AN2DL - First Homework Report
OverfittingExorcists

Daniele Laganà, Marcello Martini, Gianluigi Palmisano, Samuele Pozzani

danielelagana, marcellomartini, gianlupalmisano, samuelepozzani

252548, 243880, 244774, 242694

November 22, 2024

1 Introduction

This project focuses on *image classification* using **deep learning** techniques. The goal is to maximize the accuracy of the model to classify blood cells divided in eight classes.

2 Problem Analysis

The main challenges in this problem include:

- **Class Imbalance:** Certain blood cell types may be underrepresented, leading to potential biases in model predictions.
- **Intra-class Variability:** Significant variations within the same class due to differing staining techniques, cell orientations, and overlapping cells.
- **Inter-class Similarity:** Morphologically similar cell types can be difficult to distinguish, especially at lower resolutions.
- **Noise and Artifacts:** Presence of noise, such as blurred images and artifacts, can hinder feature extraction.

3 Method

3.1 Data exploration

The dataset comprises 13,759 RGB images, each with a resolution of 96x96 pixels. Each of the eight classes represents a specific blood cell type. The dataset is highly imbalanced, with certain classes significantly underrepresented compared to others.

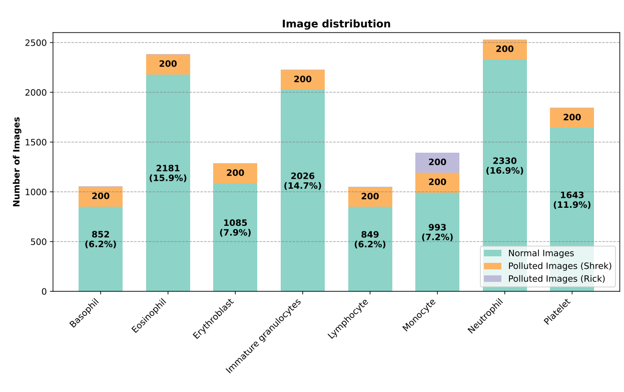

Figure 1: Distributions of the samples in the dataset and the polluted images: The dataset contains 1,600 images with a Shrek Saxophone overlay and 200 images with a Rick Astley overlay.

3.2 Data cleaning

The provided dataset is polluted with memes, with a distribution illustrated in Figure 1. For this reason, it has been cleaned by removing all the identical images containing either Shrek or Rick Astley.

3.3 Data preprocessing

The cleaned dataset is split into **training**, **validation**, and **test** sets while maintaining the class distribution (using **stratified sampling**). Finally, the labels for all three sets are converted into **one-hot encoded** format for use in a classification model with eight output classes.

To tackle class imbalance, **class weights** are calculated and applied during training, giving greater importance to minority classes. This ensures balanced loss contributions, improving the model's performance on underrepresented categories.

3.4 Data augmentation

Data augmentation is employed to enhance the diversity of the training data by applying various transformations to the images before feeding them into the network. This technique helps the model generalize better by reducing overfitting, improving robustness to variations, and enabling it to learn more discriminative features, ultimately enhancing its performance on unseen data.

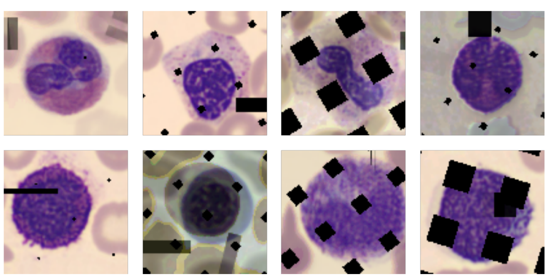

Figure 2: Example of the augmentations performed during training phase.

3.5 Tansfer learning

Transfer learning is utilized with a convolutional neural network pre-trained on **ImageNet** [1], to leverage its ability to extract rich and generalizable features. All layers are frozen to preserve the pre-trained weights, preventing overfitting on the relatively small dataset. This approach significantly reduces training time while benefiting from the robust feature representations of the model, leading to improved performance on the classification task.

3.6 Fine tuning

Fine-tuning optimizes model performance by freezing early layers to preserve low-level features while training specific layers to adapt pre-trained features to the blood cell dataset. This balances pre-trained knowledge with task-specific optimization, improving accuracy and reducing overfitting.

3.7 Test Time Augmentation

Test-time augmentation (TTA) improves the model's robustness by applying various transformations, such as flips, rotations, and brightness adjustments, to test images. The model predicts each augmented version, and the results are aggregated using **weighted averaging**. This approach enhances generalization, ensuring reliable performance across diverse input conditions.

3.8 Ensemble prediction

Ensemble prediction combines the outputs of multiple models to improve accuracy and reliability. In this work, two models, trained with distinct data augmentation strategies, are used to predict the label of an image.

4 Experiments

4.1 Pre-trained models

Multiple pre-trained architectures have been experimented, including **MobileNet** [2], **EfficientNet** [3], and **VGG19** [4], to identify the most effective backbone for the classification task. Custom layers are added on top, including **batch normalization**, **dropout**, and **dense** layers (256 neurons) with **L2 Ridge Normalization** performed, to adapt the extracted features to the classification task. Each model has been fine-tuned on the dataset to leverage their transfer learning capabilities. Among these, **EfficientNetV2S** achieves the highest accuracy, outperforming the others due to its efficient scaling and superior feature extraction capabilities. The specific results for each model are shown in Table 1.

4.2 Augmentation strategies

In addition to standard augmentation techniques, advanced algorithms are used probabilistically to

Model	Parameters	Accuracy	Precision	Recall	ROC AUC	Codabench
EfficientNetB4	18,142,055	98.41%	98.42%	98.41%	99.96%	0.80
MobileNetV3S	1,092,216	96.20%	96.22%	96.20%	99.88%	0.46
VGG19	20,158,792	98.28%	98.33%	98.28%	99.92%	0.47
EfficientNetV2S (A)	20,666,472	98.66%	98.67%	98.66%	99.99%	0.92
EfficientNetV2S (B)	20,666,472	98.87%	98.90%	98.87%	99.97%	0.92
Ensemble model	**-**	**99.04%**	**99.05%**	**99.04%**	**-**	**0.93**

Table 1: Performance metrics computed using the test dataset. The last column shows the Codabench score calculated during the development phase. **EfficientNetV2S** stands out. Model (A) is trained using a mixed augmentation pipeline, while model (B) is augmented using RandAugment only.

introduce occlusions, distortions, and structured noise. These include **RandomCutout**, which masks random portions of an image to encourage robustness to occlusions; **RandAugment** [5], which applies a randomized combination of transformations to increase data diversity; **AugMix** [6], which blends multiple augmentations while preserving image semantics; and **GridMask** [7], which overlays grid-like masks to improve spatial awareness. An example of augmented samples is shown in Figure 2.

4.3 Training settings

The training settings address dataset imbalance using **class weights** and **Focal Loss** [8] where $\gamma = 2$. The **Adam optimizer** [9] is selected, with smaller learning rates applied during the fine-tuning phase compared to transfer learning. A batch size of 32 is used to balance memory usage and gradient estimation accuracy. **Learning rate decay** and **early stopping** callbacks are implemented to improve training efficiency. Additionally, **TensorFlow data input pipeline**[1] with auto-tuning is utilized to optimize GPU performance.

5 Discussion

MobileNetV3S exhibits suboptimal performance, likely due to its limited capacity for feature extraction on this dataset. Data augmentation plays a critical role in improving generalization, but it must be carefully designed to avoid introducing unnecessary noise. Class balancing through oversampling proves less effective than using class weights in addressing dataset imbalance. The most challenging class to predict is Class 6 (Neutrophil), likely due to its inter-class similarities and variability. It is suspected that the test set used for Codabench scoring may contain a higher proportion of underrepresented classes, influencing the observed accuracy.

6 Conclusions

The implemented strategies proved highly effective for the blood cell classification task, incorporating data augmentation, class weighting, Focal Loss, transfer learning, and fine-tuning. **EfficientNetV2S** achieves top performance with two different augmentation pipelines as explained in Table 1. The final predictions are generated by averaging the outputs of the two models (A) and (B) after applying TTA, resulting in a robust and reliable ensemble. Ultimately, the performance validated on the test set is presented by the confusion matrix in Figure 3.

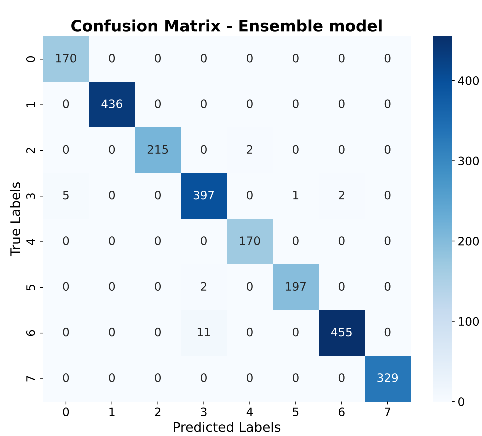

Figure 3: Confusion Matrix of the Ensemble model with TTA.

[1]TensorFlow Data documentation

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[2] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[3] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training, 2021.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[6] 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, 2020.

[7] Pengguang Chen, Shu Liu, Hengshuang Zhao, Xingquan Wang, and Jiaya Jia. Gridmask data augmentation, 2024.

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.