UNIVERSIDAD MILITAR NUEVA GRANADA

DEEP LEARNING

PROFESSOR

DIEGO RENZA TORRES

STUDENTS

EDWIN ESTEBAN PINEDA BOCIGA
JAVIER STIVEN MUÑOZ CORTES

PROGRAM

TELECOMMUNICATIONS ENGINEERING

DATE

11/10/2025

Blood cell recognition using pre-trained CNN models based on the BloodMNIST and White Blood Cells datasets

Introduction

Human blood is composed of different types of cells, each with specific functions that are vital for the body. Among them are red blood cells (erythrocytes), responsible for transporting oxygen; white blood cells (leukocytes), which participate in immune defense; and platelets, which are involved in coagulation. Within leukocytes, there are several subpopulations such as neutrophils, lymphocytes, monocytes, eosinophils, and basophils whose correct identification is essential for the diagnosis of multiple hematological and infectious diseases.

Traditional analysis of these cells is carried out through peripheral blood smears observed under a microscope. Although this procedure is routine, it requires expert personnel, is time-consuming, and may show variability between observers. Therefore, automating the process through image analysis algorithms has become a research area with high potential clinical impact.

In this context, deep learning offers a promising alternative. Convolutional neural networks (CNNs) have shown great effectiveness in biomedical image classification tasks by directly learning discriminative features from data. More recently, transformer-based architectures have introduced improvements in capturing global relationships within an image, expanding possibilities for precision and robustness in blood cell analysis.

The problem addressed in this work is the classification of microscopic images of blood cells into different biological categories. It is a supervised multiclass learning problem, where the input corresponds to a digital image of a cell and the output to its corresponding label, such as red blood cell, lymphocyte, neutrophil, monocyte, eosinophil, basophil, among others. Solving this task automatically not only speeds up diagnosis but also reduces human errors and increases processing capacity in clinical laboratories.

This work defines the problem, reviews literature related to models previously applied to blood cell classification, and describes the characteristics of the dataset used. In this way, it lays the groundwork for implementing and comparing deep learning architectures in the biomedical domain, highlighting both the benefits and challenges associated with automatic blood image classification.

State of the art

Author(s)	Title	Year	Summary	Hyperparameters
Rakesh Chandra Joshi, Saumya Yadav, Malay Kishore Dutta, Carlos M. Travieso-Gonzalez	An Efficient Convolutional Neural Network to Detect and Count Blood Cells	2022	Proposes an efficient CNN to detect and count red blood cells, white blood cells, and platelets in blood images. Reports high accuracy (70–99%) and low computational complexity, with applications in fast medical diagnostics.	Dataset split into 70% training, 15% testing, 15% validation, with 4,888 images. Used YOLO-v3, 400 epochs, learning rate 0.001, IoU of 50%. Batch sizes from 4 to 16; steps per epoch between 512 and 128 depending on batch size.

Author(s)	Title	Year	Summary	Hyperparameters
César Cheuque, Marvin Querales, Roberto León, Rodrigo Salas, Romina Torres	An Efficient Multi-Level Convolutional Neural Network Approach for White Blood Cells Classification (Diagnostics)	2022	Develops a multi-level classification scheme using Faster R-CNN and MobileNet-based CNNs to identify leukocyte subtypes. Achieves ~98.4% accuracy, highlighting its usefulness as a computer-assisted diagnostic tool.	Learning rate 0.001; β1 = 0.9, β2 = 0.999, epsilon = 0.0000001. Models implemented in Keras and TensorFlow, trained on Kaggle/Google Colab with GPU support.
Arindam Halder, Sanghita Gharami, Priyangshu Sadhu, Pawan Kumar Singh, Marcin Woźniak, Muhammad Fazal Ijaz	Implementing Vision Transformer for Classifying 2D Biomedical Images	2024	Analyzes the use of Vision Transformers (ViT) for biomedical image classification on MedMNISTv2, evaluating four subsets: BloodMNIST, BreastMNIST, PathMNIST, and RetinaMNIST. Results show ViT outperforms existing benchmarks, achieving 97.9% accuracy on BloodMNIST, 90.4% on BreastMNIST, 94.6% on PathMNIST, and 57% on RetinaMNIST.	Batch size = 32, learning rate = 0.00005, optimizer AdamW. Trained for 10 epochs. Accuracy ranged from 97.9% (BloodMNIST) to 57% (RetinaMNIST), surpassing dataset baselines of 96.6% and 53.1% respectively.
Oscar Veloz	Application and Evaluation of Neural Networks for Blood Cell Recognition with the BloodMNIST Database	2025	Evaluates deep neural networks for blood cell recognition using BloodMNIST. Analyzes performance of several models in classification tasks, showing high accuracy and feasibility for clinical use.	Layers: Input (Flatten), Hidden (dense 128 neurons, ReLU), Output (dense 8 neurons, softmax). Images resized to 3232 pixels. Dataset split: 70% training, 10% validation, 20% testing. Trained for 50 epochs with CrossEntropyLoss and Adam (lr=0.001).
Mohamad Abou Ali, Fadi Dornaika, Ignacio Arganda-Carreras	White Blood Cell Classification: Convolutional Neural Network (CNN) and Vision Transformer (ViT) under Medical Microscope	2023	Evaluates ViT against traditional CNNs for automated reading of peripheral blood smears (PBS) to detect hematological abnormalities. Using PBC and BCCD datasets, results show ViT is more effective with scarce data and more robust to noisy images, outperforming CNNs in blood cell classification.	Optimizer Adam, categorical cross-entropy loss, accuracy metric, 10 epochs, 90% training and 10% validation split.

Dataset characteristics

The dataset consists of 17,092 blood cell images, distributed across eight classes corresponding to human blood cell types, including red blood cells, lymphocytes, monocytes, neutrophils, eosinophils, basophils, platelets, and other related categories. The images were obtained from microscopic samples and processed to standardize their format.

Each image is represented in color (RGB) with an original resolution of 28×28 pixels. For this work, the images are resized to 224×224 pixels to make them compatible with modern deep learning architectures such as ResNet, DenseNet, or Vision Transformers, which require higher-resolution inputs. Pixel values are normalized, and data augmentation techniques (rotations, translations, flips) are considered to increase variability and reduce overfitting.

Model used

The dataset is divided into three predefined subsets: training (11,959 images), validation (1,712 images), and testing (3,421 images). This structure ensures experiment reproducibility and allows standardized model evaluation, avoiding bias from manual splitting.
Based on the BloodMNIST dataset, the decision was made to increase the amount of data by incorporating the White Blood Cells dataset. However, this additional dataset includes only five of the eight categories present in BloodMNIST. Due to this limitation, a reorganization of the datasets was carried out in order to unify them, resulting in a combined dataset of 33,725 samples divided as follows: 23,607 for training (70%), 5,059 for validation (15%), and 5,059 for testing (15%). This is a significant increase compared to the original BloodMNIST dataset, which contained only 17,092 samples.

The model used was a pre-trained EfficientNetB0, originally trained on ImageNet. This model is lightweight and efficient, specialized in detecting edges, textures, and shapes making it ideal for image recognition tasks. In our case, it was used to learn the structural differences between blood cell types. EfficientNetB0 has various versions depending on the dataset size, and one important aspect considered was the amount of training data. Thanks to the unified dataset, it was possible to assign over 23,000 images for training.

Additional layers were added on top of the pre-trained EfficientNetB0 model, which are specifically trained on our dataset. This approach helps to prevent both overfitting and underfitting, ensuring the model does not simply memorize the training data especially given the large volume of over 23,000 images.

To validate the dataset's effectiveness, the test set (15%) was used to evaluate the model's classification accuracy on new, unseen data. Accuracy and loss functions were employed to assess the model's performance. The output is a prediction vector with eight positions, each corresponding to one of the categories, assigning a probability score to each. The category with the highest probability is selected as the final prediction. Finally, the code displays a random image along with its actual label and the label predicted by the model.

Methodology

1. Selection of the dataset based on its categories
2. Unification of the dataset
3. Loading the dataset into the execution environment
4. Resizing images to 224x224 pixels
5. Splitting the data into train, validation, and test sets
6. Selection of the pre-trained model (EfficientNetB0)
7. Model validation using accuracy testing
8. Result analysis and image classification

Model diagram

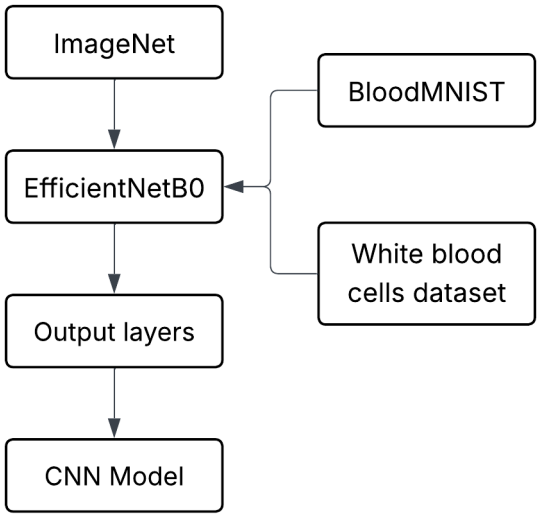

Figure 1. Model diagram.

For the dataset processing, the pre-trained EfficientNet model was used, which was trained on the ImageNet dataset. It was then fine-tuned with the BloodMNIST and White Blood Cells datasets, adding output layers that were adjusted to the eight categories of our selected dataset. Finally, we obtained our CNN model trained on our data (blood cells).

Obtained Results

When comparing our results, we observe that our model achieves better accuracy performance with a shorter training time, using only 10 epochs with batches of 32. In contrast, other models use larger batch sizes and a greater number of epochs, yet obtain lower performance results.

```
history_transfer = transfer_model.fit(
    train_ds,        # Use the training dataset
    epochs=num_epochs, # Use the defined number of epochs
    validation_data=val_ds # Use the validation dataset for validation
)
```

```
Epoch 1/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 245s 221ms/step - accuracy: 0.8342 - loss: 0.5122 - val_accuracy: 0.9569 - val_loss: 0.1360
Epoch 2/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 154s 141ms/step - accuracy: 0.9643 - loss: 0.1053 - val_accuracy: 0.9597 - val_loss: 0.1329
Epoch 3/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 105s 142ms/step - accuracy: 0.9798 - loss: 0.0558 - val_accuracy: 0.9634 - val_loss: 0.1416
Epoch 4/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 144s 145ms/step - accuracy: 0.9868 - loss: 0.0363 - val_accuracy: 0.9644 - val_loss: 0.1341
Epoch 5/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 104s 141ms/step - accuracy: 0.9908 - loss: 0.0309 - val_accuracy: 0.9652 - val_loss: 0.1401
Epoch 6/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 145s 145ms/step - accuracy: 0.9907 - loss: 0.0269 - val_accuracy: 0.9615 - val_loss: 0.1744
Epoch 7/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 105s 142ms/step - accuracy: 0.9903 - loss: 0.0295 - val_accuracy: 0.9628 - val_loss: 0.1780
Epoch 8/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 105s 142ms/step - accuracy: 0.9911 - loss: 0.0303 - val_accuracy: 0.9640 - val_loss: 0.1593
Epoch 9/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 143s 144ms/step - accuracy: 0.9933 - loss: 0.0207 - val_accuracy: 0.9664 - val_loss: 0.1819
Epoch 10/10
738/738 ━━━━━━━━━━━━━━━━━━━━ 103s 140ms/step - accuracy: 0.9957 - loss: 0.0135 - val_accuracy: 0.9634 - val_loss: 0.2031
```

Figure 2. Model training and accuracy.

In Figure 2, we can see the stages and the accuracy and loss percentages for both the training and validation data, achieving an accuracy of 99.57% at the end of the training.

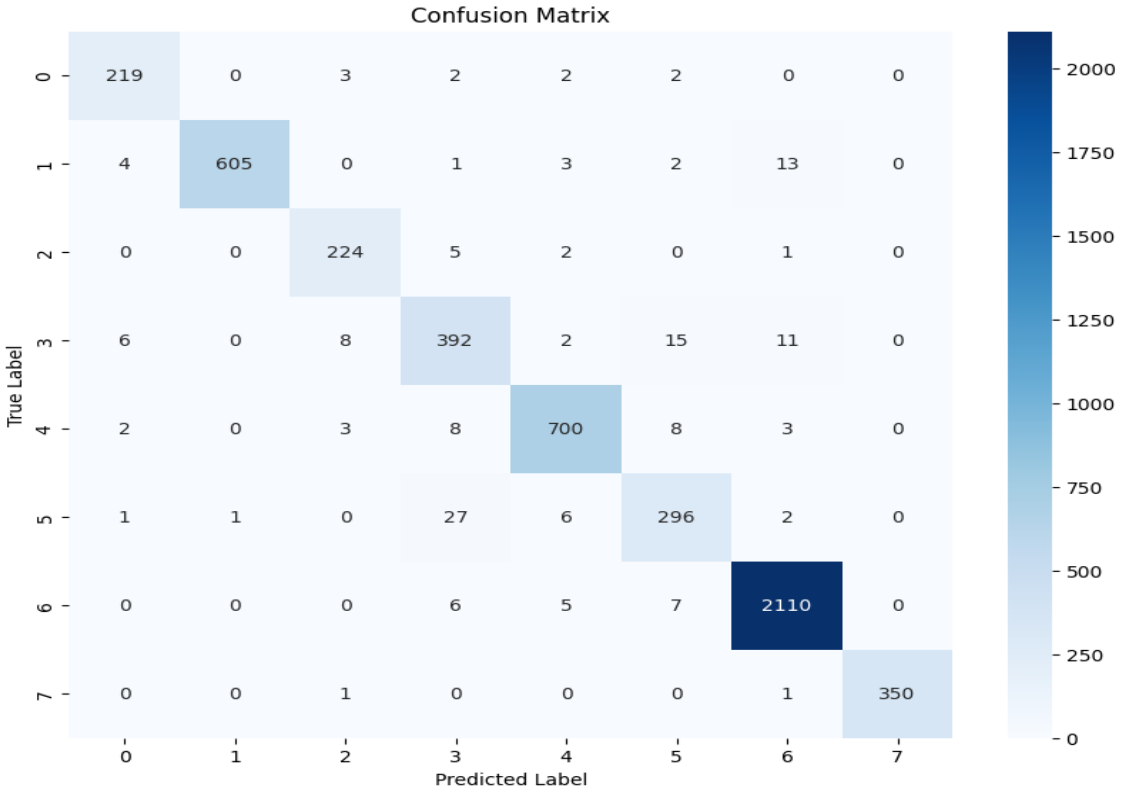

Figure 3. Confusion matrix

In the confusion matrix, we can observe the number of correct and incorrect predictions for each category. The matrix shows that the model has high accuracy in predicting the type of cell based on the eight categories, with minimal errors.

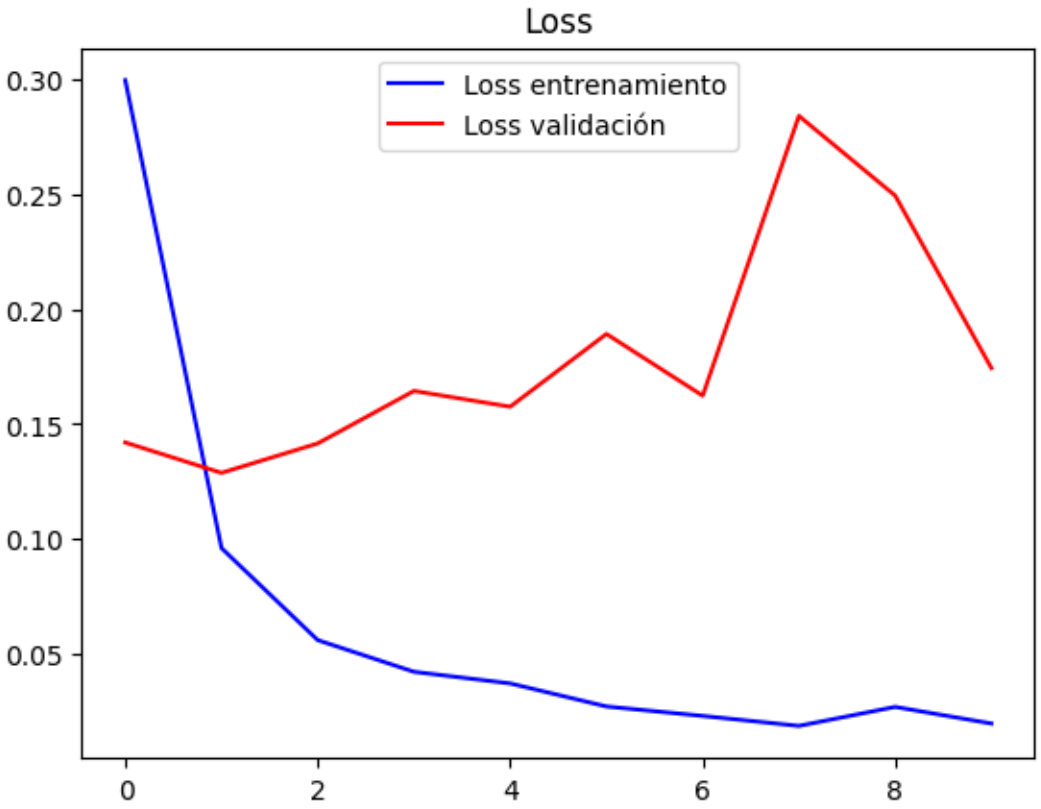

Figure 4. Loss graph for training and validation data.

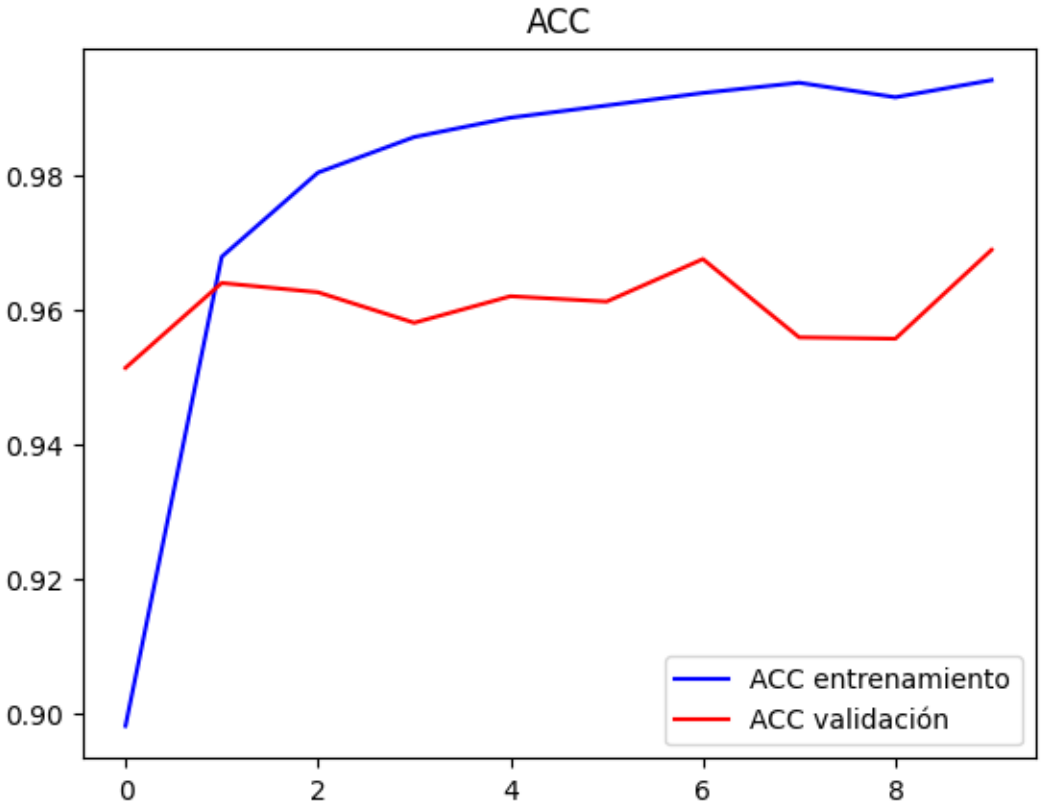

Figure 5. Accuracy graph for training and validation data.

In the accuracy and loss graph, due to the scale of the graph, the curves appear to be distant, but this is due to the minimal differences between the data. The training data achieved an accuracy of 99.57%, while the validation data reached an accuracy of 96.34%, which are quite close. In the end, a random image is displayed along with its actual category and the category predicted by the model, indicating the level of accuracy.

Figure 6. Model testing.

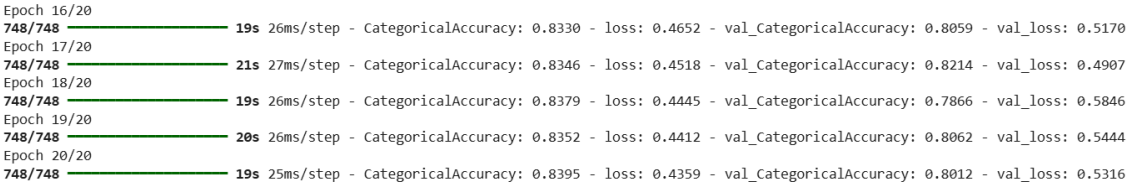

Figure 7. Accuracy with MLP model.

Analysis and Conclusions

By using CNN models instead of MLPs, higher accuracy results are achieved with fewer epochs. In Figure 7, we can see an MLP model, which, despite being trained for twenty epochs, only achieves an accuracy of 83.95%. In contrast, the model using CNNs reached an accuracy of 99.57% after just ten epochs.

By combining two image datasets, we can increase the amount of both training and testing data, allowing us to train the model with more information and achieve higher accuracy. The pre-trained EfficientNet model was very valuable, as being trained on ImageNet allowed it to detect the differences between each type of cell more effectively than if we had trained the model from scratch. This meant that the model already had a high accuracy percentage from the start.

Using CNN models requires more computation time, making the use of GPUs essential to significantly reduce training time. Otherwise, training will take several hours if using a CPU.

References

[1] Abou Ali, M., Dornaika, F., & Arganda-Carreras, I. (2023). White Blood Cell Classification: Convolutional Neural Network (CNN) and Vision Transformer (ViT) under Medical Microscope. *Algorithms*, *16*(11), 525. https://doi.org/10.3390/a16110525

[2] Chandra Joshi, Rakesh, Yadav, Saumya, Kishore Dutta, Malay, & Travieso-Gonzalez, Carlos M.. (2022). An efficient convolutional neural network to detect and count blood cells. *Uniciencia*, *36*(1), 449-457. https://dx.doi.org/10.15359/ru.36-1.28

[3] Cheuque, C., Querales, M., León, R., Salas, R., & Torres, R. (2022). An Efficient Multi-Level Convolutional Neural Network Approach for White Blood Cells Classification. *Diagnostics (Basel, Switzerland)*, *12*(2), 248. https://doi.org/10.3390/diagnostics12020248

[4] Halder, A., Gharami, S., Sadhu, P., Singh, P. K., Woźniak, M., & Ijaz, M. F. (2024). Implementing vision transformer for classifying 2D biomedical images. *Scientific reports*, *14*(1), 12567. https://doi.org/10.1038/s41598-024-63094-9

[5] Veloz, O. (2025). *Application and evaluation of neural networks for blood cell recognition with the BloodMNIST database*. https://doi.org/10.13140/RG.2.2.10588.09609