

AN2DL - First Homework Report

NeuralDropouts

Pinar Erbil, Sergio Pardo, Angela Remolina, Matteo Sissa

perbil, sergiopardo, angelaremolina, matteosissa

Matricola1, 10985243, Matricola3, 247064

November 20, 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. Additionally, technologies like deep learning and deep neural networks have 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 represent one among 8 possible classes of blood cells, respectively Basophil,

Eosinophil, Erythroblast, Immature granulocytes, Lymphocyte, Monocyte, Neutrophil, Platelet.

By observing the dataset and acquiring some knowledge on the domain of the problem (blood cell classification), it was clear that the task was complex because of multiple factors also coming from the field of application. For instance, intra-class variability is not always high for all classes. This means some cells belonging to different classes might appear very similar under the microscope, making it challenging to differentiate them, even for experts..

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 experimentations were therefore conducted without any augmentation to verify the system response and performance.

The correctness of the labels provided in the original dataset, presumably derived from expert knowledge, was assumed throughout the entirety of the project.

3 Method

To address the classification problem, the following methodology was employed. First off, the dataset was inspected using Principal Component Analysis (PCA) to identify and remove outliers and erroneous data points.

The cleaned data set was preprocessed by first split-

ting it into training set (90%), validation set (5%) and test set (5%) and then applying augmentation techniques to increment the number of samples. 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 transformations applied on the original data were designed to mimic common variations that can result from image acquisition at a microscope (like color degeneration or random saturation).

For training purposes, the initial labels in the dataset were transformed into one-hot encoding vectors.

The augmented dataset was then used for training the model. The chosen architecture leverages transfer learning techniques. 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 [2], trained on the ImageNet dataset, 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.

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 the classes. The softmax function is used to normalise the output layer to a probability distribution over the classes. Dropout and batch normalisation techniques are also applied in these last layers to make the model faster to train and more robust. This completes the description of the model design.

As for the training process, it was split into two parts. Initially, the weights of the feature extraction network (ConvNextXLarge) were frozen to ensure stability and allow the training of the classifier net-

work at the output of the model. The loss function adopted is categorical cross-entropy and backpropagation with Adaptive moment estimation (Adam) is used to update the weights. To complete the setup of the training procedure, the number of epochs was set to 10 and the batch size to 64, which seemed reasonable values as they were yielding good experimental results. Early-stopping was also incorporated in the solution, to prevent overfitting.

The second training phase involved fine tuning the ConvNextXLarge layers to fit the specific problem at hand as well as possible. All the convolutional and depth-wise convolutional layers from ConvNextXLarge were unfrozen and set as trainable, apart from the ones belonging to the very first 124 layers of the network. The second training procedure was run with the same options and techniques as the first one.

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 started from 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. Indeed the initial accuracy outcomes were below 20%. Nevertheless, these attempts highlighted some of the most important aspects to be addressed, first of all the need to better inspect the data and also to have a larger amount of samples to control overfitting. PCA [1] was therefore employed to detect erroneous data points and remove them, while various augmentation techniques were devised to tailor the production of augmented data to the needs of the network. In particular, the transformations applied on the images were selected to mimic the common variations and characteristics of biomedical microscopy imaging.

Even with the augmented and clean dataset, the initial basic model was still performing very poorly, with an accuracy below 20%.

As a possible solution, transfer learning was considered, to enhance the recognition abilities of the model especially on subtle differences, which were

probably the cause of the bad performance yielded by a network with a few layers, not really capable of detecting minimal differences. VGG and ConvNextXLarge were identified as two eligible options. They were both 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. Initially, to verify whether using transfer learning could lead to better performances, both models were trained just with the cleaned dataset (without any augmentation) and without fine tuning. The results were much better than the previous ones, as the accuracy metric resulted (??) for the VGG network and 74% for the ConvNextXLarge one. There was still space for improvement, but this experiment showed that employing transfer learning with ConvNextXLarge could lead to a very sound model.

Putting together the pieces, it was clear that to enhance the capabilities of the model further, it was necessary to use more data for training and then fine tuning the model correctly. So the augmentation step of the dataset was thoroughly redesigned, applying more advanced techniques like FourierMix [3], CutMix [4] and MixUp [5]. The training phase for the model was further split into two parts, to accommodate for the fine tuning process at the end, as explained in the Method section.

The result was better than the previous attempts as expected, reaching 84% accuracy. This table summarises the accuracy metrics achieved by the models described in this section, going from top to bottom in chronological order:

5 Results

The final model presented in this paper achieved an overall **accuracy of 84%** 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.

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. [5]

References

- [1] I. T. Jolliffe and J. Cadima. *Principal Component Analysis*. Springer, 2016.
- [2] 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.
- [3] Y. Rong, L. Zhang, and C. Liu. Fouriermix: Resolving dataset bias via fourier data augmentation. *arXiv preprint arXiv:2106.01287*, 2021.
- [4] 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.
- [5] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

Table 1: Summary of models accuracy

Model	Accuracy	Precision	Recall	ROC AUC
Base model (unclean data)	72.20 ± 3.06	94.95 ± 0.52	86.95 ± 0.55	80.16 ± 0.81
Base model (clean data + augmentation)	27.71 ± 3.19	75.70 ± 1.07	55.75 ± 2.16	36.60 ± 1.26
VGG (clean data)	89.24 ± 2.38	95.54 ± 0.49	93.43 ± 1.30	91.68 ± 0.71
ConvNextXLarge (clean data)	89.24 ± 2.38	95.54 ± 0.49	93.43 ± 1.30	91.68 ± 0.71
ConvNextXLarge (clean data, augmentation, fine tuning)	89.24 ± 2.38	95.54 ± 0.49	93.43 ± 1.30	91.68 ± 0.71