AN2DL - First Homework Report
vuoilaglio

Ricardo André Araújo De Matos, Melvin Curinier, Pedro Fonseca, ~~Thomas Lance~~

ricardomatos, mcurinier, pedrofonseca, ~~thomaslance~~

276544, 276717, 276962, 277002

November 24, 2024

1 Introduction

The goal of this project was to develop a **robust deep learning model** for the *image classification* of blood cells into one of eight classes. The dataset consisted of 96x96 RGB images provided as Numpy arrays, with corresponding labels indicating the class of the blood cell.

This is a *multi-class classification task* that required designing, training, and evaluating deep learning models, while tackling challenges such as overfitting and generalization to unseen data.

2 Problem Analysis

The dataset was initially analyzed and was characterized by

- 13,759 images of shape $(96, 96, 3)$.
- Labels ranging from 0 to 7, corresponding to different blood cell classes.
- The cells are blood types specialized in immunity, oxygen transport, and clotting from hematopoietic stem cells.
- **1800 fabricated images** of Shrek and Rick Astley.
- **120 outliers** were identified using the *Isolation Forest Algorithm*, consisting of images that contained either multiple classes or more than one blood cell.
- A set of **similar duplicated images** was removed based on a *cosine similarity* threshold of 99.5%, resulting in the elimination of 200 platelet images.
- To deal with **class imbalance**, two main techniques where experimented: *downsampling* and *augmented upsampling*. In the *augmented upsampling* technique, data augmentation was employed to generate new unique images by applying simple transformations to the original images. However, the model that achieved the best results on the server data was, in fact, **unbalanced**.

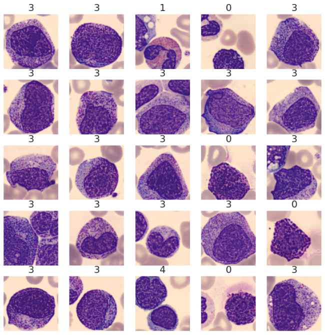

Figure 1: Some of the removed outliers

- Another technique explored was **removing**

the red blood cells from the images. However, due to the differences in the image distributions of the test data, this approach was **not pursued** for the final, more robust models.

Key challenges:

- Some **cleaning techniques** directly impact model quality, but **their effects are hard to assess**. For example, as shown in Figure 1, most detected outliers were from the immature granulocytes class, which was later found to be the most challenging for the model. Therefore, it remains unclear whether removing these outliers improved performance.
- The **different distribution of the testing set** created the necessity of performing strong data augmentation pipelines.

Early experiments highlighted severe overfitting, with local validation accuracies exceeding 95%, but competition results ranging from 20% to 50%. This discrepancy revealed **insufficient generalization**.

3 Method

Multiple deep learning architectures were explored:

- **EfficientNet Models:** Known for its efficiency and performance.
- **MobileNetV2:** Lightweight architecture suitable for small datasets.
- **RestNet Models:** ResNet enables the training of very deep models (e.g., ResNet-50, ResNet-101) while maintaining stable gradients.
- **Basic CNN models:** Used initially to test whether and how the data was being overfitted, and to determine if the model could fully understand the training set. A **dense layer of 128 neurons** was added after *GAP* as we noticed that the both transfer learning models and Basic CNNs where **not able to overfit** the data with only a linear layer of 8 neurons at the output.

Due to limited computational resources on Colab, a decision had to be made regarding the choice of pre-trained architectures. It was found that EfficientNet includes **built-in data augmentation** and **preprocessing** [3]. As a result, **EfficientNetB7** was selected for the final submissions of the competition. That being said, it is important to note that neither of the other architectures was tested with the data augmentation described in section 3.1. Instead, they were tested with simpler data augmentation pipelines that required fewer resources.

3.1 Data Augmentation

To combat lack of generalization, extensive data augmentation was employed[2], including:

- **RandomAug:** Randomly applies multiple augmentations to the input images.
- **CutMix:** Combines patches from multiple images to create new training samples.
- **MixUp:** Generates new samples by linearly interpolating pairs of images and their labels.

Additionally, after each dense layer, a **batch normalization layer** and a **dropout layer** (with a 0.2 probability) were added to prevent overfitting.

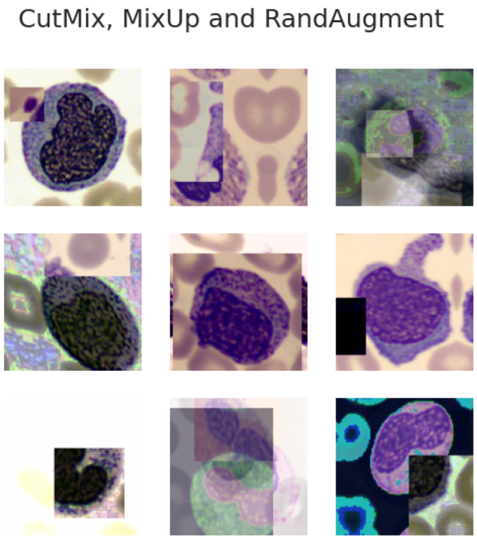

Figure 2: Augmented images

3.2 Training Setup

Models were trained using:

- Optimizer(s): both *Adam* and *RMSProp* were tested.

- Loss Function(s): *Categorical Cross-Entropy* and *Categorical Focal Cross-Entropy*. By using *Categorical Focal Cross-Entropy*, the goal was to make the model more sensitive to underrepresented data points, particularly those that are more difficult to learn, such as the immature granulocytes class [1].

- Batch Size: Varied based on architecture.

- Callbacks: *Early Stopping*, to prevented overtraining, and *ReduceLROnPlateau*, to help the model get to its best parameters when a plateau is reached.

3.3 Test-time Augmentation (TTA)

To enhance model performance, *test-time augmentation* was implemented. In this process, the model predicts 10 times on randomly transformed versions of the same image. After obtaining the predictions, the mean of these predictions is calculated, and the softmax function is applied to obtain the final predicted class. This technique led to an improvement of 2-3% in accuracy on the test set.

Mathematically, the process can be described as follows:

$$\hat{y} = \text{softmax}\left(\frac{1}{N}\sum_{i=1}^{N} f(T_i(x))\right)$$

Where:

- $f(T_i(x))$ represents the model's prediction on the i-th augmented version of the input image x,

- N is the number of augmentations (in this case, 10),

- $\hat{y}$ is the final predicted class after averaging the predictions.

4 Experiments

Multiple models and approaches were evaluated, with the results from local validation and competition submissions summarized in Table 1.

Table 1: Summary of Model Performance.

Model	Local Test Accuracy	Competition Accuracy
EfficientNetV2B3	92.1%	∼45%
MobileNetV2	93.5%	∼50%
FocalLossAug	98.11%	∼89%
Best Model	98.28%	**90%**

It is important to note that the table includes only the most relevant results, with the first two models lacking advanced augmentation and test-time augmentation (TTA). The FocalLossAug model is a fine-tuned EfficientNet-B7 (with transfer learning), where the loss function is *Focal Categorical Cross-Entropy*. In contrast, the best-performing model, also a fine-tuned EfficientNet-B7 transfer learning model, was trained using the *Categorical Cross-Entropy* loss function.

Please note that a dense layer with 128 neurons was added to each model following the *GAP* layer. Experiments on assembly approaches were conducted.

5 Results

The experiments led to the following results:

- A highly accurate model, achieving **90% accuracy on unseen data**.

- **High recall** across most of the classes.

- Unexpectedly, the model **struggles** to capture the characteristics of some **immature granulocytes**.

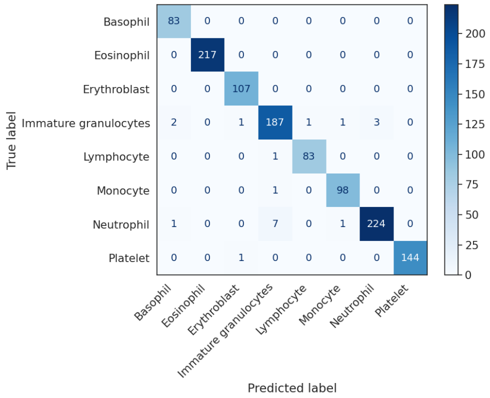

Figure 3: Confusion matrix on the local test set

6 Discussion

The results show:

- The effectiveness of data augmentation techniques (e.g., *CutMix/MixUp*) in reducing overfitting and improving competition performance.
- The observed discrepancies between local validation and competition results, likely due to shifts in data distribution.

Limitations:

- The model **struggles** to understand some characteristics of the immature granulocytes class. If we think on the biological context, this is expected as **immature granulocytes** are **a type of white blood cell that are in the early stages of development**. This means that **immature granulocytes or other new blood cells** may exhibit **similar characteristics** to one another. As demonstrated in Figure 4, some cells closely resemble other white blood cells, and vice versa.
- It is also important to note that these mistakes made by the model are typically associated with **low confidence levels**, as shown in Figure 5. Although we experimented with applying multiple TTAs (20x) to these images, the results did not lead to an increase in test server accuracy.

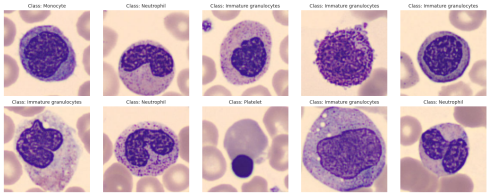
Class: Monocyte
Class: Neutrophil
Class: Immature granulocytes
Class: Immature granulocytes
Class: Immature granulocytes
Class: Immature granulocytes
Class: Neutrophil
Class: Platelet
Class: Immature granulocytes
Class: Neutrophil

Figure 4: Some wrong guesses on local test set

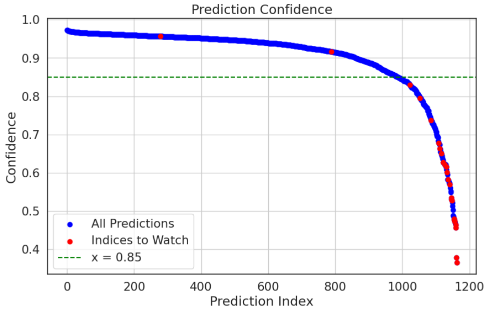

Figure 5: Predictions sorted by confidence, with red dots marking the misclassified instances.

7 Conclusions

A model was successfully developed that achieved 90% accuracy in the competition by mitigating overfitting with advanced data augmentation techniques. Future work may explore:

- Investigate ensembles of **negatively correlated models**, allowing some to specialize in identifying different patterns. For example, one model could be trained to excel in detecting immature granulocytes.
- **Explore fine-tuning other models** beyond EfficientNet-B7. Due to computational limitations on Colab, only one model was fine-tuned in this work.

8 Contributions

The team is composed of Erasmus students from diverse backgrounds, each making equal contributions. In this section, we outline the specific contributions:

- Melvin focused on experimenting with various pre-trained models, architectures, and data cleaning techniques.
- Pedro concentrated on investigating data augmentation pipelines and evaluating their effectiveness across different dataset distributions.
- Ricardo was responsible for designing and implementing various architectures, data augmentation and cleaning pipelines, test-time augmentation (TTA) methods, and ensemble strategies.
- Thomas didn't contribute towards the project.

References

[1] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection, 2018.

[2] lukewood. Cutmix, mixup, and randaugment. `https://keras.io/guides/keras_cv/cut_mix_mix_up_and_rand_augment/`, April 2022.

[3] M. Tan and Q. Le. Efficientnetv2: Smaller models and faster training. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10096–10106. PMLR, 18–24 Jul 2021.