IMA205 Challenge 2026: White Blood Cell Classification

Théo Palagi

April 2026

1 Abstract

Automatic classification of white blood cells (WBC) from microscopic images is a major challenge for haematological diagnosis. In this project, we address a 13-class classification problem from 28,901 training images, with an extreme class imbalance (1183:1 ratio between the majority and the rarest class). We first develop a classical machine learning approach based on handcrafted feature extraction (morphological, colorimetric, and textural), inspired by the foundational work of Bikhet et al. [1] and Rustam et al. [2]. Progressive feature enrichment, dual cell/nucleus segmentation, and SMOTE rebalancing [7] significantly improve the macro F1-score. This first part of the report describes the ML approach before addressing deep learning methods in the second part. This chronological ordering reflects the pedagogical progression of the IMA205 course, where classical ML techniques were studied before deep learning.

2 Introduction

2.1 Medical context

White blood cells (leukocytes) constitute the first line of defence of the human immune system. Five main types are distinguished in a healthy individual: neutrophils (40–80%), lymphocytes (20–40%), monocytes (2–10%), eosinophils (1–6%), and basophils (<1%) [2]. An abnormality in the proportion of these cells can reveal serious pathologies, including leukaemia. Traditionally, the differential count is performed manually by haematologists under a microscope, a time-consuming task subject to inter-observer variability [1].

2.2 Objective

The objective is to develop an automatic WBC classification system from microscopic images. In this first part, we adopt a classical machine learning approach based on handcrafted feature extraction. This choice is motivated by:

- **Pedagogical progression**: at the time of starting this project, deep learning techniques had not yet been covered in the course. We therefore naturally began with the classical ML methods studied in the IMA205 module.
- **Interpretability**: each extracted feature corresponds to an observable property of the cell (nucleus size, cytoplasm colour, texture), which allows understanding and validating the model's decisions.

2.3 Dataset

The dataset comprises 28,901 training images and 9,634 test images, distributed across 13 white blood cell classes. The challenge evaluation metric is the **macro F1-score**, defined as:

$$F1_{\text{macro}} = \frac{1}{C} \sum_{c=1}^{C} F1_c \tag{1}$$

where $C = 13$ is the number of classes and $F1_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$.

The class distribution is highly imbalanced, which constitutes one of the major challenges of this project:

Table 1: Class distribution in the training set.

Class	Images	%
SNE (Segmented Neutrophils)	13,015	45.0
LY (Lymphocytes)	8,101	28.0
MO (Monocytes)	2,746	9.5
BL (Blasts)	2,012	7.0
EO (Eosinophils)	861	3.0
MY (Myelocytes)	441	1.5
BA (Basophils)	415	1.4
BNE (Band Neutrophils)	391	1.4
VLY (Variant Lymphocytes)	366	1.3
MMY (Metamyelocytes)	360	1.2
PMY (Promyelocytes)	114	0.4
PC (Plasma Cells)	68	0.2
PLY (Prolymphocytes)	11	0.04

We observe an extreme imbalance. The majority class (SNE) contains 13,015 samples while the rarest class (PLY) contains only 11, yielding a ratio of 1183:1.

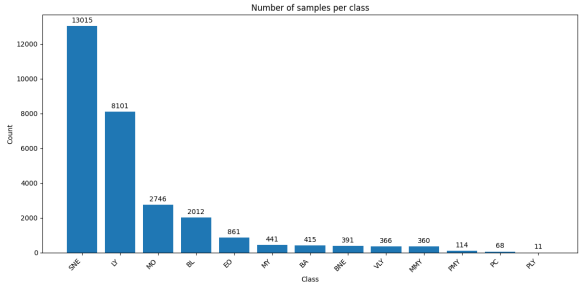

Figure 1: Class distribution in the training set. The imbalance between majority classes (SNE, LY) and minority classes (PLY, PC) is clearly visible.

3 Methodological approach

The processing pipeline consists of four steps, illustrated in Figure 2. Cell segmentation, feature extraction, class rebalancing, and classification via machine learning algorithms.

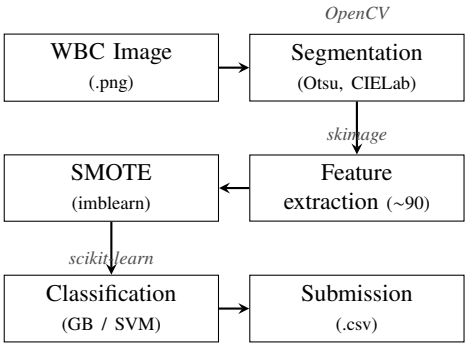

Figure 2: ML classification pipeline.

3.1 Cell segmentation

3.1.1 Initial version: HSV thresholding

The first step consists of isolating the cell of interest from the image background. White blood cells exhibit significantly higher colour saturation than the background and red blood cells. The initial procedure is:

1. Conversion of the BGR image to HSV colour space and extraction of the saturation channel.

2. Automatic Otsu thresholding on the saturation channel. Otsu's method automatically determines the optimal threshold by minimizing intra-class variance [8], without manual parameter tuning.

3. Contour detection and selection of the largest contour (the cell), eliminating artefacts and debris.

4. Creation of a filled binary mask for feature extraction.

The choice of HSV saturation-based segmentation rather than direct greyscale thresholding is motivated by the fact that saturation is more discriminative for separating coloured cellular structures from the pale background, as highlighted by Bikhet et al. [1].

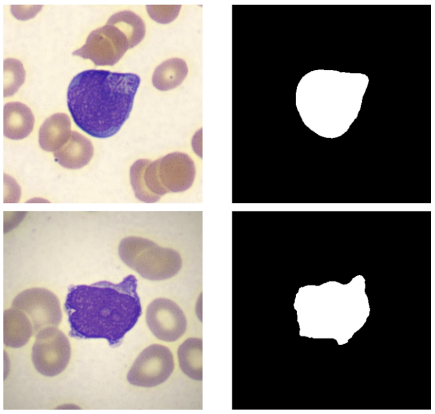

Figure 3: HSV saturation-based cell segmentation. Left: original image. Right: binary mask obtained by Otsu thresholding on the saturation channel, isolating the white blood cell from the background.

3.1.2 Improved version: dual cell/nucleus segmentation

Analysis of the confusion matrices revealed that biologically similar classes (BNE vs SNE, PLY vs LY) were systematically confused.

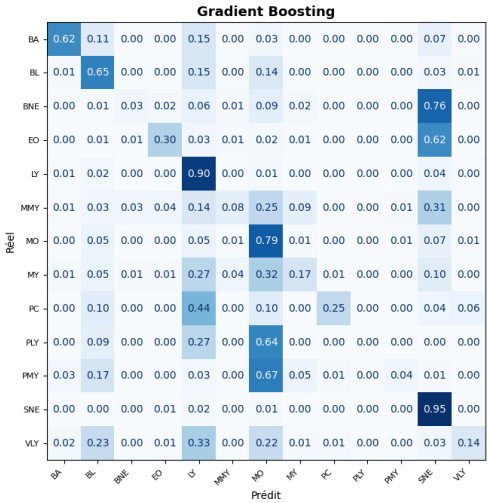

Figure 4: Confusion matrix (GradientBoosting, 89 features, cell segmentation only). BNE is heavily confused with SNE, motivating the addition of nucleus-specific features.

As highlighted by Bikhet et al. [1], it is the nucleus that most differentiates these classes: number of lobes, relative size, and shape. We therefore developed a dual segmentation:

1. **Cell mask**: conversion to CIELab colour space, inversion of the L (luminance) channel, Otsu thresholding, morphological closing, and hole filling.

2. **Nucleus mask**: a second Otsu threshold applied only to pixels within the cell on the L channel. The nucleus, being denser and darker, is isolated by this double thresholding. Morphological closing and hole filling refine the result.

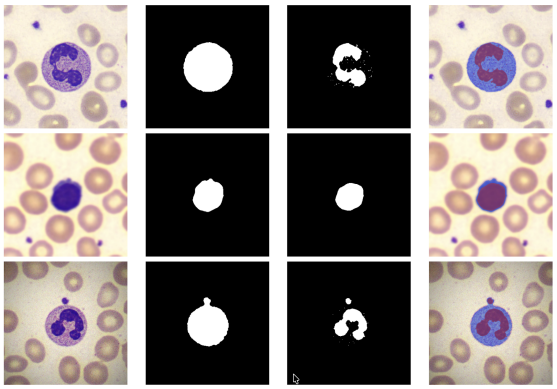

Figure 5: Dual cell/nucleus segmentation. Left: original image. Centre: cell mask (CIELab + Otsu). Right: nucleus mask (second Otsu threshold).

3.2 Feature extraction

3.2.1 Initial version: 9 features

In our first iteration, we extracted 9 features per image, grouped into three categories:

Morphological features. Computed from the binary mask via the `skimage.measure` library: **area** (pixel count), **perimeter** (contour length), **eccentricity** (0 = circle, 1 = elongated), and **solidity** (area/convex hull ratio). Cell size is indeed an important criterion.

Colorimetric features (RGB). Different types of leukocytes bind differently to the stains (Giemsa, Wright) used during smear preparation [1]. We compute the means of the R, G, B channels on the cell pixels only (via the mask), excluding the background.

Texture features (GLCM). To quantify texture, we use the Grey-Level Co-occurrence Matrix (GLCM), introduced by Haralick et al. [4]. We extract **contrast** (abrupt greyscale transitions) and **homogeneity** (texture uniformity).

3.2.2 Enriched version: 89 features

Analysis of preliminary results showed that 9 features were insufficient to discriminate 13 classes. We therefore considerably enriched the extraction, drawing from the literature:

Enriched morphological features (~21 features).

- **Circularity** ($4\pi \times \text{area}/\text{perimeter}^2$): identifies cell shape. Some cells are very round while others have numerous lobes.
- **Hu moments** [6]: 7 statistical moments invariant to translation, rotation, and scaling, capturing the overall cell shape regardless of orientation.
- **Nucleus/cell ratio**: identified by Bikhet et al. [1] as the most discriminative feature.
- **Nucleus features**: area, circularity, eccentricity, and solidity of the nucleus separately, and cytoplasm area.

Enriched colorimetric features (~39 features).

- **RGB standard deviations**: capture colour heterogeneity.
- **HSV statistics**: hue (H) encodes the dominant granule colour.
- **Colour ratios** (R/G, R/B, G/B): normalize brightness differences across images.
- **Region-wise statistics**: means and standard deviations computed separately on the whole cell, the nucleus alone, and the cytoplasm alone (cell − nucleus). This separation is made possible by the dual segmentation.

Enriched texture features (~30 features).

- **Complete GLCM**: 5 properties (contrast, homogeneity, energy, correlation, dissimilarity) over 4 angles (0°, 45°, 90°, 135°), yielding 20 features.
- **LBP (Local Binary Patterns)** [5]: a descriptor complementary to GLCM. LBP encodes the local structure around each pixel as a binary pattern. This adds 10 features.

Table 2: Summary of extracted features in the final version.

Category	V1	V2
Morphology (cell + nucleus)	4	~21
Colour (RGB + HSV, 3 regions)	3	~39
Texture GLCM	2	20
Texture LBP	0	10
Total	**9**	**89**

3.3 Handling class imbalance

The extreme dataset imbalance (1183:1 ratio) is addressed by two mechanisms:

Class weighting. The `class_weight='balanced'` option, natively available in scikit-learn, assigns a weight inversely proportional to each class frequency in the loss function. Thus, an error on a PLY sample (11 images) carries as much weight as ~1183 errors on SNE.

SMOTE. The SMOTE technique (*Synthetic Minority Oversampling Technique*) [7] generates synthetic samples for minority classes by interpolation between neighbours in feature space. We use `k_neighbors=3` (instead of the default 5) because the PLY class contains only 11 samples. SMOTE is integrated into an `imblearn` pipeline to be applied only on the training data of each fold, preventing data leakage.

3.4 Evaluated classifiers

We evaluated six classification models by 5-fold cross-validation, after feature normalization with `StandardScaler`:

- **Logistic Regression**: linear model, `class_weight='balanced'`.
- **KNN** (*K-Nearest Neighbours*): classification by majority vote of the k nearest neighbours. Does not support `class_weight`.
- **SVM (RBF)**: support vector machines with RBF kernel, `class_weight='balanced'`.
- **Random Forest**: `class_weight='balanced'`.
- **AdaBoost**: adaptive boosting.
- **Gradient Boosting**: gradient boosting.

4 Results and analysis

4.1 Preliminary results (9 features, no rebalancing)

Table 3: Results with 9 features (initial version).

Model	Accuracy	Macro F1
Random Forest	78.3%	32.6%
Gradient Boosting	76.6%	30.5%
KNN	74.0%	29.3%
SVM (RBF)	76.6%	25.6%
AdaBoost	66.3%	17.4%
Logistic Regression	66.5%	16.5%

Two observations emerge:

- **Accuracy is misleading**: Random Forest achieves 78.3%, but a naive model systematically predicting SNE would already reach 45%. Accuracy is therefore an insufficient metric for an imbalanced dataset.
- **Macro F1 is very low**: the best macro F1 (32.6%) indicates that models largely fail on minority classes.

4.2 Results after enrichment (89 features, with `class_weight`)

Table 4: Results with 89 features and `class_weight='balanced'`.

Model	Accuracy	Macro F1
Gradient Boosting	82.4%	40.0%
Random Forest	83.1%	39.1%
SVM (RBF)	68.3%	38.1%
Logistic Regression	66.9%	38.1%
KNN	76.2%	27.7%
AdaBoost	73.5%	20.0%

The transition from 9 to 89 features and the addition of `class_weight='balanced'` yield an improvement of ~7 macro F1 points. However, performance remains modest (~40%).

4.3 Confusion matrix analysis

Figure 6: Normalized confusion matrices (per-class recall) for the six evaluated models.

Analysis of the normalized confusion matrices reveals the following confusion patterns for Gradient Boosting:

Well-recognized classes. SNE (recall 0.95), LY (0.90), MO (0.79), BA (0.62). These are the classes with the most data.

Biologically coherent confusions. As discussed in the practical sessions, it is important to analyse the coherence of our models with biological reality.

- **BNE → SNE**
- **PLY → LY**
- **MMY/MY/PMY**

The transition to 89 features and the addition of `class_weight='balanced'` yields a clear improvement in F1 score.

4.4 Results with SMOTE and optimization

For the best model (Gradient Boosting), hyperparameter optimization is performed via `GridSearchCV` with SMOTE integrated into the pipeline. The application of SMOTE combined with `GridSearchCV` optimization on the best model's hyperparameters allows comparing the effect of rebalancing. The GridSearch explores `n_estimators` $\in \{100, 200, 300\}$, `max_depth` $\in \{3, 5, 7\}$. It ultimately identifies the optimal parameters `n_estimators=200`, `max_depth=5`. This yields the best ML result with a macro F1 of **0.491**.

Figure 7: Confusion matrix of the best model.

We observe numerous patterns. First, BNE is confused with SNE. Second, the granulocytic lineage (MMY/MY/PMY) exhibits significant mutual confusions. Finally, PLY is never correctly predicted. SMOTE cannot compensate for such an extreme lack of diversity. These errors confirm that handcrafted features encode overall size and colour well but fail on the fine patterns that distinguish biologically similar classes.

5 Discussion

5.1 Limitations of the classical ML approach

Despite feature enrichment and rebalancing, performance remains limited by several factors:

- **Segmentation quality**: some images exhibit touching cells, staining artefacts, or blurry/noisy images. Automatic segmentation sometimes fails, including neighbouring red blood cells or merging adjacent cells into the mask.
- **Lack of data for rare classes**: even with SMOTE, synthesizing features from 11 samples (PLY) cannot compensate for the lack of real diversity.
- **Intrinsic biological similarity**: handcrafted features, even enriched, struggle to capture subtle differences.
- **Loss of spatial information**: feature extraction reduces an entire image to a vector of 89 values, losing spatial information and local patterns that convolutional networks can capture.

5.2 Comparison with the literature

Rustam et al. [2] achieve 97% accuracy with Random Forest on a 5-class dataset rebalanced by SMOTE. However, their context is very different. They have only 5 classes (vs 13), a much smaller dataset (3,539 images vs 28,901), and raw features (flattened RGB pixels + texture, selected by Chi^2) rather than handcrafted features. Bikhet et al. [1] achieve 91% on 71 cells in 5 classes with morphological features similar to ours. The 13-class problem with extreme imbalance is considerably more difficult.

Toğaçar et al. [3] use CNNs (AlexNet, GoogLeNet, ResNet-50) as feature extractors followed by a QDA classifier, achieving 97.95% on 5 classes. This hybrid approach (deep features + classical classifier) will be explored in the second part of this report.

5.3 Transition to deep learning

The identified limitations therefore motivate the transition to deep learning. Convolutional networks (CNNs)

automatically learn representations directly from pixels, without requiring prior segmentation or manual feature selection. This transition is the subject of the second part of this report.

6 Conclusion of the ML part

These results, while insufficient for the challenge, constitute a solid and interpretable baseline. They highlight the intrinsic limitations of handcrafted features for this strongly imbalanced 13-class problem, and motivate the exploration of deep learning methods in the following part.

7 Deep learning approach

7.1 Motivations

The limitations identified in the ML part motivate the transition to deep convolutional networks. Three obstacles prove insurmountable with handcrafted features: the loss of spatial information when reducing to a vector of 89 values, the inability to capture fine patterns distinguishing biologically similar classes such as BNE and SNE, and the near-impossibility of handling PLY with only 11 images. CNNs automatically learn hierarchical representations directly from pixels, without prior segmentation or manual feature selection. In accordance with the challenge rules, only models pre-trained on ImageNet are allowed; any pre-training on medical images is prohibited.

7.2 First architecture: EfficientNet-B3

7.2.1 Transfer learning and imbalance handling

The first model tested is EfficientNet-B3 [9] pre-trained on ImageNet-1K. Rather than training a network from scratch, we adopt a **transfer learning** strategy. Pre-trained ImageNet weights provide generic visual representations (edge detectors, textures, shapes) learned from 1.2 million images, which are then specialized to our white blood cell classification task through fine-tuning. This approach is particularly relevant here because our dataset, while containing 28,901 images, remains insufficient to train a deep network from scratch without overfitting. Indeed, rare classes such as PLY (11 images) or PC (58 images) directly benefit from the pre-learned generic features. The entire backbone is unfrozen from the start of training with a differentiated learning rate: 10^{-4} for the convolutional layers and 10^{-3} for the classification head. This preserves low-level representations while adapting the deeper layers to the medical domain. The original classification head is replaced by an adapted sequence: a `Dropout`(0.4) layer, followed by a linear layer $1536 \to 512$, a `ReLU` activation, a second `Dropout`(0.3), and a final linear layer $512 \to 13$. The extreme dataset imbalance is addressed by two complementary mechanisms. A `WeightedRandomSampler` resamples batches with weights inversely proportional to class frequency, ensuring that a PLY image has as much chance of appearing in a batch as an SNE image. In parallel, a weighted `CrossEntropyLoss` penalizes errors on rare classes more heavily. The weights are computed using a log-dampened formula that prevents weight explosion for ultra-rare classes:

$$w_c = \frac{\log\left(1 + \frac{n_{\max}}{n_c}\right)}{\frac{1}{C}\sum_{c'} \log\left(1 + \frac{n_{\max}}{n_{c'}}\right)} \tag{2}$$

This choice is motivated by the fact that a classical $1/n_c$ weight assigns a weight of 9.24 to PLY ($n = 11$), creating training instability. The logarithmic formulation reduces this weight to 2.15, a PLY/SNE ratio of 10 instead of 1320, while maintaining sufficient over-representation of minority classes.

7.2.2 Results and embedding analysis

Training for 15 epochs with a `CosineAnnealingLR` scheduler achieves a best validation macro F1 of **0.646**, corresponding to a leaderboard score of **0.6850**. This result already represents a major improvement over classical ML (+19 points). However, to understand the model's limitations, the embeddings from the `avgpool` layer are extracted and projected into 2D via t-SNE.

SNE (cyan) and LY (red) form compact and well-separated clusters. In contrast, BNE, MMY, MY, PMY, and PLY form an indistinct central blob. The model has therefore not learned to separate these classes in its representation space. This visualization is revealing: the biologically similar classes of the granulocytic lineage

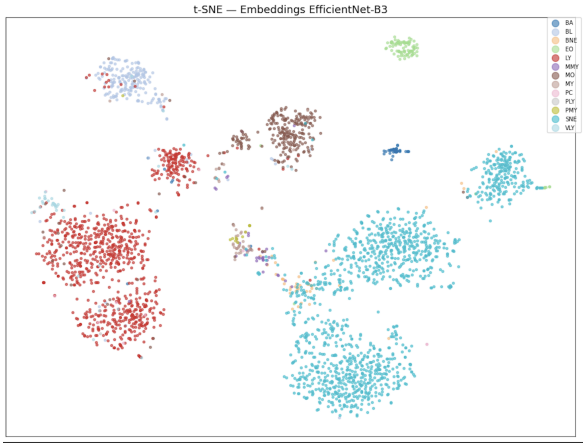

Figure 8: t-SNE projection of EfficientNet-B3 embeddings on the validation set.

(MMY, MY, PMY) and PLY are all superimposed at the centre of the projection, which directly explains their low individual F1 scores (PLY: 0.00, MMY: 0.42, PMY: 0.17). The feature space learned by EfficientNet does not sufficiently structure these classes. This observation motivates two developments: changing architecture to better separate the embeddings, and adopting a loss function better suited to difficult classes.

The confusion matrix, Figure 9, indeed confirms the observations made with the t-SNE projection. We note that PLY is systematically confused with LY, which dramatically lowers the F1 score. Similarly, biologically similar cells are correctly recognized only 50% of the time. They are extensively confused with each other. This is partly due to the fact that the embedding does not adequately separate these classes in the feature space.

7.3 Towards better separation: ConvNeXt-Tiny and Focal Loss

7.3.1 Architecture change

EfficientNet-B3 is replaced by **ConvNeXt-Tiny** [10], with LayerNorm layers and a GELU activation that improves gradient flow in the deeper layers. The classification head

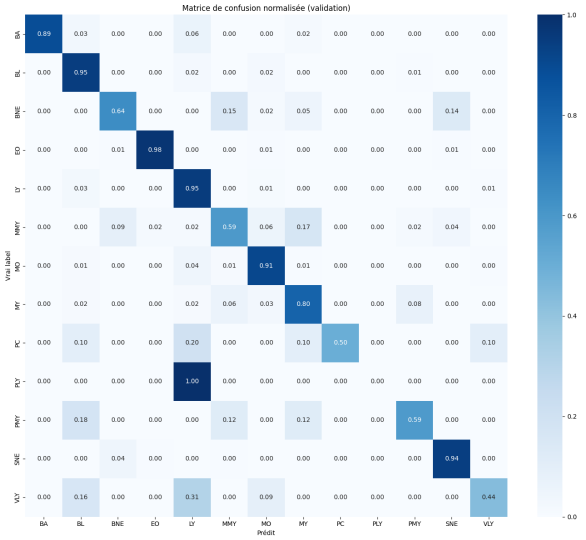

Figure 9: Normalized confusion matrix of ConvNeXt-Tiny + Focal Loss (validation set, F1 macro = 0.696).

is adapted:

$$\begin{aligned} &\text{Linear}(768 \to 512) \\ &\to \text{GELU} \to \text{Dropout}(0.3) \\ &\to \text{Linear}(512 \to 13) \end{aligned} \tag{3}$$

The intermediate layer of 512 neurons with GELU and Dropout allows the model to learn non-linear combinations of backbone features before final classification, while regularizing through Dropout to prevent overfitting. The backbone is fine-tuned with a differentiated learning rate: 10^{-4} for the convolutional layers and 10^{-3} for the classification head, preserving ImageNet representations while adapting the head to the problem.

7.3.2 Focal Loss

The EfficientNet embedding analysis showed that the network concentrated its efforts on SNE and LY (the easy-to-classify classes), to the detriment of rare classes. **Focal Loss** [11] addresses exactly this problem by dynamically reducing the contribution of well-classified examples:

$$\mathcal{L}_{\text{focal}}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{4}$$

With $\gamma = 2$, a very well-classified example ($p_t = 0.9$) sees its loss contribution reduced by a factor of $(1 - 0.9)^2 =$

0.01, forcing the network to focus its learning on difficult examples. This mechanism is particularly suited to our context where SNE (45% of the dataset) is trivially classified from the first epochs, while PLY, PC, and PMY remain systematically mispredicted.

7.3.3 Enriched augmentation

The augmentation is considerably enriched compared to the EfficientNet version, leveraging the biological specificities of blood cells. Indeed, blood cells have no preferred orientation, which justifies a full 360 rotation instead of 15. A `RandomAffine` with translation and shear simulates positioning variations during slide preparation. `RandomErasing` randomly masks small regions, forcing the model not to rely on a single cell region.

Table 5: Final augmentation pipeline.

Transformation	Parameters
HorizontalFlip	$p = 0.5$
VerticalFlip	$p = 0.5$
Rotation	$[-180, +180]$
Affine	transl. 0.10, scale $[0.85, 1.15]$
ColorJitter	bright. 0.3, contr. 0.3, sat. 0.2
Erasing	$p = 0.25$, scale $[0.02, 0.12]$

7.4 Resolving the local minimum problem: CosineAnnealingWarmRestarts

During the first runs with ConvNeXt-Tiny and a classical `CosineAnnealingLR` scheduler over 30 epochs, a systematic plateau is observed. The validation F1 stagnates around 0.6460 from epoch 8, and early stopping triggers at epoch 15 without notable improvement. The validation loss stops decreasing well before the end of training, a sign that the optimizer is trapped in a local minimum that the learning rate, too low at the end of the cycle, no longer allows it to escape.

To remedy this, the scheduler is replaced by a `CosineAnnealingWarmRestarts` with $T_0 = 15$ epochs over 60 total epochs. The principle is to periodically restart the learning rate to its maximum value, allowing the model to escape local minima:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min})\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_0}\pi\right)\right) \quad (5)$$

At each restart (epochs 16, 31, 46), the learning rate returns to $\eta_{\max}$, causing a controlled perturbation of the weights and enabling exploration of new regions of parameter space. The number of epochs is increased to 60 to cover 4 complete cycles, and the early stopping patience is increased to 20 to avoid premature interruption during the exploration phases at the beginning of each cycle.

The restarts are visible on the training curves (Figure 10) as periodic oscillations in the validation F1. The best score is reached at epoch 28, at the end of the second cycle, with a validation macro F1 of **0.696**.

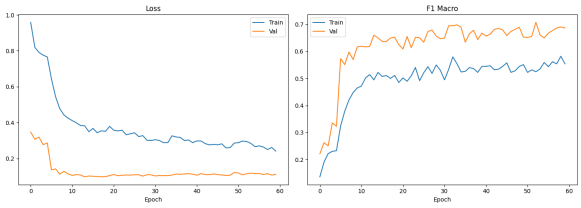

Figure 10: Loss and macro F1 curves during training (ConvNeXt-Tiny, 60 epochs, WarmRestarts $T_0 = 15$).

7.5 Improvement: Test Time Augmentation inference

The final model is evaluated with **Test Time Augmentation** (TTA). Rather than predicting each test image once, we perform $R+1$ passes (1 standard pass without augmentation and $R = 8$ passes with random augmentations) and average the softmax probabilities. A temperature $T = 0.8$ is applied before the softmax to sharpen the distributions:

$$\hat{p}_c = \frac{1}{R+1}\sum_{r=0}^{R} \text{softmax}\left(\frac{f_\theta(\tilde{x}^{(r)})}{T}\right)_c \quad (6)$$

This averaging reduces prediction variance without requiring retraining, providing a gain of +0.014 points on the leaderboard.

7.6 Results

Table 6: Macro F1 score progression on the Kaggle leaderboard.

Configuration	Val F1	Leaderboard
Classical ML (GradientBoosting)	0.521	0.491
EfficientNet-B3 + CrossEntropy	0.646	0.6850
ConvNeXt-Tiny + FocalLoss + WarmRestarts	0.6960	0.7327
+ TTA ($R = 8$, $T = 0.8$)	0.6983	**0.7467**

We note a systematic gap of approximately +0.04 between the validation F1 and the leaderboard score. This gap is explained by the fact that the validation set, drawn from a 15% stratified split, contains proportionally more images of ultra-rare classes (PLY: 2 images, PC: 10 images), making the macro F1 more volatile and more pessimistic than on the official test set. Moreover, this gap is even more pronounced for TTA since we do not apply it on the validation set to reduce the already long computation time.

7.6.1 Per-class analysis

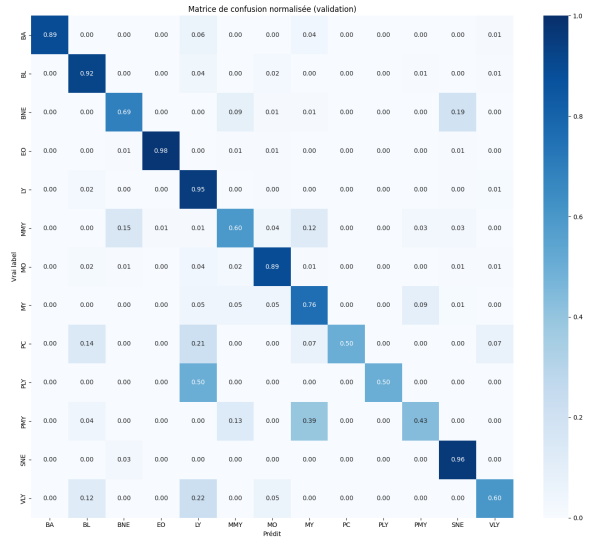

Figure 11: Normalized confusion matrix of the best ConvNeXt-Tiny + Focal Loss model on the validation set.

The confusion matrix analysis reveals significant improvements over EfficientNet, particularly on rare classes. PLY goes from being systematically misclassified to 0.50 and VLY from 0.44 to 0.60. These improvements are consistent with the expected effect of Focal Loss, which forces the model to better learn difficult classes. The residual confusions remain biologically coherent. BNE is confused at 19% with SNE. PLY remains confused with LY, which are morphologically very similar. The granulocytic lineage (MMY, MY, PMY) exhibits mutual confusions.

7.6.2 Embedding evolution

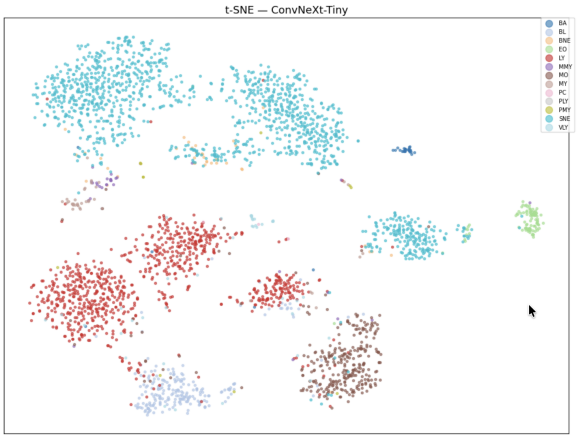

Figure 12: t-SNE projection of ConvNeXt-Tiny embeddings on the validation set.

Compared to EfficientNet (Figure 8), the clusters are better separated. SNE, LY, MO, and EO form distinct regions. The granulocytic lineage nevertheless remains partially mixed. The comparison of t-SNE projections before and after the architecture change is instructive. This improvement in embedding structure is directly correlated with the performance gain observed on the leaderboard.

7.7 Successive optimizations and overfitting diagnosis

The leaderboard score of 0.7467 was an encouraging starting point, but the analysis of training curves and the confusion matrix revealed several avenues for improvement. This section retraces the successive optimizations, each motivated by a precise diagnosis of model behaviour.

7.7.1 Adding MixUp

Diagnosis. The training curves showed an increasing gap between train F1 (∼0.85) and val F1 (∼0.70), a sign of overfitting. The model was memorizing training examples instead of generalizing.

Solution. MixUp [12] generates virtual examples by convex interpolation:

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j, \quad \mathcal{L} = \lambda \mathcal{L}(f(\tilde{x}), y_i) + (1-\lambda)\mathcal{L}(f(\tilde{x}), y_j) \tag{7}$$

with $\lambda \sim \text{Beta}(0.4, 0.4)$. This regularization prevents the model from learning overly sharp decision boundaries between classes.

Result. The validation F1 increases from 0.696 to **0.730** (+3.4 points). However, the Kaggle submission yields 0.7279, *lower* than the previous score of 0.7467. This paradoxical result (better in val, worse in test) indicates that the model now overfits the *validation split*: improvements observed on this split do not generalize to the test set.

Table 7: Result after adding MixUp.

Configuration	Val F1	Leaderboard
Baseline (TTA)	0.696	0.747
+ MixUp ($\alpha = 0.4$)	0.730	0.728

This is what motivated us to strengthen regularization to reduce overfitting, increase dropout probability to 0.5, and enlarge the validation set (from 15% to 20%) to make it more representative of the test set.

7.7.2 Diagnosing triple rebalancing

Problem identified. The pipeline stacked three imbalance-handling mechanisms. The `WeightedRandomSampler` (oversampling of rare classes), the Focal Loss with $\gamma = 2$ (reducing the contribution of easy examples), and log-dampened class weights in the CrossEntropy. This triple stacking produced an abnormally low loss, a sign that gradients were being crushed. Easy examples from dominant classes contributed nearly zero to the loss (Focal Loss), rare classes were over-weighted three times, and the model was effectively learning only from a fraction of the data.

Solution. After experimenting with several configurations, the retained combination is:

- Sampler retained (rebalancing at batch level);
- Class weights computed as $w_c = 1/\sqrt{n_c}$, normalized by the mean, producing weights in $[0.2, 3.0]$.

7.7.3 Backbone freezing

Diagnosis. Despite the differentiated learning rate (5×10^{-5} for the backbone, 10^{-3} for the head), the first training epochs showed instability. The backbone was receiving gradients from a randomly initialized classification head—essentially noisy gradients that degraded the ImageNet representations.

Solution. The backbone is frozen for the first 5 epochs (only the head is trained), then unfrozen at epoch 6 with a new optimizer and a `CosineAnnealingWarmRestarts` scheduler. This two-phase strategy:

1. allows the head to converge towards a decision space consistent with the ImageNet features;
2. prevents destruction of pre-trained representations.

7.7.4 Adding CutMix

Motivation. MixUp blends pixels globally, creating unnatural "ghost" images. CutMix [13] is complementary: it cuts a rectangular region from one image and pastes it into another, preserving local spatial information:

$$\lambda = 1 - \frac{(x_2 - x_1)(y_2 - y_1)}{H \times W} \tag{8}$$

This technique forces the model to exploit the entire image rather than relying on a single discriminative region. At each batch, a random draw determines whether MixUp or CutMix is applied (50/50 probability).

7.7.5 Increasing weight decay

Despite backbone freezing and MixUp/CutMix, the validation loss was increasing from epoch 28 onwards, a sign of residual overfitting. Comparative analysis with a competing pipeline revealed a weight decay of 10^{-2} (default value in their code), versus 5×10^{-4} in ours—i.e., 20× less L2 regularization.

The weight decay is therefore increased to 10^{-2} in AdamW, penalizing large-magnitude weights and forcing more compact representations.

7.7.6 Cumulative results

Table 8 summarizes the progression at each optimization step.

Table 8: Score progression at each optimization step.

Modification	Val F1	Leaderboard
Baseline ConvNeXt + TTA	0.696	0.747
+ MixUp ($\alpha = 0.4$)	0.730	0.728
+ Freeze backbone 5 ep.	0.723	0.7601
+ CutMix 50/50	0.708	0.7626
+ WD 10^{-2} + γ=1 + $1/\sqrt{n}$	0.725	**0.7739**

7.7.7 Final confusion matrix analysis

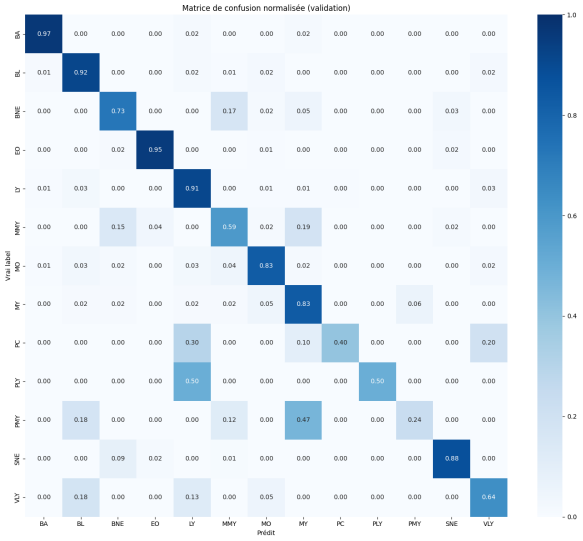

Figure 13: Normalized confusion matrix of the optimized model (ConvNeXt-Tiny, MixUp/CutMix, backbone freezing, WD=10^{-2}, γ=1). Dominant classes achieve recalls above 0.88. PLY (0.50) remains confused with LY due to morphological similarity and only 11 training samples.

The optimized model achieves high recalls on dominant classes: BA (0.97), BL (0.92), EO (0.95), LY (0.91), MO (0.83), SNE (0.88). Rare classes remain problematic:

- **PLY** (recall 0.50): confused with LY, which is biologically expected—morphologically near-identical, and 11 training images are insufficient to learn this distinction.
- **PC** (recall 0.40): confused with LY and VLY.
- **PMY** (0.24): heavily confused with MY.
- **VLY** (0.64): confused with LY and BL, as the variant lymphocyte morphology approaches that of the monocyte.
- **MMY** (0.59): confused with BNE and MY, at the crossroads of the granulocytic lineage.

These confusions are consistent with haematological biology and do not reflect methodological shortcomings

but rather the intrinsic limitations of the dataset for ultra-rare classes.

8 General conclusion

This project illustrates how an iterative approach, guided by systematic analysis of intermediate results, enables methodical progress on a difficult classification problem.

Classical ML, based on 89 handcrafted features and GradientBoosting with SMOTE, achieves a macro F1 of 0.491. This approach, while interpretable, is limited by the loss of spatial information and the inability to capture fine patterns distinguishing biologically similar classes.

The transition to deep learning proceeds in three phases, each motivated by a precise diagnosis:

1. **EfficientNet-B3 + weighted CrossEntropy** (leaderboard 0.685): t-SNE analysis of embeddings reveals insufficient separation of rare classes, motivating the architecture change.

2. **ConvNeXt-Tiny + Focal Loss + WarmRestarts + TTA** (leaderboard 0.747): the observation of a validation plateau leads to the restart scheduler, and Focal Loss forces learning on difficult classes.

3. **MixUp/CutMix + backbone freezing + recalibration** (leaderboard **0.7739**): the overfitting diagnosis (val F1 > test F1) motivates the addition of successive regularizations, and the analysis of triple rebalancing leads to a recalibration of the Focal Loss and class weights.

Overall, the progression from 0.491 to **0.7739** represents a gain of +28.3 macro F1 points. The residual confusions (PLY→LY, BNE→SNE, granulocytic lineage) are biologically coherent and reflect the dataset limitations: with 11 images for PLY, no method can compensate for the fundamental lack of diversity in the training data.

References

[1] S. F. Bikhet, A. M. Darwish, H. A. Tolba, S. I. Shaheen, "Segmentation and Classification of White Blood Cells," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[2] F. Rustam, N. Aslam, I. D. L. T. Díez, Y. D. Khan, J. L. V. Mazón, C. L. Rodríguez, I. Ashraf, "White Blood Cell Classification Using Texture and RGB Features of Oversampled Microscopic Images," *Healthcare*, vol. 10, no. 11, p. 2230, 2022.

[3] M. Toğaçar, B. Ergen, Z. Cömert, "Classification of white blood cells using deep features obtained from Convolutional Neural Network models based on the combination of feature selection methods," *Applied Soft Computing*, vol. 97, p. 106810, 2020.

[4] R. M. Haralick, K. Shanmugam, I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[5] T. Ojala, M. Pietikäinen, T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[6] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[8] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[9] M. Tan, Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, "A ConvNet for the 2020s," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986, 2022.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, "Focal Loss for Dense Object Detection," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[12] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," *ICLR*, 2018.

[13] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features," *ICCV*, pp. 6023–6032, 2019.