White Blood Cell Microscopy Classifier

M. Alipoor*, G. Annaloro*, M. Tirabassi*

*All authors contributed equally; authors are listed alphabetically.

1 Introduction

This project addresses the classification of white blood cells from microscopic images utilizing deep learning techniques. The objective is to employ pre-trained convolutional neural network (CNN) models for automatic feature extraction, in contrast to traditional machine learning methods that require manual feature engineering. Throughout this report, "F1 score" refers to local test performance, while "Score" represents online test results.

2 Dataset Analysis

The dataset consists of 13,759 images with dimensions 96×96×3, categorized into 8 classes.

2.1 Outlier Removal

An initial analysis with ResNet50 feature vectors visualized using t-SNE revealed class 5 outliers (Shrek images) and a single Rick Roll image affecting all classes. These anomalies, confirmed via visual inspection, were removed by truncating the dataset at index 11,595. This ensured a cleaner dataset and more reliable results.

Figure 1: Outliers: Shrek (left) and Rick Roll (right).

3 Dataset Splitting

The dataset was divided into an unbalanced training-validation set and a balanced test set. The balanced test set was designed under the assumption that, in real-world scenarios, different cell types are equally represented. This approach prevents biases caused by skewed population statistics and ensures that accuracy and F1 score yield identical results on the test set. An unbalanced validation set was used to avoid further amplifying the inherent imbalance in the training data. F1 score was selected as the primary evaluation metric to effectively address class imbalance.

4 Class Imbalance

Two strategies were evaluated to address class imbalance [1]:

1. Generating additional data through transformations [2].
2. Assigning weights to classes during training.

While synthetic data balanced the dataset, class weights yielded better generalization and performance, reflected in higher F1 scores.

Method	F1 Score (%)
Synthetic Data	92.31
Class Weights	**96.78**

4.1 Synthetic Data

Synthetic data creation through traditional augmentation techniques aimed to moderate class imbalance by generating realistic variations of underrepresented classes. Transformations such as flipping, rotation, zooming, brightness, and contrast

adjustments maintained biological plausibility by preserving key structural features such as proportions and shapes.

4.2 Class Weights

During training, higher weights were assigned to underrepresented classes, ensuring proportional contributions to the loss function. This reduced bias toward dominant classes. Despite this approach, a common issue across all models was the misclassification of classes 5 and 6 as class 3, indicating potential challenges in distinguishing features specific to these classes.

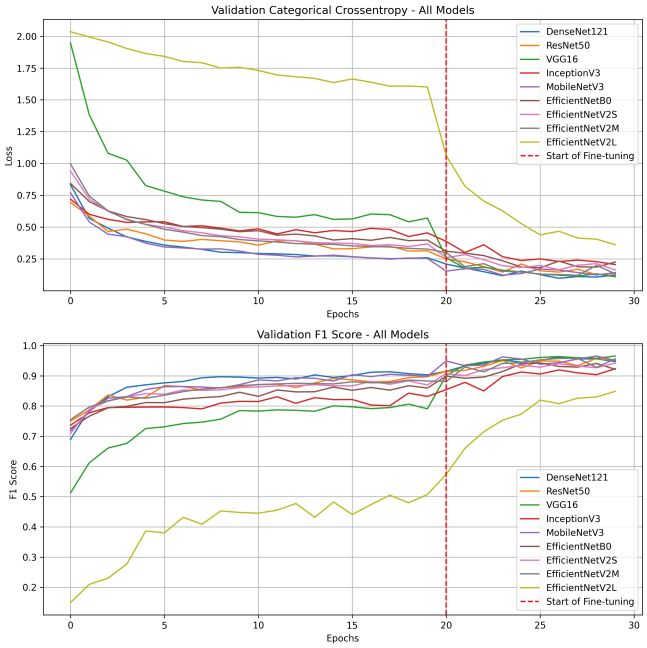

Figure 2: Validation set performance during transfer learning and fine-tuning of base models: Categorical cross-entropy loss (top) and F1 score (bottom).

5 Feature Extractor

Pretrained CNN architectures were evaluated for both their performance and computational efficiency [3]. Among these, **EfficientNetV2M** emerged as the optimal choice, offering an ideal balance between computational requirements and F1 score. All models were compared using a consistent set of hyperparameters to ensure a fair and unbiased evaluation.

Model	F1 Score (%)
VGG16	95.89
ResNet50	96.45
DenseNet121	95.23
InceptionV3	92.85
MobileNetV3Large	96.26
EfficientNetB0	79.33
EfficientNetV2S	96.64
EfficientNetV2M	**96.80**
EfficientNetV2L	83.54

6 Ensemble Learning

The idea behind combining different models stemmed from the belief that if each model could extract specific features, they might complement each other when combined. To leverage this, a promising approach was to concatenate the feature vectors generated by these extractors and feed them into the same classifier, allowing the model to learn from a diverse set of features. However, the complexity and size of such models, combined with the need for advanced data augmentation techniques to increase dataset size, ultimately made further experimentation with this architecture infeasible due to the limited computational resources and time required.

Configuration	F1 Score (%)	Score (%)
V2S + V2M + V2L	87.88	65.00

7 Data Augmentation

The significant discrepancy between local test scores and online test scores was attributed to the poor generalizability of the models. While standard data

augmentation techniques produced higher results in local tests, they were believed to be insufficient for generating enough variability in class features. To address this limitation, advanced augmentation methods, such as MixCut, CutMix, and RandAugment, were explored to introduce greater variability.

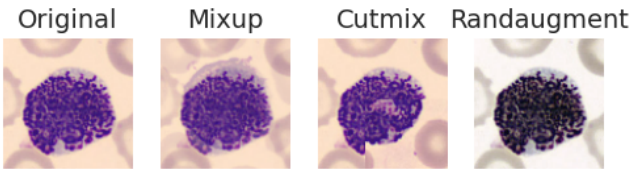

Figure 3: Examples of different augmentation techniques applied to the same image.

The configurations of these techniques, both standalone and in combination with traditional methods, are detailed in the following table.

Configuration	F1 Score (%)
Traditional	97.01
MixUp + CutMix	81.27
MixUp + CutMix + Traditional	87.67
RandAugment	95.58

Among these, RandAugment demonstrated notably superior performance, and as a result, it will be adopted as a standard for future versions.

7.1 Grad-CAM

Grad-CAM (Gradient-weighted Class Activation Mapping) was employed to visualize the regions of the images that contributed most to the model's predictions. This technique highlights the features the model without dense layers focuses on during classification, providing interesting insights into the decision-making process.

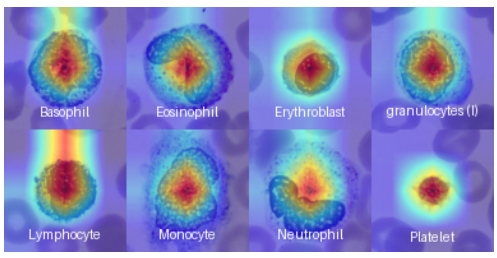

Figure 4: Grad-CAM visualizations highlighting the regions of interest for each cell type.

8 Classifier Head

Once the data augmentation pipeline was established, testing initially focused on fine-tuning the hyperparameters, specifically the learning rate (LR) and the percentage of unfrozen layers (UL), before configuring the dense layers. This sequence was driven by time and resource constraints. To evaluate the effects of unfreezing, initial tests were conducted with a single dense layer configuration. Following best practice, BatchNormalization layers were kept frozen. In hindsight, starting with the dense layer configuration and then fine-tuning the hyperparameters would have been the best practice and likely led to better results.

UL (%) + LR	F1 Score (%)	Score (%)
60 + 1e-4	97.69	84.00
80 + 5e-4	97.53	85.00

It became clear that a more aggressive learning rate and increased unfreezing yielded marginally better results. After setting the unfreezing percentage, different dense layer configurations were then explored.

Classifier (units)	F1 Score (%)	Score (%)
128	97.53	85.00
256 + 128	98.75	82.00

It was observed that adding more dense layers while keeping the unfreezing percentage constant led to worse results, likely due to overfitting.

9 Final Model

The final model achieved an accuracy of 85% on the online test.

Layer (type)	Output Shape	Param
input	(None, 96, 96, 3)	0
efficientnetv2-m	(None, 1280)	53,150,388
global_avg_pool	(None, 1280)	0
dense	(None, 128)	163,968
re_lu	(None, 128)	0
dropout	(None, 128)	0
dense	(None, 8)	1,032
softmax	(None, 8)	0

References

[1] R. Asghar, S. Kumar, and A. Shaukat. A review on classification of white blood cells using machine learning models. 08 2023.

[2] F. Rustam, N. Aslam, I. De la Torre Díez, Y. Khan, J. L. Mazón, C. Rodríguez, and I. Ashraf. White blood cell classification using texture and rgb features of oversampled microscopic images. *Healthcare*, 10:2230, 11 2022.

[3] T. Tamang, S. Baral, and M. Paing. Classification of white blood cells: A comprehensive study using transfer learning based on convolutional neural networks. *Diagnostics*, 12:2903, 11 2022.