

AN2DL - First Homework Report

DeepL

Matteo Bonfadini, Elena Lippolis, Lorenzo Cossiga, Michele Baggi

50mgk, elenali, lorenzocossiga, mik01

243786, 252310, 242309, 252119

November 22, 2024

1 Introduction

The project aims to implement *multi-class image classification* using **deep learning** techniques for blood cell analysis. This *supervised learning* task requires balancing model complexity to ensure generalization to unseen data while minimizing *overfitting*. The primary **goal** is to design a solid neural network that classifies blood cells into the correct categories with high accuracy.

The **approach** involves dataset analysis to identify potential issues such as class imbalance, image quality, and the need for preprocessing. We then developed simple deep learning models to establish a performance benchmark, which was subsequently refined into more complex models. We applied techniques such as data augmentation, transfer learning, and fine-tuning to improve model accuracy and generalization and to reduce overfitting. By following this approach, we were able to develop a classification model for blood cells.

2 Problem Analysis

The **dataset** consists of 13759 RGB images designed for the classification of different types of blood cells. Each image has size 96x96 pixels and it is labeled with one of eight (from 0 to 7) classes, representing the following blood cell types: basophils (0), eosinophils (1), erythroblasts (2), immature

granulocytes (3), lymphocytes (4), monocytes (5), neutrophils (6), and platelets (7). We show in Figure 1 a sample for each class.

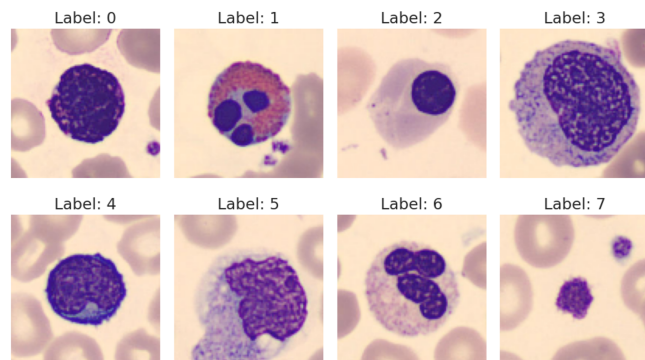


Figure 1: Example of an image for each class

We tackled the problem in a 3-phase approach.

2.1 Phase 1: Data Inspection

Firstly, we inspected the dataset and printed its shape, as shown in Table 1. Secondly, a deeper inspection revealed the presence of *obvious* outliers, as shown in Figure 2, which could skew the learning process: 200 samples of the very same blood cell for each class and 200 rick roll. Therefore we proceed by erasing those images.

We also noticed some images with grayscale backgrounds, but further analysis through *PCA*

based on the Mahalanobis distance [3] (normalizing the dataset) showed this pattern did not negatively influence our performance. This indicates that the grayscale background images do not impact the model’s ability to learn effectively.

Table 1: Shape of the dataset.

	Class	Number of samples
	0	1052
	1	2381
	2	1285
	3	2226
	4	1049
	5	1393
	6	2530
	7	1843
Total:	8	13759

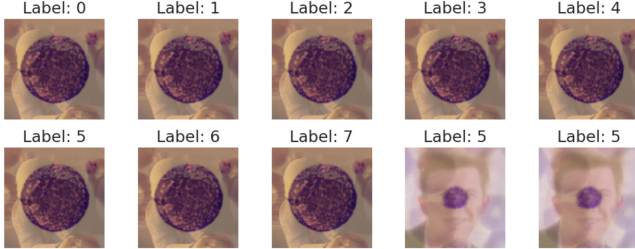


Figure 2: Samples of *obvious* outliers.

2.2 Phase 2: Class Imbalance

The new dataset, for a total of **11959** blood cell images, has shape reported in Figure 3.

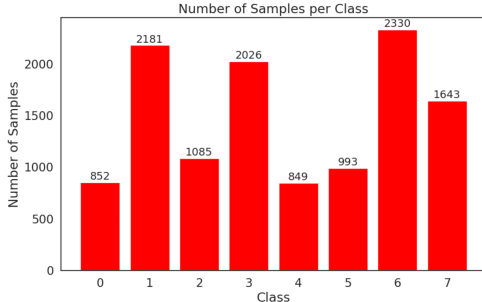


Figure 3: Number of samples in each class.

Upon closer inspection, one might observe that the dataset exhibits class imbalance. This problem poses additional challenges for training a model that generalizes effectively across all categories.

To address this, after the splitting of the dataset into training, validation and test sets according to an 83:13:4 ratio, we balanced the training one by replicating images in the underrepresented classes. The result is a new training set with 2000 images for each class.

Finally, we applied **data augmentation techniques** to create variations of these images (*oversampling*) [7]. In particular, after trials and errors, we decided to use **AugMix** and **RandAugment** [5].

2.3 Phase 3: Mixed Precision

One of the main **challenges** we encountered was the low training speed, which was solved by implementing mixed precision [2]. This technique uses lower precision (16-bit floating point numbers) for computations while retaining single precision (32-bit) for weight updates. Mixed precision accelerated training and reduced energy consumption, all while maintaining a reasonable accuracy level. Additionally, overfitting of the training set reduced the model’s ability to perform well on unseen (and possibly *modified*) test data. However, we intentionally allowed initial overfitting assuming that we needed to verify that the model was able to fit the training data before addressing generalization issues, to employ only later techniques like early stopping and dropout layers to mitigate overfitting and enhance performance on unseen data [6].

2.4 Assumptions

We **assumed** the dataset to be solid, with minimal labeling error or inconsistencies beyond the identified outliers. Moreover, we hypothesized that a simple baseline model would provide reasonable performance before moving to more complex architectures and that some overfitting was necessary at first, and we would have to address it later.

The last assumption we made was that Codabench would not let us down when we needed it to test our final models ... apparently that was a bit too much to ask for.



3 Method

The task suggests to use Categorical Cross-Entropy as loss function. If y_{ij} is the output probability of class j for observation i , then:

$$\text{Loss}(y_{\text{pred}}, y_{\text{true}}) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^J y_{\text{true},ij} \log(y_{\text{pred},ij})$$

where N is the number of samples and $J = 12$ is the number of classes.

As metric we used Accuracy as requested:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{[\text{argmax}_j(y_{\text{pred},ij})=\text{argmax}_j(y_{\text{true},ij})]}$$

In addition, we also considered the F1-score per class to better understand the weaknesses of the model.

Layer
Layer
Dense(128)
BatchNormalization
Dropout

4 Experiments

For your experiments, you might want to present your results in tables. Here's an example of a wide table comparing different models:

For more specific measurements, you might use a narrower table:

Table 3: An example of table. Best results may be highlighted in **bold**.

Time [μs]	Distance [mm]
22 \pm 4	8 \pm 1
17 \pm 3	7 \pm 1
15 \pm 3	6 \pm 1
13 \pm 2	5 \pm 1
10 \pm 2	4 \pm 1
8 \pm 2	3 \pm 1
5 \pm 1	2 \pm 1
37 \pm 1	1 \pm 1

You can also include figures to visualise your results:



Figure 4: Example figure showing [describe what the figure shows]

Reference figures using like:“As shown in Figure 4...”

5 Results

Present your main findings here. You might want to:

- Compare your results with baselines
- Highlight key achievements using **bold text**
- Explain any unexpected outcomes

6 Discussion

In this section, analyse your results critically. Consider:

- Strengths and weaknesses
- Limitations and assumptions

7 Conclusions

Summarise your work and discuss potential future directions. This is where you can:

- Restate main contributions
- Suggest improvements
- Propose future work

8 Logbooks

Here we can write our personal logbooks.

8.1 Matteo

- Google Drive folder
- L^AT_EX repository
- A bit of data inspection and noised aummentation

8.2 Lorenzo

-

8.3 Michele

-

8.4 Elena

-

9 Intro

- This project focuses on image classification about blood cells
- goal: create network to recognize blood cells
- approach: understand the dataset and its problems, start with simple models, discover augmentation & class balance, transfer learning, fine tuning

10 Problem analysis

- Dataset characteristics: presenza di ovvi outlier (200 x classe + 200 rick roll classe 5), classi non bilanciate
- Main challenges: lentezza allenamento (mixed precision), problemi di generalizzazione al test set (overfitting), codabench
- Initial assumptions: che il codabench non funzionava;

11 Method

- Bilanciamento classic
- augmentation
- modello (VGG16)
- fine tuning

References:

- Use of VGG16 on medical images: [4], [1]
- Balance of classes through replication
- Lecture 5: Facing Overfitting, October 2: "A good practice for designing neural networks: train a model that overfits the data. This ensures the model has enough complexity (e.g. sufficient layers, neurons, and parameters) to solve the problem effectively. At this stage, generalization is not the focus: it's about verifying that the model is capable of fitting the training data. Once overfitting is confirmed, techniques like early stopping and dropout layers are introduced to prevent overfitting and improve generalization to unseen data."

References

- [1] A. Acevedo, S. Alf  rez, A. Merino, L. Puigv  , and J. Rodellar. Recognition of peripheral blood cell images using convolutional neural networks. *Computer Methods and Programs in Biomedicine*, 180:105020, 2019.
- [2] M. D  rrich, M. Fan, and A. M. Kist. Impact of mixed precision techniques on training and inference efficiency of deep neural networks. *IEEE Access*, 11:57627–57634, 2023.
- [3] H. Ghorbani. Mahalanobis distance and its application for detecting multivariate outliers. *Facta Universitatis Ser. Math. Inform.*, 34, 2019.
- [4] Q. Guan, Y. Wang, B. Ping, D. Li, J. Du, Y. Qin, H. Lu, X. Wan, and J. Xiang. Deep convolutional neural network vgg-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study. *J Cancer*, 10(20):4876–4882, 2019.
- [5] lukewood. Cutmix, mixup, and randaugment image augmentation with kerascv. Online available at https://keras.io/guides/keras_cv/cut_mix_mix_up_and_rand_augment, April 2022.
- [6] M. Matteucci. Facing overfitting. Online available at <https://politecnicomilano.webex.com/politecnicomilano/ldr.php?RCID=066c548095ab281858fd7a2913ed13c5>, October 2024.
- [7] K. T. Shorten, C. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 2019.