

AN2DL - First Homework Report

NeuralDropouts

Pinar Erbil, Sergio Pardo, Angela Remolina, Matteo Sissa

perbil, sergiopardo, angelaremolina, matteosissa

244638, 243066, 242814, 247064

November 24, 2024

1 Introduction

From healthcare to transportation, AI-powered systems are enhancing efficiency, accuracy, and accessibility in ways previously unimaginable. Technologies like deep learning and deep neural networks have also allowed huge improvements in the computer vision field. Medical Image Analysis falls in-between these two areas, it implements cutting edge deep learning models to boost the capability humanity has of studying our body and its composition. This is the context of the present project, which aims to **classify images of blood cells among 8 possible classes**.

2 Problem Analysis

Classifying blood cells is a complex task due to the subtle differences among various cell types, which can lead to significant challenges in accurate identification. The analysis of the problem starts with the inspection of the available dataset which is composed of 13759 96x96 RGB, all labeled with an integer value ranging between 0 and 7 to representing the 8 classes of blood cells. The correctness of the labels provided in the original dataset, presumably derived from expert knowledge, was assumed throughout the project.

3 Method

3.1 Data wrangling

First, the dataset was inspected using **Principal Component Analysis (PCA)** [2] to identify outliers and erroneous data points. In this analysis, it was clear that

the points located farther from the origin were identified as outliers. See figure 1.

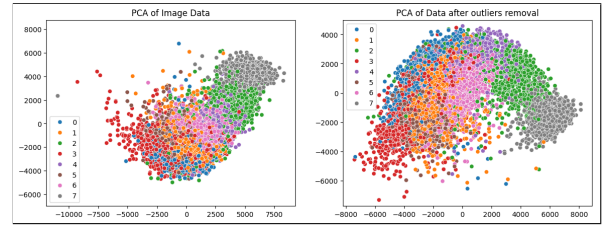


Figure 1: PCA before and after outlier removal

Upon examining these points, it becomes evident that the organizers of the competition included several irrelevant samples. Specifically, a total of 200 images of meme 1 and 1600 images of meme 2 were found, amounting to 1800 samples that were deleted from the dataset.

3.2 Data Augmentation

Augmentation turned out to be a key step to mitigate overfitting, given the variability in imaging conditions and the limited size of the original dataset.

The first augmentation attempt was trying to highlight key features of the cell like, removing background and showing more clear edges, see 2.

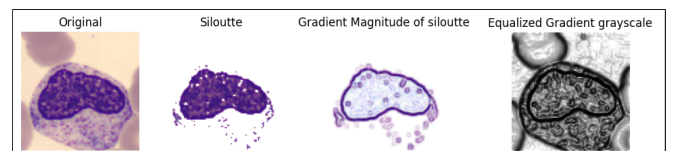


Figure 2: Edge highlighting augmentation

These looks really good to the human eye, but the performance obtained by the model were not good, so this technique was discarded.

The augmentation step of the dataset was thoroughly re-designed, applying more advanced techniques like FourierMix [5], and CutMix [9]. See figure 3 These augmented samples plus the original one were fed to the model for training.

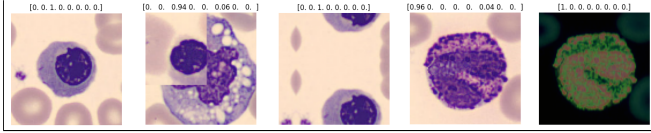


Figure 3: FourierMix, CutMix, RandAugment

3.3 Model design and transfer learning from pretrained model

Instead of training a custom model from scratch, **transfer learning** was applied. This approach is advantageous as it exploits the strong feature extraction capabilities of a pre-trained large model that are crucial for distinguishing subtle differences in blood cells with a limited dataset.

ConvNextXLarge [3], trained on the ImageNet dataset [1], was selected to provide the feature extraction network for the classifier of this project. It is well-suited for the goal because of its hierarchical feature extraction capabilities, able to capture fine details both at low and high level, and its robustness to image noise, very common in biomedical imaging. [7]

At the end of the ConvNextXLarge feature extraction network, there is a **custom neural network classifier** with three additional dense layers and a final output layer composed by eight output neurons corresponding to each class. The softmax function is used to normalize the output layer to a probability distribution over the classes. Dropout and batch normalization techniques are also applied in these last layers to make the model faster to train and more robust.

3.4 Training, callbacks and finetuning

The cleaned dataset was divided into training (90%), validation (5%), and test (5%) sets, and the augmentation pipeline from Section 3.2 was applied to expand the training set. Training was conducted in two phases: transfer learning and fine-tuning.

In the **transfer learning** phase, the feature extraction network (ConvNextXLarge) weights were frozen to train the classifier at the model’s output. The training used categorical cross-entropy loss, the Adam optimizer [8], and parameters of 15 epochs and a batch size of 64, which provided effective results. Early-stopping and Reduce Loss Rate on Plateau callbacks were employed to prevent overfitting and enhance convergence.

The **fine-tuning** phase unfroze all but the first 124 layers of the ConvNextXLarge model to refine it for the specific task. Training settings mirrored the transfer learning phase. Final evaluations on the test set included accuracy, precision, recall, F1 score, and a confusion matrix to assess performance.

4 Experiments

4.1 Early experiments

Experimentation first started from a very basic convolutional neural networks with a few layers to assess the complexity of the problem and identify the main concerns to be addressed. The initial trials only had two convolutional layers and a single dense layer at the end, and were working with unclean data. Although, the local training results were promising, on CodaBench the accuracy were below 20%.

These attempts highlighted the importance of inspecting the data in detail and increasing the amount to control overfitting.

PCA was therefore employed as it was mentioned on section 3.1.

4.2 Mid-stage experiments

After establishing a clean data set, transfer learning was implemented using VGG [6], and the ConvNext family (Base, Large and XLarge) [4] as possible model options. They were all used as feature extraction networks, completing them with an input layer suitable for the input images of the problem and a classification network at the end composed of three dense fully-connected layers.

Several different settings were employed while using these models. Initially, simple models with no augmentation were implemented. Then, fine-tuning and augmentation were utilized in the pipeline for the more complex models.

This prove that using transfer learning leads to better performances, since the accuracy incremented from 20% to 85%.

4.3 Late experiments

In the late-stage experiments, it became evident that ConvNext models consistently outperformed VGG. However, there was no definitive superiority among the ConvNext models themselves, as they all demonstrated comparable levels of accuracy. Upon closer analysis, it was determined that the key factor influencing performance was the augmentation techniques applied. To investigate this further, a final phase of experimentation was conducted where augmentation strategies were significantly enhanced, selecting the best base model so far (ConvNextXLarge). Due to memory constraints, it was

not feasible to apply a large number of augmentations simultaneously. Therefore, a block-based approach was implemented. In this method, augmented samples were sequentially fed into the model, and once utilized, they were removed from memory to make room for subsequent batches. This approach allowed for a more extensive exploration of augmentation effects while adhering to the system’s limitations.

5 Results

All the detailed results of the performed experiments are presented in table 7 with their corresponding settings.

The final model presented in this paper achieved an overall **accuracy of 87%** on test data. Compared to the initial attempts that were made with very simple models, this shows an improvement of **around 70 percentage points** throughout the design process for this solution. The two most important contributions to this achievement can be attributed to using **augmentation** and **transfer learning** techniques to craft a precise solution for the problem at hand.

6 Discussion

The final achievement shows a robust solution to distinguish blood cells even in the case of classes with very low variability and subtle differences. At the same time, the reliance on pre-trained weights from ImageNet, a non-biomedical dataset, could limit the model’s ability to capture domain-specific features, particularly those unique to blood cell microscopy images.

7 Conclusions

In this work, we proposed a robust solution for blood cell classification leveraging transfer learning with the ConvNeXt-XLarge backbone and advanced data augmentation techniques.

The approach effectively addressed challenges, achieving competitive performance across most cell types. It was proven that by carefully balancing class weights and fine-tuning key hyperparameters (such as the number of epochs, batch size, patience, ...) it was possible to achieve the same accuracy (85%) with a ConvNext-Base model as with a ConvNext-Large model. Given that ConvNext-Base is significantly lighter and more computationally efficient, it presents a compelling alternative to ConvNext-Large, offering equivalent performance while requiring fewer resources. This underscores the effectiveness of optimization techniques in maximizing the efficiency of machine learning models.

Future work could focus on generating synthetic data for rare classes or collecting additional images for easier and more precise training. It is also possible to experiment with lighter models for transfer learning and verify the change in performance.

8 Team contributions

Angela focused on data wrangling and implementing initial data augmentation techniques, including edge highlighting and developing the base model. **Pinar** worked on the ConvNext family models (Base, Large and XLarge models), applying fine-tuning to improve performance. **Sergio** contributed by building and fine-tuning all VGG-based models, aiding comparative analysis. **Matteo** tackled memory constraints through on-the-fly augmentation, optimizing the ConvNextXLarge model for better results. **All team members** collaborated on the project equally ensuring a balanced workload and a cohesive submission.

Model	Data	Augmented	Fine-tuning	Local Results				CodaBench
				Accuracy	Precision	Recall	F1	Accuracy
Base Model	Unclean	No	No	0.7	0.72	0.7	0.68	0.14
VGG	Clean	No	Yes	0.87	0.91	0.87	0.88	0.49
ConvNextBase	Clean	Yes	Yes	0.97	0.97	0.97	0.97	0.85
ConvNextLarge	Clean	Yes	Yes	0.98	0.98	0.98	0.98	0.85
ConvNextXLarge	Clean	No	No	0.95	0.95	0.95	0.95	0.74
ConvNextXLarge	Clean	Yes	Yes	0.98	0.98	0.98	0.98	0.87
Block ConvNextXLarge	Clean	Yes	Yes	0.99	0.99	0.99	0.99	0.79

Table 7: Summary of the models’ performances

References

- [1] 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.
- [2] I. T. Jolliffe and J. Cadima. *Principal Component Analysis*. Springer, 2016.
- [3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [4] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.
- [5] Y. Rong, L. Zhang, and C. Liu. Fouriernet: Resolving dataset bias via fourier data augmentation. *arXiv preprint arXiv:2106.01287*, 2021.
- [6] K. Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [7] K. Team. Applications in keras, 2024. Accessed: 2024-11-23.
- [8] B. Warner. Which optimizer should i use?, 2024. Accessed: 2024-11-23.
- [9] 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 (ICCV)*, pages 6023–6032, 2019.