AN2DL - First Homework Report
NeuralDropouts

Pinar Erbil, Sergio Pardo, Angela Remolina, Matteo Sissa

perbil, sergiopardo, angelaremolina, matteosissa

244638, 243066, 242814, 247064

November 23, 2024

1 Introduction

Artificial intelligence has revolutionized the world we live in, the way we look at things, and the problems we aim to solve. 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. It 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 labelled with an integer value ranging between 0 and 7 to representing the 8 classes of blood cells, respectively Basophil, Eosinophil, Erythroblast, Immature granulocytes, Lymphocyte, Monocyte, Neutrophil, Platelet.

The correctness of the labels provided in the original dataset, presumably derived from expert knowledge, was assumed throughout the project. As for the dataset size, it was uncertain at first whether it would have been possible to rely solely on the original data without any augmentation technique. The first experimentation were therefore conducted without any augmentation to verify the system response and performance.

3 Method

To address the classification problem, the following methodology was employed.

3.1 Data wrangling and augmentation

First, the dataset was inspected using **Principal Component Analysis (PCA)** [2] to identify and remove outliers and erroneous data points.
The cleaned dataset was split into training set (90%), validation set (5%) and test set (5%) and then augmentation techniques were applied to increment the number of samples in the training set.
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. So, the augmentation step of the dataset was thor-

oughly redesigned, applying more advanced techniques like FourierMix [5], and CutMix [9]. Later, augmented dataset and one-hot encoded labels were fed to the model for training.

3.2 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 input side, the feature extraction network was enriched with an input layer compliant with the image size (96x96x3) and an additional augmentation layer that applies random transformations as data is fed into the system. This enhances variability in the dataset and mitigates overfitting.

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.

3.3 Training, callbacks and finetuning

The training part was split into two parts: transfer learning and fine-tuning. In the **transfer learning** phase, the weights of the feature extraction network (ConvNextXLarge) were frozen to ensure stability and allow the training of the classifier network at the output of the model. The loss function adopted is categorical cross-entropy and back-propagation with Adaptive moment estimation (Adam) [8] is used to update the weights. To complete the setup of the training procedure, the number of epochs was set to 15 and the batch size to 64, which seemed reasonable values as they were yielding good experimental results. Callbacks such as Early-stopping and Reduce Loss Rate on Plateau were incorporated in the solution to optimize training efficiency by preventing overfitting and dynamically adjusting the learning rate when the model's performance stagnates, ensuring better convergence and improved generalization. The second training phase involved **fine-tuning**. The ConvNextXLarge layers were unfrozen and set as trainable to fit the specific problem. All the convolutional and depth-wise convolutional layers but the first 124 of them, were unfrozen and trained. Training procedure followed the same settings as the first training.

The final result was finally evaluated on the test set, measuring accuracy, precision, recall, f1 score and plotting a confusion matrix to examine the outcome.

4 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 [2] was therefore employed to detect outlier data points and remove them, while various augmentation techniques were devised to tailor the production of augmented data to the needs of the network. See figure 1.

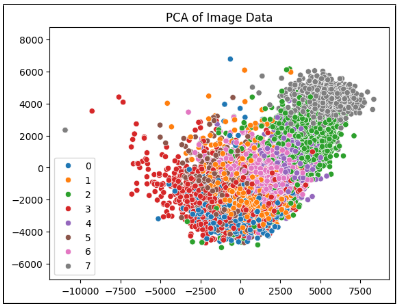

Figure 1: Plot showing PCA ditribution

The points located farther from the origin are identified as outliers. 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. See figure 2

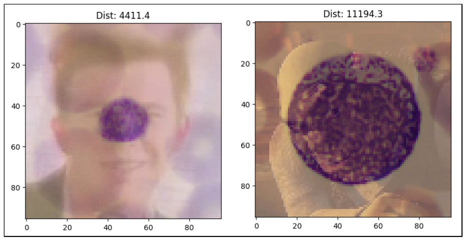

Figure 2: Memes found in dataset

After establishing a clean data set, transfer learning was implemented using VGG [6], ConvNextBase [4], ConvNextXLarge[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. All the results are presented in Table 1 with their corresponding settings.

From these results it can be concluded that using transfer learning leads to better performances. Although ConvNext models outperformed VGG, there is no clear winner between the ConvNext models. It was also proven that by carefully balancing class weights and fine-tuning key hyperparameters (such as the number of epochs, batch size, patience, ...) it is 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.

MORE DISCUSSION HERE AFTER XLARGE COMES and block train comes

5 Results

The final model presented in this paper achieved an overall **accuracy of 85%** 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.

add image of confusion matrix of the best model

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.
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. [10]

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.

				Local Results				CodaBench
Model	**Data**	**Augmentation**	**Fine-tuning**	**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					
Block ConvNextXlarge	Clean	Yes	Yes					

Table 1: Summary of the models' performances

[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. Fouriermix: 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.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.