AN2DL - First Homework Report
Madapenguins

Irene Caridi, Francesca Girolami, Kristina Tas, Mengshuang Tang

irenecaridi, girolafra, kristinatas, Mengshuang

243023, 226700, 249022, 270498

November 22, 2024

1 Introduction

This project focuses on *image classification* using **deep learning** techniques. In particular, the aim of the project is to develop a neural network capable of classifying eight distinct types of blood cells based on provided image data. The dataset includes 13,759 RGB images, each sized at 96x96 pixels, alongside corresponding labels ranging from 0 to 7, which represent the different blood cell categories.

2 Dataset analysis

2.1 Data preprocessing

Reviewing the dataset, it is observed that some images were duplicated, in particular some images are overlapped to two different backgrounds. After removing all duplicates, the presence of *outliers* is assessed using the isolation forest technique and then removed. With these removals the intention is to make the dataset more robust.

The next step involves visualizing the distribution of images across the different classes. It is observed that the **classes are imbalanced**, with the highest class 6 containing 2280 images, while 817 in the lowest, class 0.

Finally, an 80:20 split is chosen for the train-validation partition. For both phases of the challenge, the validation set is used to evaluate the model performance.

2.2 Data augmentation

In order to increase the generalization capability of the model, data augmentation is applied. After several discussions and trials, a dataset is created by applying various levels of transformations, including rotation, flip, zoom, brightness and Gaussian noise.

3 CNN

3.1 Feature extraction layers

To determine the structure of the feature extraction (FE) layers, a first attempt is the implementation of some basic CNN structures. Despite these efforts, the generalization performance remains limited (train and validation accuracy around 94% and only 12% for test set). Consequently, a **transfer learning** approach is adopted, considering several pre-trained architectures. ResNet50, InceptionV3, Xception and InceptionResNetV2 are chosen for the investigation because they are designed to handle complex patterns while minimizing overfitting, making them ideal for the detailed and challenging task of blood cell image classification [1]. They also have a good balance between computational efficiency and accuracy instead of VGG and MobileNet. In all model tested, the pre-trained weights from ImageNet are used.

3.2 Classifier

The architecture of the classifier (table 1) is composed by a global average pooling layer and two fully connected (Dense) layers with L2 regularization, the first with 512 units the second with 256. Each dense layer is followed by dropout layer with a rate of 0.5. Finally, the output layer consists of 8 units with a softmax activation function, which is suitable for multi-class classification.

Layer	Details
GAP	-
Dense	512, ReLu
Dropout	0.5
Dense	256, ReLu
Dropout	0.5
Dense	8, Softmax

Table 1: classifier structure

3.3 Transfer learning result

The model is compiled with Lion optimizer with a learning rate of 0.0001 and sparse categorical cross-entropy loss function used when target labels are integer values. The model tracks also the accuracy level. The used hyperparameters are:

- *Epochs*: 30, few epochs due to the pre-trained knowledge of the model;
- *Batch size*: 64;
- *Dynamic learning rate*: ReduceLROnPlateau monitors validation accuracy and helps the model converge more effectively by slowing down training adjustments when progress plateaus;
- *Early stopping configuration*: callback monitors the minimum validation loss, restore the best weights after stopping and has 15 as patience value, low value due to the pre-trained model's knowledge.

In table 2 are reported the maximum validation accuracy values across all epochs.

Model	Accuracy (%)	Recall (%)	F1 score (%)
ResNet50	**87.32**	**87.32**	**87.33**
Xception	87.36	87.36	87.32
InceptionV3	82.96	82.96	83.02
InceptionResNetV2	86.53	86.53	86.59

Table 2: Performance metrics of transfer learning models.

3.4 Fine-tuning

Weights used in FE layers are retrained with fine-tuning technique. The best model, according to F1 score and confusion matrix, ResNet50, is analyzed from now on: just the last block of the model is unfreezed and this leads to an improvement of 10.86% of F1 score (98.19%).

In this step epochs are increased at 70 and learning rate of the Lion optimizer at 0.00001.

4 Regularization strategies

The dropout layers play a crucial role in regularizing the model, ensuring that it does not rely too heavily on any single feature, ultimately improving its performance on unseen data.

Another strategy to deal with the class imbalancing is assigning weights to each class: each weights are inversely proportional to the number of observations which compose that specific class. During the training of the model, these weights are included to ensure that the contribution of each class to the loss function is proportional to its importance.

Finally, the Lion optimizer, a stochastic-gradient-descent method, a callbacks function and a dynamic learning rate are used for a more stable outcome and overfitting prevention.

5 Results

Testing the model on the dataset described in section 1, the model makes good results on the dataset in section 1; while testing on an external dataset, the accuracy value decreases. In table 3 there are the scores of the final model.

Set	Accuracy (%)
Train	98.28
Validation	98.18
Test	51.00

Table 3: Accuracy of the final model

This is due to the misclassification of some classes, which are more similar to each other.

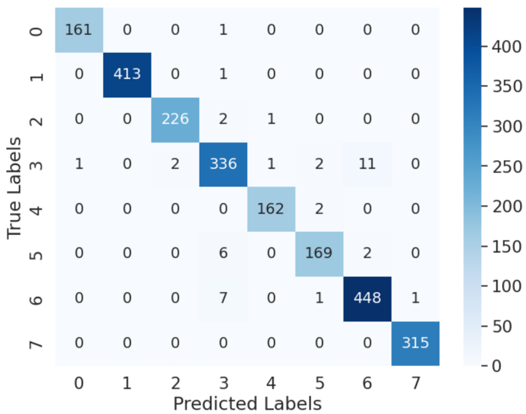

Figure 1: Confusion matrix of the ResNet50 performance on validation set

6 Others attempts

This section collects all attempts done to achieve the final model.

6.1 Data preprocessing

Another possible solution to deal with unbalanced classes is performing downsampling then some data augmentation is made [2].

6.2 MiniCNN

The first CNN is composed by 5 blocks. Each block consists of a convolutional layer with a ReLU activation, and a max-pooling layer. The convolutional layers use GlorotNormal initialization and L2 regularization. After the convolutional layers, the classifier is composed by five fully connected layers with units decreasing from 512 to 32, L2 normalization and ReLU activation. The output layer uses a softmax activation function.

The second CNN is created by concatenating three ResNet blocks increasing step by step the filters number from 32 to 128. The classifier is composed by fully connected layers with dropout.

The last CNN is developed with four inception blocks. Each block processes the input through parallel convolutional paths with differing kernel sizes and combines the outputs using pooling operations. The model is then passed through a series of fully connected layers with progressively smaller.

7 Discussion

As said, our first challenge is the creation of a completely new CNN. Three possibilities are explored: one using a sequential combination of different arbitrary chosen layers, one following the principle of an inception block and the last one using ResNet blocks (as discussed in section 6.2). Despite some trials, these models aren't good enough to achieve a good classification results. As a result, a transfer learning and fine tuning implementation is pursued.

During data augmentation analysis the translation is not included to not modify the geometry of the cell.

8 Conclusions

A neural network model for multiclass classification of eight distinct types of blood cells from microscopy images is successfully implemented. ResNet50 gives good performance with an F1 score of 87.33% after transfer learning, with a further improvement of 10.86% when the last block of the model is retrained.

Other key challenges, such as class imbalance and data preprocessing, are address with some strategies like duplicate removal, outlier detection, data augmentation and class weighting. Additionally, regularization techniques, including dropout layers and dynamic learning rate adjustments to prevent overfitting.

Dealing with these steps, it's observable a good performance of the model on our train-validation set, but a worse performance on the test set: misclassifications highlight the need of further optimization.

Future developments include the exploration of additional techniques, such as advanced augmentation methods, to extend the generalization capability of the model.

References

[1] R. U. Khan, S. Almakdi, M. Alshehri, A. U. Haq, A. Ullah, and R. Kumar. An intelligent neural network model to detect red blood cells for various blood structure classification in microscopic medical images. *Heliyon*, 10(4):e26149, Feb 2024. Published 2024 Feb 13.

[2] Jay M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

Contributions

Irene Caridi: creation of new CNN, literature study, model conceptualization, model development (data inspection, choice of classification layers, hyperparameter tuning), report writing.

Francesca Girolami: creation of new CNN, model conceptualization, model development (data inspection, data augmentation, hyperparameter tuning), report writing.

Kristina Tas: model conceptualization, model development (data inspection, outlier removal), report revision.

Mengshuang Tang: model conceptualization, model development (hyperparameter tuning, data augmentation, dynamic learning rate).