POLITECNICO
MILANO 1863

AN2DL - First Homework Report
Hendrix

Julie Ronesen Landaas, Ole Thorrud, Trygve Myrland Tafjord, August 11, 2025

1 Introduction

The following is a report on the development of a CNN (*Convolutional Neural Network*) designed to classify images of eight different types of blood cells. The groups primary goals for this project are to develop a **high-performing model** and to gain insights into **optimizing workflow efficiency** when working with convolutional neural networks. The approach to achieve these goals consist of analyzing the dataset, doing research on how similar problems have been solved before and testing the concepts presented in lectures.

2 Problem Analysis

Upon the primary review of the dataset, the following elements were taken into consideration throughout the work on the project:

1. The dataset contains 1806 duplicated images uniformly distributed among the eight classes. Most of these duplicates are of the same two illegitimate images.

2. The dataset contains 321 images with a notable blue color cast.

3. The dataset is unbalanced, containing a much larger amount of images corresponding to certain labels than others.

The main challenge of this project is to develop a robust model that generalizes well. This despite the dataset's significant imbalance and the subtle visual differences between classes. The groups initial assumption was that focusing on pre-processing and augmentation will be essential for developing a high performing model.

3 Method

Our approach was the following. Firstly, the group removed the duplicated and illegitimate images. Furthermore, the images with a blue color cast were treated as outliers and removed from the data. To address the dataset imbalance, the group tried two main strategies: first, we would apply selective data augmentation to increase the representation of underrepresented classes by creating varied versions of certain images; secondly, we would incorporate class weights to balance the influence of each class during training.

After doing initial research on how similar problems has been solved [1], the group started by making a baseline model. This included a *preprocessing* part consisting of splitting the training set into training, validation and test sets. Then all images was augmented using the *RandAugment* layer from KerasCV, see the values in Table 1. To increase the volume of the training set, all training images were rotated by 90 degrees three times, and each of these were then flipped about the y-axis. As a result, each original image was expanded into a set of eight distinct augmented images resulting in a baseline training set with **76536** distinct images. The model design leveraged *transfer learning*, and three models trained on the same dataset with the same classification head were compared. The

results from this screening can be seen in the rows with bold writing in Table 1. The best performer, and therefore the groups choice for a *baseline model*, was **ResNet50** with a single dense layer and dropout with a probability of 0.3 as the classification head. This gave the so far highest validation score of 0.88, and a corresponding test-score of 0.55.

4 Experiments

The group decided on a set of experiments to increase model performance. Considering the large difference between the training error and test error, we decided to primarily focus on reducing overfitting. The experiments were based on tuning the model in the following ways:

1) Add batch normalization. Add batch normalization to improve stability and convergence.

2) Increase intensity of image augmentation. The augmentation was tuned by using *RandAugment*, and setting the *magnitude* and *augmentations_per_image* attributes.

Test	Magnitude	Augmentations_per_image
1	0.35	1
2	0.45	2
2	0.60	2

3) Use fine-tuning in the convolutional layer. Through fine-tuning, we hoped to adapt the ResNet50 model to the dataset, boosting performance.

4) Increase dataset size through augmentation. The dataset size was increased through a combination of different augmentation levels, creating a new data set consisting of **120244** images.

5) Experiment with classification heads. The number of layers, the amount of neurons in each layer and the dropout rate was tuned.

5 Results and Discussion

5.1 Model Architecture

As seen in Table 1, **ResNet50** performed the best among the three transfer models considered. Furthermore, using a transfer model instead of a custom model gave the largest measured performance gain during the exercise. ResNet50s superior performance among the tested transfer models could be due to several factors. The blood cell dataset contains subtle visual differences between classes, which suggests that a deeper network could be required. This makes ResNet50 suitable as it is a deep network mitigating the risk of vanishing gradient by use of residual connections [2].

5.2 Batch normalization

Unexpectedly, the model performed worse with batch normalization. The model achieved 0.79 accuracy compared to 0.81 without normalization as seen in Table 1. The decrease in performance with batch normalization may be attributed to the small batch size of 32 used during training. Small batch sizes can make batch normalization less stable, as outliers and noise within each mini-batch can disproportionately influence the calculated mean and variance. This instability may lead to inconsistent normalization across batches, ultimately affecting model performance, which is what the group observed.

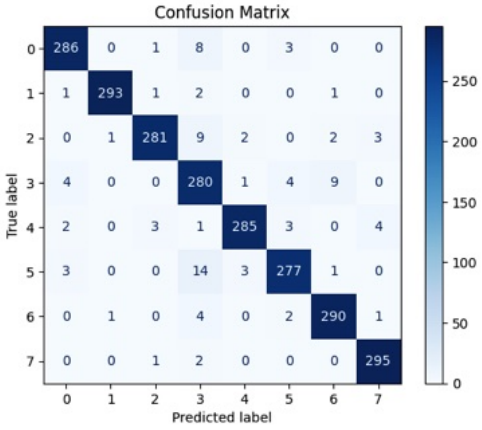

Figure 1: Confusion Matrix

5.3 Dataset imbalance

Our resulting confusion matrix 1 indicate that the model achieved a robust classification performance across all eight classes despite the class imbalance in the training set. This suggests that the model successfully handled the dataset imbalance by using class weights in training.

5.4 Increased image augmentation

By increasing data augmentation, the first significant performance boost was observed. As shown in Table 1, the best results were achieved with a magnitude of **0.45** and **2** augmentations per image. Further increases in augmentation led to de-

Table 1: Model overview with augmentation, fine-tuning, batch normalization, and accuracy values. *: Data set as described in bullet point 4 in section 4

Model	Aug.	Aug. pr. image	Fine-tuning	Batch Norm.	**Accuracy**
ResNet50	0.45	2	Yes	No	0.81
ResNet50	*	*	Yes	No	0.81
ResNet50	0.35	1	Yes	No	0.78
ResNet50	0.6	2	Yes	No	0.80
MobileNet	**0.2**	**1**	**No**	**No**	**0.39**
ResNet50	**0.2**	**1**	**No**	**No**	**0.55**
EfficientNet	**0.2**	**1**	**No**	**No**	**0.49**
Custom-model	0.2	1	-	No	0.28

creased accuracy, suggesting that beyond a certain point, additional augmentation may introduce noise reducing its ability to generalize effectively.

5.5 Increased dataset size

Expanding the dataset from the baseline of 76,535 images did not improve performance as expected. While the larger dataset matched our best model's accuracy, we had anticipated a notable boost from the additional training data. The lack of improvement suggests that the baseline dataset already captured the key features effectively, and that further gains might lie in other areas. Given the negative impact on training time, we decided to abandon the expanded dataset.

5.6 Fine tuning of filter weights

Allowing for fine tuning of the weights in the convolutional layer of our transfer model turned out to be one of the more impactful measures. Fine tuning initially led to poor convergence, however it turned out to be a question of sufficiently decreasing the learning rate. **With a learning rate of 0.00008 the increase in performance from the models with default filter weights is one of approximately 50%**, as seen in the table 1. The fact that the model performed better with fine tuned weights was no surprise to the group as this was expected to further tailor the transfer model to our data set, but the magnitude of this improvement took us by surprise.

5.7 Classification heads

Experimenting with classification head structures proved less effective. Although validation accuracy remained high, test performance declined, suggesting that deeper classification heads led to overfitting. The transfer model's feature extraction was likely strong enough that added complexity offered no benefit.

6 Conclusions

The final model achieved an accuracy of 0.81. While not the best in class, this was deemed a satisfactory result. Future work could explore alternative approaches such as *one-vs-rest* classification or *ensemble learning* with majority voting. Though some initial research and testing were conducted, limited testing opportunities caused these efforts to be put on hold. Further more, the group believe there to be additional performance gain from further tuning the augmentation.

References

[1] M. Abou Ali, F. Dornaika, and I. Arganda-Carreras. White blood cell classification: Convolutional neural network (cnn) and vision transformer (vit) under medical microscope. *Algorithms*, (11), 2023.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 2016.