical image classification: a literature review.," *BMC Med Imaging*, vol. 22, no. 69, 2022.

[2] A. W. Salehi, S. Khan, G. Gupta, and et al., "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustainability*, vol. 15, no. 7, 2023.

[3] W. Xin, Y. Feng, X. Hong, and et al., "CTransCNN: Combining transformer and CNN in multilabel medical image classification.," *Knowledge-Based Systems*, vol. 281, 2023.

[4] K. Cao, T. Deng, C. Zhang, L. Lu, and L. Li, "A CNN-transformer fusion network for COVID-19 CXR image classification.," *PLoS ONE* , vol. 17, no. 10, 2022.

[5] J. Yang, R. Shi, and D. Wei, "MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification.," *Sci Data*, vol. 10, no. 41, 2023.

[6] F. A. Mohammed, K. K. Tune, B. G. Assefa, M. Jett, and S. Muhie, "Medical Image Classifications Using Convolutional Neural Networks: A Survey of Current Methods and Statistical Modeling of the Literature," *Mach. Learn. Knowl. Extr.*, vol. 6, no. 1, pp. 699–735, 2024.

[7] A. Girdhar, H. Kapur, and V. Kumar, "Classification of White blood cell using Convolution Neural Network," *Biomedical Signal Processing and Control*, vol. 71, 2022.

[8] C. Cheuque, M. Querales, R. Leon, and et al., "An Efficient Multi-Level Convolutional Neural Network Approach for White Blood Cells Classification.," *Diagnostics*, vol. 12, no. 2, 2022.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[10] L. H. Shehab, O. M. Fahmy, S. M. Gasser, and et al., "An efficient brain tumor image segmentation based on deep residual networks (ResNets)," *Journal of King Saud University - Engineering Sciences*, vol. 33, no. 6, pp. 404–412, 2021.

[11] C. Szegedy, L. Wei, J. Yangqing, and et al., "Going deeper with convolutions.," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[12] R. Tamilarasi and S. Gopinathan, "Inception Architecture for Brain Image Classification," *Journal of Physics: Conference Series*, vol. 1946, no. 7, p. 072022, 2021.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, and et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.," *arXiv:2010.11929 [cs.CV]*, 2020.

[14] M. Abou Ali, F. Dornaika, and I. Arganda-Carreras, "White Blood Cell Classification: Convolutional Neural Network (CNN) and Vision Transformer (ViT) under Medical Microscope.," *Algorithms*, vol. 16, no. 11, 2023.

[15] R. R. Sharma, A. Sungheetha, M. Tiwari, I. A. Pindoo, V. Ellappan, and G. G. S. Pradeep, "Comparative analysis of vision transformer and cnn architectures in medical image classification," in *Proceedings of the International Conference on Sustainability Innovation in Computing and Engineering (ICSICE 2024)*, pp. 1343–1355, Atlantis Press, 2025.