POLITECNICO
MILANO 1863

AN2DL - First Homework Report
Harry Plotter

Tommaso Giordano, Sara Iovenitti, Alessandro Maddalena, Luca Olivieri

tommasogiordano, saraiovenitti, alemad01, lucaolivieri,

249075, 257696, 258680, 247455,

November 24, 2024

1 Introduction

In this homework, we tackled a computer vision challenge focused on developing and optimizing deep learning models for the automatic classification of blood cells into eight distinct classes, starting from a dataset of images.

This report covers the methods, experiments, results, challenges, and future improvements; and reflects the outcome of extensive teamwork, with every member equally contributing with commitment and ideas in order to face the challenge all together.

2 Problem Analysis

Our primary objective was to optimize the model's classification accuracy while striking a balance between computational efficiency, model complexity and generalization capabilities.

The provided dataset consisted of RGB images of size 96×96, representing eight distinct classes of blood cells. A key challenge was the **class imbalance**, which could bias the model's predictions towards over-represented classes.

Additionally, the dataset required **cleaning** to remove anomalies, ensuring improved performance and stability.

3 Methods

Among the two top-performing models, achieving the same score and sharing similar characteristics, Tommaso's was selected over Alessandro's due to its more comprehensive documentation and evaluation.

3.1 Pre-processing

Prior to the model selection and training, a data exploration and processing phase has been conducted in order to improve the quality of the learning.

In detail, balanced **class weighting** and **oversampling** techniques have been shown to bring little to no improvement over the raw imbalanced dataset.

t-SNE was performed on pre-trained ConvNeXt-XLarge GAP layer outputs to visualize embeddings. On top of this, we performed manual inspection to remove **anomalies**, namely 800 cell images superimposed with Shrek and 200 cell images superimposed with Rick Astley. All other data points were arranged in clusters and, consequently, were not considered outliers. 8 duplicates were dropped.

3.2 Model and Training

To favor robustness and performance capabilities, we did not train models from scratch, but leveraged models pre-trained on complex and heterogeneous problems as FENs (Feature Extractor Networks)

for their generalization ability. This approach saves time and resources while benefiting from features learned on a large dataset.

In more detail, we deployed ConvNeXt-XLarge model, a modern CNN architecture [4], trained on ImageNet. On top of it, the original classifier is replaced with a custom one, trained for 15 epochs to reason on visual features to effectively classify input images (**transfer learning**). Lastly, all layers have been trained for 10 epochs to achieve optimal performance (**fine-tuning**). Lion was adopted as optimizer for its improved stability, generalization and convergence capabilities [3]; and learning rates were exponentially decayed and cut by half in plateaus.

3.3 Data Augmentation

We applied data augmentation to enhance the diversity of the training set, helping the model generalize better and reducing the risk of over-fitting.

We mainly used geometric and photometric transformations, such as random translations, saturation, contrast and visual occlusion, as well as some more sophisticated approaches. More on this in Sub-section 4.1.

4 Experiments

In this section, we provide an overview of the experiments conducted to assess the performance of the proposed model, along with their strengths and limitations.

4.1 Transformations

We initially trained a baseline model without augmentation. Accuracies were solid in training ($\sim$0.99) and validation ($\sim$0.97), but dropped on test set 1 ($\sim$0.68).

To address this issue, we implemented data augmentation strategies to increase the model's robustness. After testing several options, we found that the KerasCV ***RandAugment*** layer, implementing a set of transformations of various types shown to work well in many scenarios [1], allowed our model to achieve significantly better test results (>0.78).

We further experimented with various augmentation techniques, leveraging several kinds of transformations. These were arranged in sequences and applied in varying orders and probabilities. Initially, we opted for offline augmentation, but it was dropped for storage and memory issues in favor of online application. **TTA** was tested as well, but failed to increase performance.

4.2 Classifiers

Both depth and complexity of the final layers were subject to experimentation.

Single layer, multiple layers classifier, dropout layers, batch normalization layers, as well as ReLU and Swish activation functions were tested.

Our tests showed that a single layer is enough to score a good accuracy (more on this in Sub-section 5.1); however, a deeper one definitely allowed to achieve top performance. Besides this, the other elements were not shown to be critical to the network's performance.

4.3 FEN Models

The network used to extract feature is of paramount importance and allowed us to scale upwards the model's capabilities. We approached the FEN selection problem starting from simpler, smaller networks to more complex and deeper ones.

Model families such as ResNet, MobileNet and EfficientNet were considered, but, ultimately, ConvNeXt showed to deliver greater performance while keeping over-fitting under control. Surprisingly, the largest networks from this family led to a negligible improvement compared to smaller ones.

4.4 Training

Models were transfer-learned and fine-tuned for a range of 10-50 both, depending on the network's potency and ability to converge. We adopted AdamW and Lion as optimizers (delivering roughly equivalent results) with default parameters and with learning rates ranging from 10^{-3} to 10^{-7}, decreasing along with epochs and plateaus, the batch size ranging from 8 to 256, and early stopping. Moreover, during fine-tuning, different configurations of layer freezing and unfreezing were deployed.

5 Results

The selected model, described in Section 3, scored surprising results regarding performance and ro-

bustness. The metrics are as follows:

	Acc.	Prec.	Rec.	F1
Train.	0.9988	0.9988	0.9988	0.9988
Val.	0.9972	0.9972	0.9972	0.9972
Test 1	0.97	-	-	-
Test 2	0.97	-	-	-

The model scores excellent results on public data, with minimal over-fitting, being it able to generalize optimally on test sets. Augmentations have been shown to be very successful in further reducing the over-fitting of the baseline model, closing the train-val gap from ∼0.99-0.97 to ∼0.99-0.99.

5.1 FEN Fine-Tuning Analysis

Further analysis underlined the importance of the FEN on the model's prediction. The two following side-by-side figures visualize the embeddings computed from 5000 data points, each colored by its label, by means of the **t-SNE** method.

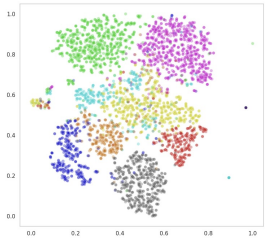

Figure 1: Before FT

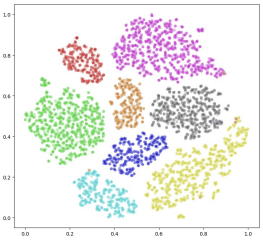

Figure 2: After FT

The pre-trained FEN alone is able to decently separate labels in clusters, and a properly trained, simple classifier can comfortably handle the rest of the reasoning. However, by means of fine-tuning, the FEN progressively learned which visual features to look for and succeeded in outputting richer latent representations. This results in a much clearer separation between clusters, and proves that a complex classification head is not needed to achieve optimal performance.

6 Discussion

The proposed solution demonstrates strong accuracy and robustness, achieving a test accuracy of 0.97; however, depending on the context, it may still be insufficient for certain applications requiring higher precision. Additionally, increasing model size or classifier complexity yielded diminishing returns, indicating limited scalability.

6.1 Interpretability (Grad-CAM)

We used **Grad-CAM** [2] to visualize and interpret the model's decision-making process, helping to identify the regions of the input images that influenced the most its predictions.

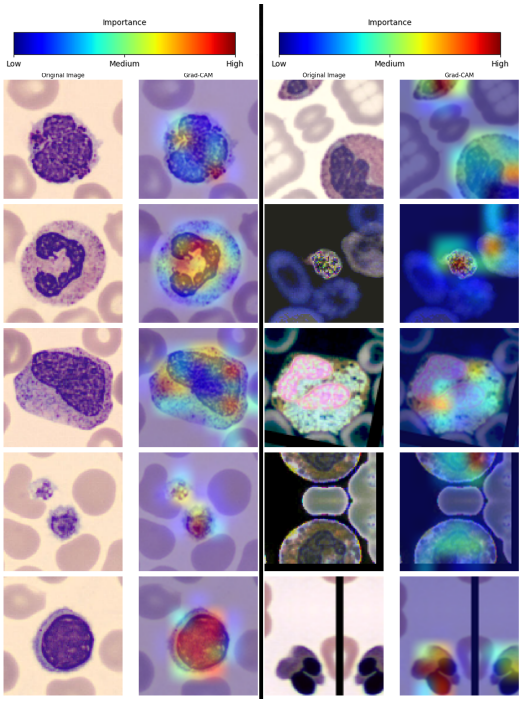

Figure 3: Model attention for normal and augmented images

Heatmaps visualize the predicted class neuron's gradient w.r.t the last 7×7 convolutional filter, averaged along the channel dimension. We see that the model consistently attends to insightful regions of the image, even under challenging augmentations, showcasing its perceptiveness in feature-based reasoning.

7 Conclusions

We developed a deep learning model for blood cell classification using state-of-the-art techniques and data augmentation. The model performed excellently with fine-tuning, achieving strong generalization as well. Future work could focus on refining the dataset and augmentation techniques for even better accuracy.

References

[1] E. C. et al. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv*, 2019.

[2] R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. *arXiv*, 2016.

[3] X. C. et al. Symbolic discovery of optimization algorithms. *arXiv*, 2023.

[4] Z. L. et al. A convnet for the 2020s. *arXiv*, 2022.