Medical Blood Cell Image Classification using U-Net Encoder Based Deep Learning Model

Abstract

The main components of human blood are plasma, red blood cells (RBC), white blood cells (WBC), and platelets. Blood is responsible of transporting nutrients to various organs and are utilized to protect our body from certain infections. Therefore, the examination of blood can assist medical professionals in determining a patient's physiological state. Blood cells are sub classified into 8 groups as, Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, Platelet based on their cytoplasm, form, and nucleus. These blood cells have been studied by pathologists and haematologists in labs using microscopes before classifying them manually. The manual methods are slower and more prone to human mistake so there is the need of automating this process. U-net is a model architecture widely used for medical image segmentation tasks. In this paper a U-net inspired encoder classifier has been proposed to automatically classify eight types of blood cells with increased accuracy. This classifier retains hierarchical feature extraction capability of the U-Net encoder while replacing the decoder with a lightweight global pooling-based classification head. A simple CNN and a deeper CNN were used as two baseline models. The proposed U-net inspired encoder classifier has been tested on the given data set of 1000 blood cell images. The outcome of the experiment demonstrates that this U-net inspired encoder classifier designed for blood cell classification attains a validation accuracy of around 98.75% and a test accuracy of around 98% on the given test dataset.

1. Introduction

In human body blood acts as a specific type of circulating connective fluid which takes oxygen from our lungs and transport that oxygen to all the cells in our body. Blood is compromised of two parts as plasma which is the liquid portion and the cell fragments. The cell fragments are composed white blood cells, red and platelets.

Blood cell classification is a popular research area among the scientists and medical professionals. In laboratories medical professionals use two different ways to analyze blood. First method is complete blood count (CBC) test which calculates the total percentage of red blood cells, white blood cells and platelets, while the second method is peripheral blood smears (PBS) test. Using the results of these tests the overall health of a patient can be determined. These microscopic blood images can reveal the types of red blood cells, white blood cells and platelets accurately in a blood sample making it easier to diagnose diseases at early stages. A change in the count of blood cell categories signals an illness or a disease. For an example low count of white blood cells can be an indication of a blood cancer and lower count of healthy red blood cells can be an indication of illnesses like Anaemia. [1]

The time-consuming and inaccurate nature of traditional blood cell classification techniques emphasizes the importance of precise systems for rapid and precise blood cell analysis. Conventional machine learning methods such as support vector machines, decision trees, k-nearest neighbors, naive bayes and artificial neural networks have been used to classify blood cells in microscopic blood smear images. Traditional machine learning techniques often involves preprocessing blood smear images, segmentation to separate cells, features extraction, feature selection to remove unwanted data and classification.

The performance of conventional machine learning algorithms in classification is greatly impacted by feature extraction and selection, despite some encouraging outcomes. It is complex and time-consuming to select the best features and determine the best feature extraction algorithm.

To address this challenging subject, several convolutional neural networks (CNNs), deep learning (DL) algorithms have recently been put forth. We can now estimate the type of blood cells from microscopic photos thanks to recent advances in deep learning.

DL-based methods may extract and select features on their own, unlike traditional ML methods, and CNNs has outperformed conventional ML methods for blood cell classification, according to the earlier studies [2].

The aim of this study is to develop a model for blood cell classification using machine learning and deep learning methods. Motivated by the effectiveness of U-Net encoder in medical imaging this study proposes a U-Net Encoder

Classifier which uses the down sampling (encoder) path of U-Net for feature extraction while replacing the decoder with a lightweight global pooling and fully connected classifier head. This design captures both local morphological patterns as well as deeper semantic features of blood cells while remaining computationally efficient. The dataset provided for this task consists of microscopic images of eight blood cell types, Basophil, Eosinophil, Erythroblast, Immature Granulocyte, Lymphocyte, Monocyte, Neutrophil, and Platelet. The labelled training set is used for supervised learning while the unlabelled test set is used for final inference. The dataset is split into training and validation sets using stratified sampling to preserve class balance. And standard preprocessing techniques, such as resizing, normalization, and data augmentation, are added to improve the generalization.

Two additional baseline models are implemented for comparison. Baseline 1 is a Simple CNN which is a shallow two-block convolutional network and Baseline 2 is a Deep CNN with a deeper architecture with four convolutional blocks and dropout regularization. These baselines provided insight into how increasing model depth and representational power effects model performance.

The main contributions of this work are, a U-Net inspired encoder classifier tailored for blood cell classification with eight blood cell classes, a systematic comparison with two baseline CNN architectures to evaluate depth and design choices, a comprehensive ablation study analyzing the effect of key components (batch normalization, dropout, U-Net encoder depth) and high classification accuracy approximately 98.75% demonstrating that encoder style architectures derived from medical segmentation networks can be effectively repurposed for blood cell classification tasks.

2. Related Work

Blood-cell classification has been widely explored using both traditional machine-learning pipelines and modern deep-learning approaches. Earlier methods relied on multi-stage processing pipelines involving segmentation, handcrafted feature extraction, and classical classifiers. For example, Rabul and Salam [3] used Otsu's thresholding combined with Gray Level Co-occurrence Matrix (GLCM) features and K-Nearest Neighbours, achieving 94.25% accuracy. Although effective, such methods depend heavily on carefully engineered features, limiting their robustness across datasets.

With the emergence of deep learning, convolutional neural networks (CNNs) have become the dominant approach for blood-cell analysis. Elhassan et al. [4] proposed a two-stage DCAE-CNN hybrid model for classifying atypical white blood cells, reporting performance close to 97–98%. Ahmad et al. [5] used DenseNet201 and Darknet53 to extract deep features achieving almost 99.9% accuracy on a five-class WBC dataset. These approaches highlight the benefit of deep feature extraction but rely on heavy pretrained backbones and complex feature-selection procedures.

Several studies have analysed the impact of different CNN architectures. Meena Devi and Ambary [6] evaluated multiple CNN models including AlexNet, VGG16, GoogleNet, and ResNet50 and found that VGG16 fine-tuned via transfer learning achieved the highest accuracy ($\approx 97\%$). These works demonstrate the effectiveness of hierarchical CNN feature extractors but primarily rely on standard image-classification architectures rather than medical-imaging-specific designs.

More recently, U-Net-based models traditionally used for segmentation have been adapted for classification tasks. Their encoder pathways capture fine-grained morphological details and multi-scale representations useful for biomedical images. Studies such as those by Deepika et al. [7] and Ansari et al. [8] employ deep encoders for analysing leukemia-related cells and report accuracies above 97-99

Unlike prior methods that depend on heavy pretrained models or multi-stage feature engineering pipelines, this approach adopts a lightweight U-Net encoder only architecture specifically tailored to blood-cell morphology. By leveraging U-Net's multi-scale feature extraction while removing the decoder, the proposed model remains computationally efficient yet achieves higher accuracy (98.75%) on the given eight-class dataset. It positions this method as a balance between heavy pretrained CNNs and shallow, handcrafted feature-based approaches.

3. Method

The proposed model is a U-Net Encoder Classifier which is a hybrid architecture that adapts the encoder path of the U-Net segmentation model for 8 class blood cell classification. [9]

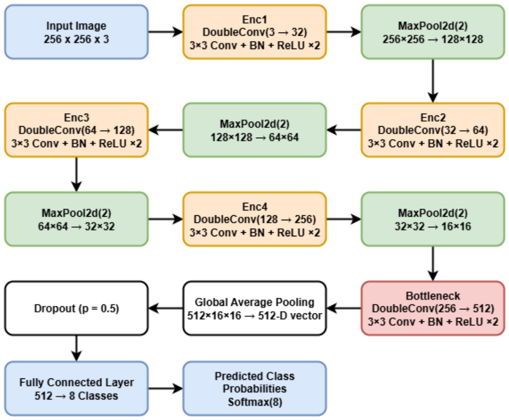

Figure 1. *Diagram of the model*

3.1. Model architecture and Algorithm

Double Convolution Block

The Double Convolution Block consists of two stacked 3×3 convolutions with Batch Normalization and ReLU activations starts each encoder stage.

Batch Normalization ensures that the activations have a stable mean and variance by normalizing feature maps within each mini batch. This improves gradient flow, lowers internal covariate shift, speeds up convergence and enables faster learning rates. BN normalizes each channel c of the feature map according to this calculation:

$$\hat{x}_c = \frac{x_c - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \qquad \mathrm{BN}(x_c) = \gamma_c \hat{x}_c + \beta_c, \qquad (1)$$

where μ_c and σ_c^2 denote the batch mean and variance for channel c, and γ_c, β_c are learnable scale and shift parameters.

When combined with ReLU, it prevents dead neurons and stabilizes nonlinear activations.

Both low-level textures and mid-level morphological structures, which are crucial for differentiating blood-cell types are captured in this block.

Encoder (Down sampling path)

The encoder consists of four stages. After each Double Convolution block with feature extraction, a 2×2 Max Pooling operation reduces the spatial resolution by half.

Stage	Input Size	Output Channels
Enc1	256×256	32
Enc2	128×128	64
Enc3	64×64	128
Enc4	32×32	256
Bottleneck	16×16	512

Table 1. *Spatial resolutions and feature dimensions*

Pooling progressively increases the receptive field. The receptive field refers to the spatial region of the input image that influences the activation of a neuron. In convolutional networks, stacking 3×3 convolutions and max-pooling layers increases the receptive field, allowing deeper layers to capture more global features. In the proposed U-Net encoder classifier, shallow layers capture local patterns such as edges, granules, and nucleus boundaries, while deeper stages capture global cell morphology.

Bottleneck layer

The bottleneck applies another Double Convolution at 512 channels. This block integrates global context and extracts deep hierarchical patterns representing, cell morphology and shape irregularities.The semantic layer in U-Net's segmentation encoder is analogous to this deeper model.

Global average pooling and Classification Head

After the encoder using Global Average Pooling (GAP) the feature tensor $f \in \mathbb{R}^{C\times H\times W}$ is reduced.

GAP Equation:

$$g_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_{c,i,j} \qquad (2)$$

where $f_{c,i,j}$ is the activation at channel c and spatial location (i, j), H and W denote the height and width of the feature map, and g_c is the averaged output for channel c after global average pooling.

This results in a channel vector of length 512 and this step eliminates the need for large fully connected layers, reduces parameters significantly, prevents overfitting and ensures the model can generalize to different input sizes. Global Average Pooling is standard in modern architectures such as *ResNet* and *EfficientNet* for its stability and simplicity.

Dropout Layer (Regularization)

Dropout with 0.5 regularizes the classifier. This reduces co-adaptation of neurons and helps prevent overfitting on small datasets like medical imagery.

Final Fully Connected Classification Layer

The final linear layer maps the 512-dimensional vector to 8 output neurons. The network is trained using Cross-Entropy Loss with label smoothing, and SoftMax is applied implicitly during the loss computation.

$$\hat{y} = Wg + b \qquad (3)$$

Where:

- W is the weight matrix for classification into 8 classes
- g is the Global Average Pooling output

3.2. Model Training

As the first thing before training all images were programmatically checked for corruption, unreadability, extreme low variance and invalid dimensions. A stratified 80/20 train validation split was utilized to maintain the original class distribution in both sets.Several augmentations were applied to training images to enhance generalization and minimize overfitting. These processes simulate natural variations in illumination, orientation, and cell structure, enabling the network to learn more robust features. Only deterministic transformations were applied to validation dataset to guarantee that evaluation is performed on unaltered, clean images. Images were normalized to stabilizes training, speeds convergence, and to align with standard CNN preprocessing recommendations.

During training a batch size of 32 was used with 2 worker threads for parallel data loading. Shuffling was activated only for the training set to ensure each epoch receives data in a different order, while the validation set remained deterministic.

The main model was trained for 60 epochs, while the baseline models were trained for 40 epochs since baselines do not require full optimization. Training was performed using the Adam optimizer with a learning rate of 1e-3 and weight decay of 1e-4 to limit overfitting. A dropout rate of 0.5 was applied in the classifier head of the main model to further improve regularization. For the loss function, Cross-Entropy with label smoothing ($\epsilon = 0.1$) was used to reduce overconfidence and improve generalization. Furthermore, a reduced learning rate (LR) on Plateau scheduler was employed to automatically to cut the LR in halve when the validation loss stabilized.

Overall, the model selection process relies on continuously tracking validation accuracy and restoring the best-performing checkpoint at the end, ensuring that all reported results reflect the highest-performing version of the model.

3.3. Justifications for Design choices

The model employs a U-Net encoder as it captures multi-scale features essential for distinguishing subtle blood-cell structures while remaining lighter than a complete U-Net. Each layer employs Double Convolution blocks, which extract richer features than a single convolution and improved gradient flow. Because segmentation is unnecessary the decoder part was removed, reducing computation and training time without harming classification ability. For the classifier, Global Average Pooling keeps the model compact and minimizes overfitting and Dropout applies regularization. Batch Normalization stabilizes training while ensuring faster convergence. With optimizations such as Adam, learning-rate scheduling, label smoothing, and best-model checkpointing, these choices provide a stable, efficient, and highly accurate blood-cell classifier.

4. Experiments

4.1. Experimental Setup

All experiments were performed in Google Colab with GPU acceleration (NVIDIA T4), using PyTorch 2.9.0 with support of NVIDIA CUDA version 12.6 and Python 3.12.12. Training and evaluation were conducted using a unified pipeline to ensure fair comparison across models.

All experiments were conducted with a fixed random seed for reproducibility. The training pipeline consisted of loading images in mini-batches using Data Loaders with shuffling enabled for training. Each model was trained using a standard loop that performs forward propagation, loss computation, backpropagation, and optimizer updates, followed by evaluation on the validation set.

4.2. Dataset division and parameter setting

The dataset consists of 3200 labelled blood cell images belonging to eight classes. To ensure balanced representation of all categories, the dataset was split using stratified sampling, allocating: 80% (2560 images) training, 20% (640 images) validation.

For the baseline models also the same training loop and hyperparameter structure were used as the main model to ensure controlled and fair comparison.

Table 2. *Comparison of model architectures parameter settings.*

Model Components	**Baseline1**	**Baseline2**	**Main**
Convolution Blocks	2	4	$5(\times 2)$
Max Channel Depth	64	256	512
Kernel Size	3×3	3×3	3×3
Pooling Stages	2	4	4
Batch Norm	Yes	Yes	Yes
FC Layers	1	2	1
Epochs	40	40	60

4.3. Evaluation Matrices

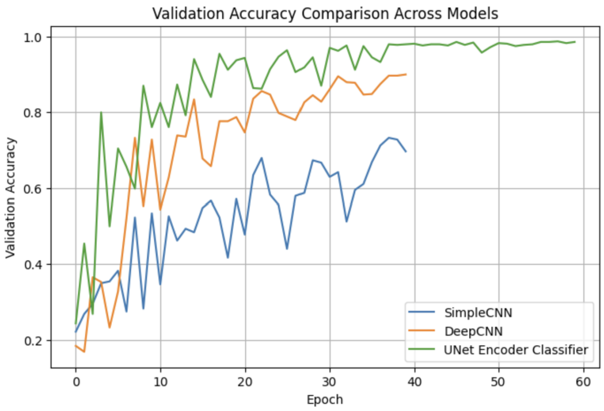

Figure 2. *Validation accuracy comparisons between baselines and main model.*

Metric	**Baseline 1**	**Baseline 2**	**Main Model**
Validation Acc	71.25%	90.00%	98.75%
Train Acc	56.40%	82.90%	97.40%

Table 3. *Comparison of training and validation accuracy across all models.*

The accuracy curves of three models across training epochs reveals that simple CNN learns slowly and shows unstable accuracy throughout training. The Deeper CNN converges faster and more stedily than simple CNN demonstarting that convolutional depth and feature capacity significantly improves performance. However the U-Net encoder classifier shows a rapid improvement in the first few epochs.

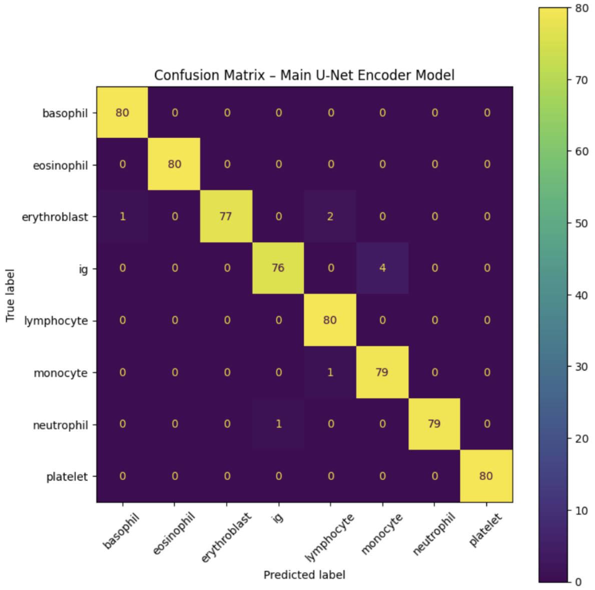

Figure 3. *Confusion Matrix Main Model*

Key observations of the confusion matrix provides insights into per-class performance. The key observations shows perfect classification for Basophil, Eosinophil, Lymphocyte, and Platelet. Each of these classes has 80 correct predictions with no misclassifications.

5. Ablation Studies

5.1. Batch Normalization Ablation

Objective: Quantify the impact of Batch Normalization (BN) on model performance and training stability.

Method: By training two variants of the model.

1. **T**he main model with Batch Normalisation layers after each convolution.
2. **A**n ablated version with all Batch Normalisation layers removed while keeping all other components identical.

Configuration	Best Val Acc	Δ from Main Model
With BN	98.75%	–
Without BN	12.50%	−86.25%

Table 4. *Batch Normalization ablation results.*

Analysis: Removing Batch Normalization resulted only 12.50% validation accuracy, demonstrating its critical role in model performance. By first 40 epochs the main model with BN has shown rapid increase in validation accuracy, however the ablated model has achieved a very low validation accuracy and has not shown any improvement.

This confirms that BN is essential for this to stabilize training activations, to improve gradient flow, convergence speed and final accuracy. So, removing batch normalization leads to major performance drop making it the most influential components in this U-Net encoder classifier architecture. [10]

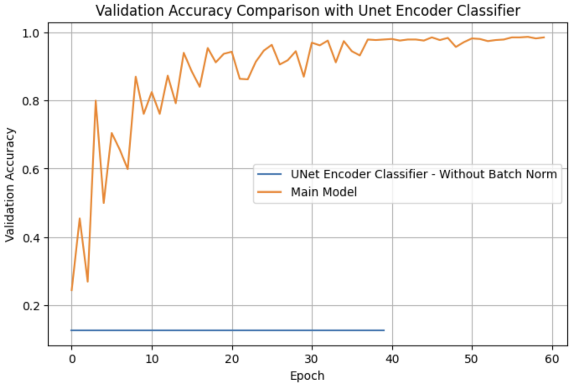

Figure 4. *Validation accuracy comparison with and without Batch Normalization.*

5.2. Shallow Encoder Ablation

Objective: Evaluate whether the deepest layers of the encoder including encoder level 4 (256 channels) and bottleneck layer (512 channels) provides sufficient contribution to classification performance.

Method: By training two variants of the model.

1. **T**he main model with 4 encoder levels and bottleneck layer.
2. **A**n ablated version with only 3 encoder stages, removing encoder level 4 and bottleneck layer.

Configuration	Best Val Acc	Δ from Main Model
Full Encoder	98.75%	–
Shallow Encoder	94.06%	−4.69%

Table 5. *Shallow encoder ablation results.*

Analysis: Removing Encoder layer 4 and the bottleneck reduces model depth, dropping the representational capacity from 512 to 128 channels. Both layers have contributed a 4.69% improvement in validation accuracy.

The shallow model displays a decrease in validation accuracy indicating that high level hierarchical features extracted in the deeper layers are crucial. The bottleneck provides marginal gains that may be valuable for complex datasets. Despite reducing model capacity, the ablated model maintained competitive performance, suggesting that the encoder layers capture most discriminative features. Overall, this ablation confirms that deep encoder stages play an essential role in capturing complex morphological patterns. [11]

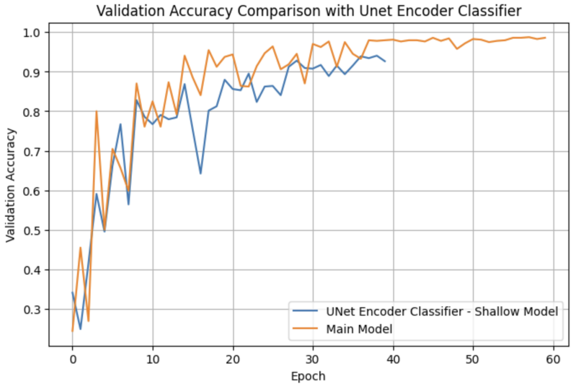

Figure 5. *Validation accuracy comparison between full and shallow encoder variants.*

5.3. Dropout Rate Ablation

Objective: To assess how dropout influences generalization and to identify the optimal level of regularization.
Method: Two variants were trained with different dropout configurations:

1. **Main model:** dropout rate of 0.5 in the classification head.
2. **UNet-NoDropout:** dropout removed entirely (0.0).

Dropout Rate	Best Val Acc	Δ from Main Model
0.5	98.75%	–
0.0	98.28%	−0.47%

Table 6. *Dropout ablation results.*

Analysis: The main model with dropout rate of 0.5 achieved optimal validation performance balancing regularization and model capacity. Without dropout the ablated model also reached a very similar best validation accuracy of 98.28% in 40 epochs. The figure shows how validation accuracy curves largely overlap, and the train validation gap remains small in both settings indicating no pronounced overfitting when dropout is removed. This result confirms that under the current training setting the U-Net Encoder classifier is already well regularized and drop out has only a marginal effect on final performance. In practice dropout mainly smooths learning curves but it is not a critical component for achieving high accuracy on this dataset. [12]

6. Discussion

The proposed U-Net Encoder classifier achieved strong performance in blood cell image classification task surpassing both the baseline CNN models and ablations variants. These results emphasize the importance of the key components in the model architecture. Batch Normaliation improved optimization stability substantially, dropout reduced overfitting which is common in smaller datasets and deeper encoder stages with bottleneck contributed meaningful gains in representational capacity.

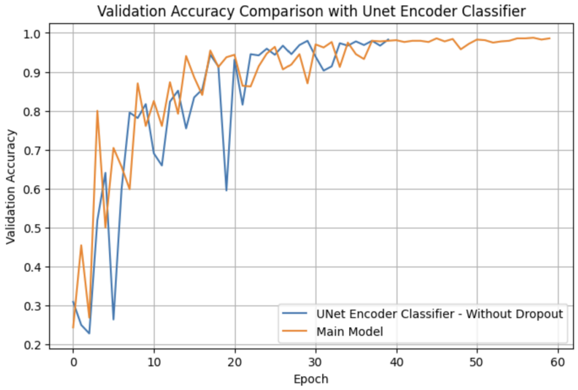

Figure 6. *Validation accuracy with and without dropout.*

Despite the high accuracy the study presents several challenges. The dataset is relatively small, increasing susceptibility to overfitting and limiting model's ability to generalize to unseen data. Moreover, some classes exhibit high morphological similarity leading to occasional misclassification. The model also assumes clean, centred cell images whereas real blood smears often contain overlapping cells, noise and multiple objects per field of view. Key limitations include the absence of external validation because the model is trained and evaluated only on one dataset. Therefore, we can't confirm if the model generalizes well to new unseen data from real clinical environments. Also, the model classifies the whole image directly, assuming the cell is centred and cleanly cropped. It does not identify or isolate the cell within the larger smear image. Consequently, the model cannot be applied directly to whole blood smear images without preprocessing.

Future work could be explore integrating a full U-Net or instance segmentation model to isolate individual cells experimenting with lightweight architectures for deployment. Also, can leverage self-supervised pretraining to overcome limited labelled data. Validating the model on datasets from multiple laboratories would further strengthen its robustness and clinical effectiveness.

In conclusion, this study demonstrates that encoder based deep architectures are highly effective for blood cell classification tasks. However additional work is required to ensure applicability of the model in real-world diagnostic settings.

References

[1] R. Asghar, S. Kumar, P. Hynds, and A. Mahfooz, "Classification of all blood cell images using ml and dl models," *arXiv preprint arXiv:2308.06300v3*, 2024. 1

[2] A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient classification of white blood cell leukemia with improved swarm optimization of deep features," *Scientific Reports*, vol. 10, no. 1, p. 2536, 2020. 1

[3] R. Saikia and S. S. Devi, "White blood cell classification based on gray level co-occurrence matrix with zero phase component analysis approach," *Procedia Comput. Sci.*, vol. 218, pp. 1977–1984, 2023. 2

[4] T. A. E. et al., "Classification of atypical white blood cells in acute myeloid leukemia using a two-stage hybrid model based on deep convolutional autoencoder and deep convolutional neural network," *Diagnostics (Basel)*, vol. 13, no. 2, p. 196, 2023. 2

[5] R. Ahmad, M. Awais, N. Kausar, and T. Akram, "White blood cells classification using entropy-controlled deep features optimization," *Diagnostics (Basel)*, vol. 13, no. 3, p. 352, 2023. 2

[6] G. M. Devi and V. Neelambary, "Computer-aided diagnosis of white blood cell leukemia using vgg16 convolution neural network," in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2022. 2

[7] D. K. et al., "Automatic detection of white blood cancer from bone marrow microscopic images using convolutional neural networks," *IEEE Access*, vol. 8, pp. 142 521–142 531, 2020. 2

[8] S. Ansari, A. H. Navin, A. B. Sangar, J. V. Gharamaleki, and S. Danishvar, "A customized efficient deep learning model for the diagnosis of acute leukemia cells based on lymphocyte and monocyte image," *Electronics*, 2023. 2

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597v1*, 2015. 2

[10] H. L. Potgieter, C. Mouton, and M. H. Davel, "Impact of batch normalization on convolutional network representations," *arXiv preprint arXiv:2501.14441v2*, 2025. 5

[11] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, "Ablation studies in artificial neural networks," *arXiv preprint arXiv:1901.08644v2*, 2019. 6

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014. 6