AN2DL - First Homework Report
Team Name

Firas Mtibaa, Karam Khammel, Sarra Mars, Mohamed Abdelhamid Kenani

fmtibaa, chakala, sarramars, mkenani

242795, 276412, 276424, 276413

November 24, 2024

1 Introduction

The use of deep learning techniques in medical imaging has become increasingly prevalent for automating image analysis tasks. Our objective is to enhance the accuracy of identifying different types of blood cells. This project mainly revolves around the creation of *deep learning* models designed to classify **96x96 RGB images of blood cells**. These images are categorized into 8 classes, each representing a different cell state. The task is framed as a *multi-class classification* problem.

2 Problem Analysis

To address the key challenges of this classification problem and to better understand it, we began with an analysis focusing on three main aspects.

2.1 Dataset characteristics

The initial step involves loading and visualizing the data to gain an understanding of its usage. The dataset consists of images spanning eight classes, with pixel intensity values ranging from 0 to 255. Through visualization of the dataset, we observed class imbalance. The following table 1 presents the composition of the dataset:

Table 1: Class distribution

Basophil	1052
Eosinophil	2381
Erythroblast	1285
Immature granulocytes	2226
Lymphocyte	1049
Monocyte	1393
Neutrophil	2530
Platelet	1843

2.2 Main challenges

The primary challenge lies in the dataset's limited variability, which increases the risk of overfitting especially in the presence of complex class distributions. Additionally, the slight imbalance in the dataset could impact the model's performance, potentially leading to bias, as some classes are overrepresented compared to others.

2.3 Initial assumptions

As initial assumptions, we considered the dataset to be correctly labelled, with duplicates and outliers to be removed during preprocessing. In addition, to address potential challenges, we assumed that data augmentation techniques would effectively mitigate the risk of overfitting. By artificially expanding the dataset's variability, we expected to enhance the model's ability to generalize.

3 Method

This section outlines the steps we took to tackle the classification problem, starting with data preprocessing and augmentation techniques, followed by building our models.

3.1 Preprocessing

This step is crucial as it prepares the data for training. This naturally involves cleaning the data by removing duplicates and identifying outliers based on deviations in pixel intensity mean and standard deviation. An image was considered a non-outlier if it satisfied the following conditions:

$$|\mu_i - \bar{\mu}_c| \leq \alpha \sigma_{\mu_c} \tag{1}$$

$$|\sigma_i - \bar{\sigma}_c| \leq \alpha \sigma_{\sigma_c} \tag{2}$$

where μ_i and σ_i are the mean and standard deviation of image i, $\bar{\mu}_c$ and σ_{μ_c} are the mean and standard deviation

of μ_i over all images in class c, $\bar{\sigma}_c$ and σ_{σ_c} are the mean and standard deviation of σ_i over all images in class c, and α is the threshold.

As a first intuition, we considered background removal as it involves the isolation of the blood cell by converting to grayscale, applying gaussian blur, and creating a binary threshold mask. This approach turned out to not be effective on the output of the models as expected.

We then considered data augmentation, as mentioned above, to enhance the diversity and robustness of our dataset. Specifically, we used **AugMix** [4] and **RandAugment** [7] to generate synthetic images, which will help improve model generalization. **AugMix** blends multiple augmented versions of an image to create a composite image, as shown in equation 3. **RandAugment** randomly selects and applies a fixed number of operations with a constant magnitude, as shown in equation 4 below.

$$x_{aug} = \lambda_0 \cdot x + \sum_{i=1}^{N} \lambda_i \cdot x_i \tag{3}$$

where N is the number of chains, x_i augmented images from different chains, and λ_i is a coefficient sampled from a Dirichlet distribution.

$$x_{aug} = A_k \circ A_{k-1} \circ \cdots \circ A_1(x) \tag{4}$$

where A_k are the augmentation operations selected at random.

We created two separate datasets using these two augmentation techniques and applied additional gaussian noise with a standard deviation of around 20%. These techniques improve the model's robustness to shifts and corruptions in input data.

3.2 Parameters and tools

We trained our models using a batch size of 32 and learning rates ranging between 10^{-3} and 10^{-4}. We chose the *categorical crossentropy* loss function, as shown in equation 5 and the *Adam* (Adaptive Moment Estimation) optimzer [6]

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{5}$$

N is the number of classes, y_i is the true label probability for class i, and $\hat{y}_i$ is the predicted label probability for class i.

3.3 Exploring models

3.3.1 Convolutional Neural Networks (CNN)

We created a custom CNN model inspired by [1]. The architecture of this model includes:

- 6 convolutional layers with increasing filters from 32 to 512, kernel size 3×3, stride 1, ReLU activation, and max-pooling 2×2 to extract hierarchical features from the images.
- Batch normalization: applied after each convolution layer.
- Dropout layer: 0.25
- 5 fully connected layers with 128 neurons, ReLU activation, and L2 regularization.
- Output layer with *SoftMax* activation for multi-class classification, as shown in equation 6.

$$\sigma_i(z) = \frac{e_i^z}{\sum_{k=1}^{p} e^{z_k}} \tag{6}$$

3.3.2 Transfer Learning

In this modeling step, we explored the usage of transfer learning. We picked models such as *ResNet50* [8], *InceptionV3* [3], *EfficientNetB0* [9] and *NASNetMobile* [2], pre-trained on the **ImageNet** [5] dataset, to assess their performance on our classification problem. The architecture of our models are shown in figure 1

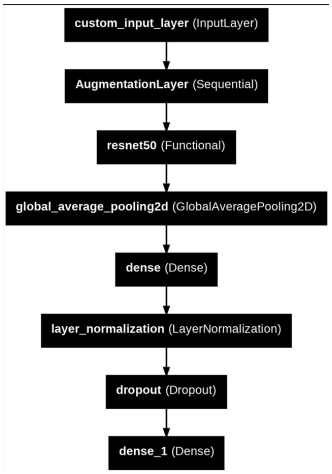

Figure 1: Model architecture

3.3.3 Ensemble learning with Model stacking

In order to enhance the performance and robustness of our blood cell classification system, we implemented an ensemble learning strategy by combining the predictions of our three best-performing models. This approach, known as *model stacking*, involves training a meta-classifier that learns to predict the correct class based on the outputs of the base models. We chose a simple model using three fully connected layers and an output layer.

Table 2: Accuracy and F1-score results on different models.

Model	Train accuracy	Validation accuracy	Train F1-score	Validation F1-score
ResNet50	89.75	**98.56**	88.05	98.41
Custom Model	90.45	92.35	89.23	93.14
NASNet	89.23	97.88	88.15	97.72
EfficientNet	83.46	95.58	81.44	94.92
InceptionV3	98.28	97.96	97.89	97.12
Stacked Model	95.72	**98.56**	95.20	**98.50**

3.3.4 Visualizing model attention with Grad-Cam

To gain insights into our convolutional neural network's decision-making process for blood cell classification, we employed *Gradient-weighted Class Activation Mapping* (**Grad-CAM**) [10]. This technique highlights regions in the input image that are important for the model's predictions, thereby enhancing interpretability.

4 Results

In this section, we will discuss the results of our models. Table 2 groups the accuracy and F1 scores on the training and the validation datasets.

Compared to simple CNN models, all of our models performed better. The models we trained managed to get very high accuracy and F1 scores on the validation dataset. But we have to point out lower scores during training. This is most likely due to the degradation of data due to augmentation and noise. We can highlight that the Ensemble model gave the best overall performance, both locally and on the hidden test set.

Using Grad-CAM, we try to understand the predictions of the models. Figure 2 represents the attention zones.

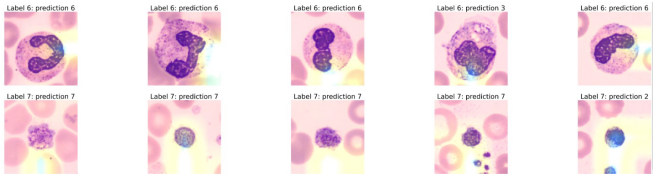

Figure 2: Grad-CAM outputs

Even though the evaluation metrics are high, we noticed through the highlighted parts that the model doesn't always focus on the center of the image, and this can explain some of the misclassified predictions.

5 Discussion

Despite achieving good results on our local validation set, our model's performance on the hidden test set was only average. This suggests that certain factors may have limited its generalization capability. One of the possible reasons for the performance drop could be an excessive data augmentation that distorted the images. This could have led the model to learn from unnatural artifacts rather than meaningful features, reducing its effectiveness on real-world data. Besides, visualization using Grad-CAM showed that the model sometimes did not focus on the central regions of the images where the blood cells are located. The model might be paying attention to irrelevant background areas, which doesn't generalize well to new images.

6 Conclusion

This project tackled the classification of blood cell images using deep learning. Our main contributions include using augmentation techniques, exploring transfer learning, and achieving robust performance with an ensemble stacking approach. Despite strong validation results, the models showed reduced performance on unseen test data. Future work could explore background removal using segmentation models to better isolate relevant features. Additionally, ensuring that augmentation techniques preserve key image characteristics without distortion could further enhance model generalization and robustness.

References

[1] R. Asghar, S. Kumar, P. Hynds, and A. Mahfooz. Classification of all blood cell images using ml and dl models. *arXiv preprint arXiv:2308.06300*, 2023. Submitted on 11 Aug 2023 (v1), last revised 7 Oct 2024 (this version, v3).

[2] J. S. Q. V. L. Barret Zoph, Vijay Vasudevan. Learning transferable architectures for scalable image recognition. *arXiv:1707.07012*, 2017.

[3] S. I. J. S. Z. W. Christian Szegedy, Vincent Vanhoucke. Rethinking the inception architecture for computer vision. *arXiv:1512.00567*, 2015.

[4] E. D. C. B. Z. J. G. B. L. Dan Hendrycks, Norman Mu. Augmix: A simple data processing

method to improve robustness and uncertainty. *arXiv:1912.02781*, 2019.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[6] J. B. Diederik P. Kingma. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[7] J. S. Q. V. L. Ekin D. Cubuk, Barret Zoph. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv:1909.13719*, 2019.

[8] S. R. J. S. Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

[9] Q. V. L. Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv:1905.11946*, 2019.

[10] A. D. R. V. D. P. D. B. Ramprasaath R. Selvaraju, Michael Cogswell. Grad-cam: Visual explanations from deep networks via gradient-based localization. *arXiv:1610.02391*, 2016.