AN2DL - First Homework Report
Four Neurons

Daniel Silva Barradas, Rigers Hado, Christian Mariano, Alessia Testa

danisilva, rigersh01, christian, alessia

281330, 252621, 260338, 259852

March 1, 2025

1 Introduction

This project focuses on the *image classification* of various blood cell types using **deep learning** techniques. The task is to develop an accurate model that can *classify images of blood cells into one of eight classes*, each representing a different cell type. The goal is to apply deep learning models, **starting from a basic convolutional neural network (CNN) and moving towards more sophisticated techniques** such as transfer learning and data augmentation, to improve the model's generalization and performance.

2 Problem Analysis

2.1 Dataset Characteristics

The dataset consists of 13,759 RGB images of blood cells, with each image labeled into one of eight categories: Basophil, Eosinophil, Erythroblast, Immature Granulocytes, Lymphocyte, Monocyte, Neutrophil and Platelet. The dataset presents the following key characteristics:

- **Image Size:** 96x96 pixels
- **Color Space:** RGB (3 channels)
- **Input Shape:** (96, 96, 3)
- **Number of Classes:** 8

This training set is however imbalanced, as is contains a higher amount of images belonging to classes 1, 3 and 6, while classes 0, 2 and 4 have fewer samples.

3 Method

To tackle this issue we opted to apply both **undersampling** and **oversampling**.
We start by simultaneously shuffling the data for both labels and images.
Since the *shuffle* is random, selecting the first 850 images from each class will always result in a random sample. In oversampling we repeat the the sampling process to get a bigger sample.
This augmentation process generates a new dataset, $\mathcal{D}_{\text{aug}}$, derived from the original dataset $\mathcal{D}_{\text{orig}}$. To further enhance fairness during training, the data is shuffled to avoid bias.
The goal is to balance two critical aspects of deep learning model performance: robustness to data variability and high accuracy on unseen data. A key factor in achieving this balance lies in the design of the augmentation pipeline for the training dataset. This involves introducing controlled randomness and applying diverse transformations (such as rotation, flipping, scaling, cropping, color jittering, and noise addition) to mimic real-world variations and distortions.

Augmentations like GridMask [1] and RandomColorJitter [2] force the model to learn more robust features. These techniques, coupled with RandAugment [2], an augmentation policy that applies a random selection of predefined augmentations , generate significantly diverse versions of the input images, improving the generalization of the model.
Similarly, **AugMix** [1] mixes augmented versions of the original image, **RandomColorDegeneration** [3] makes the model robust to changes in color information and **Fourier Mix** [4] alters frequency content, reducing over-reliance on specific image details by blending features from different images. This mimics scenarios where textures or patterns from one image influence another. It is especially significant in medical imaging, where results are sensitive to scanning devices and imaging protocols.
Splitting the dataset into training, validation, and testing ensures that a portion of the data remains exclusively for final evaluation, untouched by training or validation processes. Applying augmentation to validation data evaluates the model's resilience in more challenging scenarios, striking a balance between improving robustness and avoiding overfitting.
Deep learning models require large datasets and significant computational resources to achieve high performance. Pretrained models offer a practical solution by leveraging models that have already been trained on extensive datasets, such as ImageNet.The ConvNeXt Base model [7] uses a softmax activation function in the output layer. This complements the categorical crossentropy loss function, which is well-suited for multi-class classification problems, ensuring stable and interpretable training [8].
The model is fine-tuned by retraining specific layers while keeping feature extraction layers frozen, enabling it to adapt pretrained features to the target dataset. The updated model is trained just for 3 epochs with a low learning rate to prevent drastic weight updates, ensuring the stability of the pre-trained features and avoiding overfitting.

4 Experiments

The experiments leaned more on data pre-processing and data augmentation.
This is the outline of our project development phase

- Building our own model
- Use of a pre-trained model for transfer learning
- Fine-tune said model
- Data augmentation to improve performance

The experimentation time was mostly spent on trying to figure out the effect of different image augmentation and sampling techniques. By tinkered with oversampling (O on the table), under sampling(U on the table) and various data augmentation techniques while keeping some stable elements such as ColorDegradation and classic geometric transformation while keeping the number of epochs low (10-15) and using various optimizers such as Lion [9] and Adam.

5 Results

Despite the accuracy stated in table1 the best result in the Codabench test was given by a 0.75 CNBaseO AugMix+RandAugment,a 0.73 and a lot of 0.72 achieved in more occasion with the worse achiever being our own model with 0.25. **The biggest jump in accuracy was given by switching to a pre-trained model and augmenting the data**. Notably, the models exhibit distinct behaviors when subjected to undersampling and oversampling. **In the case of undersampling, the model shows a tendency to significantly overfit the validation data, whereas oversampling leads to the opposite effect, with the validation loss being considerably higher**.
Model loss with under sampling:

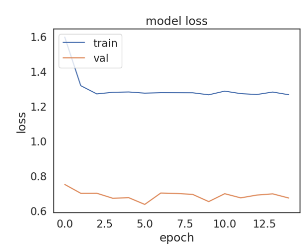

Figure 1: CNbase augmix + Fourier

Model loss with over sampling:

Table 1: **Scores obtained against an unseen and non augmented test set.**

Model	Accuracy	Precision	Recall	F1 Score
Custom Model unbalanced datset	0.8185	0.8707	0.8185	0.834
CNBaseU AugMix+RandAugment	0.9301	0.934	0.9301	0.9307
CNBaseU AugMIx+Fourier	**0.9443**	**0.9455**	**0.9443**	**0.9437**
CNBaseO AugMix+RandAugment	0.9164	0.9167	0.9164	0.9161
CNBaseO AugMix+fourier	0.9164	0.9179	0.9164	0.9157

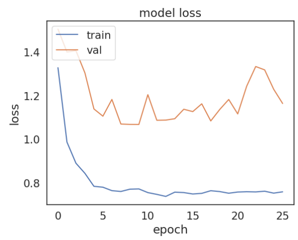

Figure 2: Model CNbase augmix + Fourier

6 Discussion

CNN architecture provided initial insights, but it required careful design to extract robust features and our architecture lacked the depth and sophistication needed for complex patterns in the data. Furthermore training a CNN from scratch is computationally expensive and time-consuming, especially without leveraging pretrained weights making the network less efficient. Switching to a pre-trained model and fine tuning it helped achieving better results. This improvement was crucial for achieving higher accuracy in our dataset without sacrificing efficiency. Nonetheless the model still struggles with classifying outliers and could be further improved.

7 Conclusions

In summary, this project demonstrates the efficacy of transfer learning and pre-trained models for achieving strong performance on a classification task with limited resources. However, there remains significant scope for improvement through more extensive experimentation with sampling methods, hyperparameters, and advanced training strategies.

- One notable challenge was balancing between under-sampling and over-sampling in the dataset. This imbalance may have contributed to variability in the loss function during training. A more nuanced sampling technique, such as a hybrid or middle-ground approach, could mitigate this issue, leading to better generalization. Additionally, our choice to freeze most layers in the pre-trained model was driven by computational limitations. With greater resources, experimenting with smaller learning rates and gradually unfreezing additional layers could allow for a more thorough fine-tuning process. This might enable the model to adapt better to the specific nuances of our dataset, yielding higher accuracy and improved convergence.

- Given additional time and resources, several directions could be explored to further improve the performance and robustness of the model:

  - Cross-Validation: Implementing cross-validation during data splitting would ensure a more even distribution of data across training and validation sets, reducing the likelihood of overfitting and providing a more reliable evaluation metric.

  - Optimization of Hyperparameters: Greater exploration of learning rates, batch sizes, and the number of epochs could fine-tune the model's performance further. Fine-grained adjustments might better balance training stability and convergence.

All the member participated in both report and the development of the model , with vast majority of the development and writing happening through zoom-meetings while screen sharing. In particular Rigers and Christian slightly more focused on the model and Daniel and Alessia a little more on the report.

References

[1] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty." *arXiv preprint arXiv:2001.04086*, 2020.

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space." *arXiv preprint arXiv:1909.13719*, 2019.

[3] Keras Team. "Random Color Degeneration Layer in Keras CV." Accessed November 2024. Available: `https://keras.io/api/keras_cv/layers/augmentation/random_color_degeneration/`

[4] Dan Hendrycks and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." *arXiv preprint arXiv:1912.02781*, 2019.

[5] Ekin D. Cubuk, Alexey Dosovitskiy, Ting Chen, Barret Zoph, Doron Holtzman, and Geoffrey Hinton. "RemixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring." *arXiv preprint arXiv:2002.12047*, 2020.

[6] Sangwoo Kim, Dongyoon Han, Sangdoo Yun, Byeongho Heo, and Sanghyuk Chun. "ViT-Mix: Improving Vision Transformers via Mixed-Token Training." *arXiv preprint arXiv:2201.03545*, 2022.

[7] Liu, Zhuang and Mao, Hanzi and Wu, Chao-Yuan and Feichtenhofer, Christoph and Darrell, Trevor and Xie, Saining. "A convnet for the 2020s" *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* pp.11976–11986, 2022.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] Chen, X., Liang, C., Huang, D., Real, E., Wang, K., Pham, H., ... Le, Q. V. (2024). "Symbolic discovery of optimization algorithms." *Advances in neural information processing systems, 36.*