POLITECNICO
MILANO 1863

AN2DL - First Homework Report
DataDreamers

AmirHossein MohsenNezhad, Shahryar Namdari, Alireza Yahyanejad,

amirmosio, shahryardfp, yahyanejadalre,

249533, 246980, 216895

November 24, 2024

1 Introduction

Blood cell classification is critical for medical diagnosis but traditionally requires manual examination, which is time-consuming, prone to error, and dependent on experts. In this project, we aim to solve this problem with the help of *machine learning* techniques and image processing to propose a scalable and reliable solution, free from human intervention and error. Specially, We focus on training and evaluating a **robust** *deep learning*[5] model leveraging state-of-the-art *augmentation* methods addressing the challenge of **variability** in the context of medical image processing.

2 Problem Analysis

The dataset for this project includes 13,759 RGB images (96x96 resolution) across eight classes: Basophil, Eosinophil, Erythroblast, Immature Granulocytes, Lymphocyte, Monocyte, Neutrophil, and Platelet. The task is to develop a model that classifies each image into one of these categories. A key challenge lies in the variability of medical imaging data, influenced by:

- Patient Specific differences
- Imaging techniques
- Sample preparation
- Environmental condition

Variability in image properties, such as contrast, lighting, and cell morphology, challenges model generalization. Additionally, class imbalance (e.g., underrepresented Immature Granulocytes) risks biasing the model toward majority classes. Preprocessing and augmentation are essential to address these issues and ensure fairness, forming the core focus of this work.

3 Method

Our approach involves several key steps: preprocessing the data, applying data augmentation, training a deep learning model, and evaluating its performance.

3.1 Preprocessing

To ensure the integrity of the dataset, we began by identifying and removing invalid data. An initial examination of 200 images revealed that some images were exact duplicates present in the dataset with different labels, leading to potential label ambiguity. To address this, we computed perceptual hashes (pHash) for each image using the *imagehash* library and identified duplicates by matching these hashes. By removing these exact or near-duplicate images, we reduced potential bias and improved data quality. Figure 1 shows examples of duplicate images that were removed during preprocessing.

To detect any abnormalities or invalid images, we explored outlier detection methods. We first used a pre-trained ResNet model [3] to encode the images into feature vectors and applied principal component analysis (PCA). However, this approach did not yield significant findings, possibly because a ResNet model pre-trained on ImageNet [2] may not capture the subtle features of blood cell images effectively. We then experimented with a simple K-Nearest Neighbors (KNN) algorithm using the most significant principal components of the images.

Our analysis indicated that the class corresponding to label 3 (**Immature Granulocytes**) had the highest number of outliers. While this observation does not lead to definitive conclusions, it suggests that the model may face greater difficulty in learning effective representations for this particular class compared to others.

We also analyzed the class distribution and found a significant class imbalance, with the Immature Granulocytes class being underrepresented. To mitigate this, we calculated class weights using scikit-learn's *compute_class_weight* function, which was later used during model training to address the imbalance.

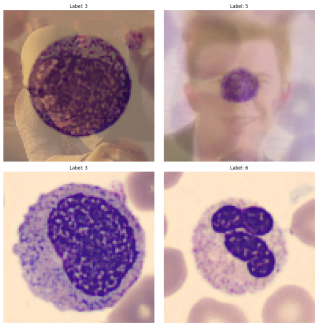

Figure 1: Examples of duplicate images detected and removed during preprocessing.

We also analyzed the class distribution and found a significant class imbalance, with the Immature Granulocytes class being underrepresented. To mitigate this, we calculated class weights using scikit-learn's *compute_class_weight* function, which was later used during model training.

3.2 Augmentation

To address variability and enhance the model's ability to generalize, we applied a combination of advanced data augmentation techniques:

- **RandAugment** [1]: Applies a fixed number of random augmentations, increasing data diversity without manual search.
- **AugMix** [4]: Combines multiple augmentation chains and enforces consistency via a Jensen-Shannon divergence loss between predictions of augmented images.
- **MixUp** [8]: Generates new training samples by interpolating both the features and labels of two random samples.
- **CutMix** [7]: Creates new samples by replacing a random patch of one image with a patch from another and blending their labels based on the patch area.
- **Gaussian Noise**: Adds random noise drawn from a Gaussian distribution to simulate sensor noise and enhance robustness.

These augmentations were implemented using the *KerasCV* library and custom functions. During training, augmentations were applied in real-time to batches of images, ensuring a diverse set of inputs without increasing the dataset size.

3.3 Model Training

We selected EfficientNetB3 [6] as our base model due to its balance between accuracy and efficiency. The model was customized as follows:

- **Base Model Initialization**: Initialized with ImageNet weights, excluding the top classification layer.
- **Custom Classification Head**: Added global average pooling, followed by fully connected layers with ReLU activation and dropout for regularization.
- **Layer Freezing and Unfreezing**: Initially froze all layers of the base model to train only the custom classification head. In subsequent phases, we gradually unfroze the top layers of EfficientNetB3 for fine-tuning.
- **Regularization**: Applied L2 regularization to convolutional and dense layers to prevent overfitting.

The model was compiled with the Adam optimizer and categorical cross-entropy loss. We used early stopping and *ReduceLROnPlateau* callbacks to optimize training and prevent overfitting. Class weights computed during preprocessing were used to address class imbalance.

Model Architecture

The architecture of our model is as follows:

- Input layer with shape (96, 96, 3).
- Preprocessing layer using EfficientNetB3's *preprocess_input* function.
- Base EfficientNetB3 model (without top layers).
- Global Average Pooling layer.
- Dense layers with ReLU activation and dropout for regularization.
- Output layer with softmax activation for the 8 classes.

3.4 Model Evaluation

We evaluated the model on the validation set using metrics such as accuracy, precision, and recall. To assess the model's robustness to noise, we also created a noisy version of the validation set by adding White noise. This simulates real-world conditions where images may have imperfections. We generated confusion matrices to analyze misclassifications and identify any patterns or classes that the model struggled with.

4 Experiments

We conducted experiments to evaluate the impact of the augmentation techniques and fine-tuning on model performance. The key experiments are included in this table 1.

5 Results

The augmented EfficientNetB3 model demonstrated substantial improvements over the baseline. The use of advanced data augmentation and careful fine-tuning led to higher accuracy and better generalization. The model achieved a validation accuracy of 99% and a Codabench accuracy of 81%, significantly outperforming the base model. It also showed robustness when evaluated on the noisy validation set, indicating its potential effectiveness in real-world applications.

While the augmented EfficientNetB3 model achieved significant improvements, the evaluation highlights areas of concern. Specifically, there is frequent misclassification between class 3 (**Immature Granulocytes**) and class 5 (**Monocyte**), likely due to morphological similarities. Additionally, errors are observed between classes 2 (**Erythroblast**) and 7 (**Platelet**) under noisy conditions. These issues suggest the need for enhanced class-specific augmentations or domain-specific features to improve differentiation.

6 Discussion

While the model achieved high accuracy, it functions as a black box, which raises concerns about interpretability in clinical settings. Medical professionals may require explanations for the model's decisions to trust and adopt it in practice. Future work should focus on incorporating explainability methods, such as saliency maps or attention mechanisms.

Additionally, the model was tested on a dataset that, while cleaned and augmented, may not represent the full diversity of real-world medical imaging. Testing the model on more diverse and larger datasets is necessary to assess its true generalizability and to ensure it performs well across different populations and imaging conditions.

7 Conclusions

We developed a deep-learning model to classify blood cell images into eight categories, achieving high accuracy through advanced augmentation and fine-tuning. This work highlights deep learning's potential in automating medical diagnostics.

For real-world deployment, future research should aim to improve the model's interpretability and validate it on more diverse datasets. Enhancing explainability will help gain the trust of medical professionals, and broader validation will ensure the model's applicability across various clinical settings.

8 Contribution

Unfortunately, Kevin's lack of engagement posed significant challenges for the team. While others actively communicated with the team and contributed to model implementation, preprocessing, and report writing, Kevin was almost absent in both areas. His lack of responsiveness led the team to take on extra responsibilities to ensure the project's completion.

Table 1: Results

Model	Val Acc	Codabench Acc	Augmentations
Base Model	87%	21%	Initial Augmentation using ImageDataGenerator from Tensorflow
EfficientNetB3[6]	99%	**81%**	RandAug, AugMix, Mixup, Cutmix, Gaussian noise

References

[1] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] 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*, pages 770–778, 2016.

[4] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[5] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[7] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[8] H. Zhang. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.