IMA205 Challenge 2026
Classification de globules blancs par apprentissage

Hussein El Gouch
Télécom Paris

3 avril 2026

Résumé

Ce rapport présente une méthode pour la classification automatique de globules blancs (WBC) dans des images microscopiques parmi 13 classes. Le problème combine deux difficultés majeures : un fort déséquilibre de classes (ratio 1 :1183 entre SNE et PLY) et la similarité visuelle entre certaines classes correspondant à des stades continus de maturation cellulaire. Notre approche associe 25 caractéristiques morphologiques extraites par des techniques de traitement d'image classiques à des deep features obtenues via un réseau EfficientNet-B2 pré-entraîné sur ImageNet. Le déséquilibre est traité par un WeightedRandomSampler modéré ($\sqrt{1/n_c}$) combiné à une Class-Balanced Focal Loss. Après ré-entraînement sur l'intégralité du set d'apprentissage et application du Test Time Augmentation, nous obtenons un **Macro F1 de 0,7472** sur le leaderboard public.

Table des matières

1 Introduction et contexte clinique

Les globules blancs (leucocytes) sont les effecteurs principaux du système immunitaire. Leur morphologie taille, forme du noyau, aspect du cytoplasme, granulations permet de poser le diagnostic de pathologies hématologiques graves : leucémies, infections bactériennes ou virales, anémies, cancers hématologiques [1]. Traditionnellement, un hématologue identifie visuellement ces cellules sur des frottis sanguins colorés (Giemsa, Wright ou Leishman). Cette tâche manuelle est chronophage et sujette à une forte variabilité inter-observateurs [8], d'où l'intérêt croissant pour les systèmes de diagnostic assisté par ordinateur (CAD).

1.1 Problème posé

Le challenge IMA205 consiste à classer 28901 images de WBC dans un ensemble d'apprentissage étiqueté et 9634 images dans un ensemble de test parmi les 13 classes suivantes, résumées Tableau 1.

Code	Nom complet	Famille	Effectif train
SNE	Segmented neutrophil	Granulocyte mature	13015
LY	Lymphocyte	Agranulocyte	8101
MO	Monocyte	Agranulocyte	2746
BL	Blast cell	Précurseur immature	2012
EO	Eosinophil	Granulocyte mature	861
MY	Myelocyte	Stade de maturation	441
BA	Basophil	Granulocyte mature	415
BNE	Band-form neutrophil	Granulocyte immature	391
VLY	Variant lymphocyte	Agranulocyte atypique	366
MMY	Metamyelocyte	Stade de maturation	360
PMY	Promyelocyte	Stade de maturation	114
PC	Plasma cell	Lymphocyte différencié	68
PLY	Prolymphocyte	Lymphocyte immature	11

TABLE 1 – Les 13 classes de globules blancs à distinguer et leur effectif d'apprentissage. Le ratio de déséquilibre maximal est de 1 :1183.

1.2 Contraintes et métrique

La métrique d'évaluation est le Macro-averaged F1 score, définie comme la moyenne non pondérée des F1 par classe :

$$\text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C} F1_c, \quad \text{avec} \quad F1_c = \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}. \tag{1}$$

Cette métrique accorde autant d'importance aux classes rares qu'aux classes fréquentes, ce qui rend le déséquilibre particulièrement pénalisant : une classe comme PLY (11 échantillons) contribue autant au score qu'SNE (13015 échantillons).

La contrainte principale imposée par l'énoncé est que seuls les modèles pré-entraînés sur ImageNet sont autorisés. Tout pré-entraînement sur des données médicales ou hématologiques est interdit, ce qui exclut les architectures spécialisées (MedNet, PathoNet, etc.) et impose une stratégie de transfer learning depuis un domaine non médical.

1.3 Démarche scientifique adoptée

Plutôt que de chercher aveuglément à maximiser le score, nous avons suivi une démarche structurée. Une baseline simple (EfficientNet-B0, 8 features manuelles) a d'abord été développée, puis a été analysée pour identifier ses points faibles. Cette analyse a guidé les améliorations successives : enrichissement des features, passage à EfficientNet-B2 plus capacitif, traitement du déséquilibre, grosse augmentation , puis ré-entraînement sur la totalité du jeu d'apprentissage. La Section 9 synthétise cette progression avec les scores obtenus à chaque étape.

2 Analyse exploratoire et difficultés spécifiques

Avant toute conception de modèle, l'analyse du dataset révèle trois difficultés qui conditionnent toutes les choix ultérieurs.

2.1 Déséquilibre extrême

La Figure 1 montre la distribution des classes, qui suit une loi de type long-tail. À eux seuls, SNE et LY représentent 73 % du corpus, tandis que PLY, PC, PMY et MMY rassemblent moins de

2 %. Pour le Macro F1, ignorer PLY coûte autant qu'ignorer SNE, d'où la nécessité d'une stratégie dédiée (cf. Section 5).

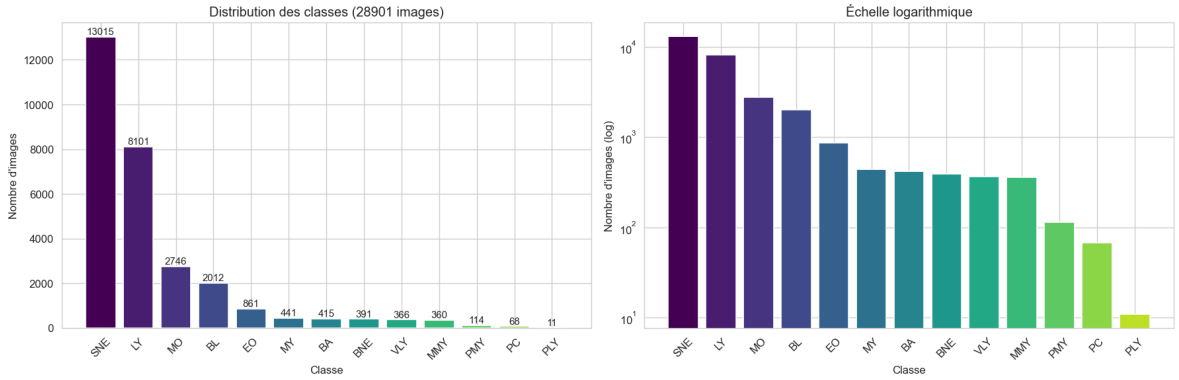

FIGURE 1 – Distribution des classes dans l'ensemble d'apprentissage. À gauche en échelle linéaire, à droite en échelle logarithmique.

2.2 Similarité visuelle entre classes

Certaines classes sont des stades continus d'un même lignage cellulaire [1]. Le continuum granulocytaire BL → PMY → MY → MMY → BNE → SNE représente la maturation d'un neutrophile, avec des différences morphologiques subtiles (degré de lobulation du noyau, apparition des granulations). De même, LY, VLY, PLY et PC sont tous issus de la lignée lymphoïde et se distinguent surtout par la taille et la forme du noyau. Ces ambiguïtés visuelles fixent un plafond théorique au score : même un hématologue expérimenté présente une variabilité inter-observateurs non nulle sur ces classes [2].

2.3 Variations d'acquisition

Les frottis sanguins sont colorés selon différents protocoles (Giemsa, Wright, Leishman), ce qui induit des variations de teinte importantes [22]. De plus, l'éclairage microscopique et la résolution des caméras varient entre laboratoires. Ces variations doivent être compensées par une stratégie d'augmentation adaptée (Section 4).

3 Extraction de caractéristiques manuelles

L'énoncé demande explicitement l'extraction de caractéristiques classiques (asymétrie, bordures, couleur, dimension). Nous avons conçu un pipeline d'extraction de 25 features regroupées en cinq catégories, détaillées Tableau 2.

3.1 Segmentation du noyau

Le noyau est la structure la plus discriminante entre types de WBC : son nombre de lobes, sa forme et sa taille relative au cytoplasme permettent à eux seuls de différencier la majorité des classes [8]. Or, grâce à la coloration Romanowsky, le noyau apparaît fortement coloré (violet sombre) dans l'image, donc il correspond à la région de plus forte saturation dans l'espace colorimétrique HSV.

Notre procédure de segmentation suit donc ces étapes (Figure 2) :

1. Conversion de l'image RGB en HSV et extraction du canal S (Saturation).
2. Seuillage automatique d'Otsu [19] sur le canal S pour séparer noyau et cytoplasme.
3. Ouverture morphologique (élément structurant 5×5) pour éliminer le bruit granulaire, suivie d'une fermeture pour boucher les petits trous à l'intérieur du noyau.

4. Extraction de la plus grande composante connexe, qui correspond au noyau principal.

Cette approche s'inspire de Rezatofighi et Soltanian-Zadeh [21] et Gautam et Bhadauria [7] ; elle est plus robuste qu'un seuillage sur niveau de gris car elle est peu sensible aux variations d'illumination.

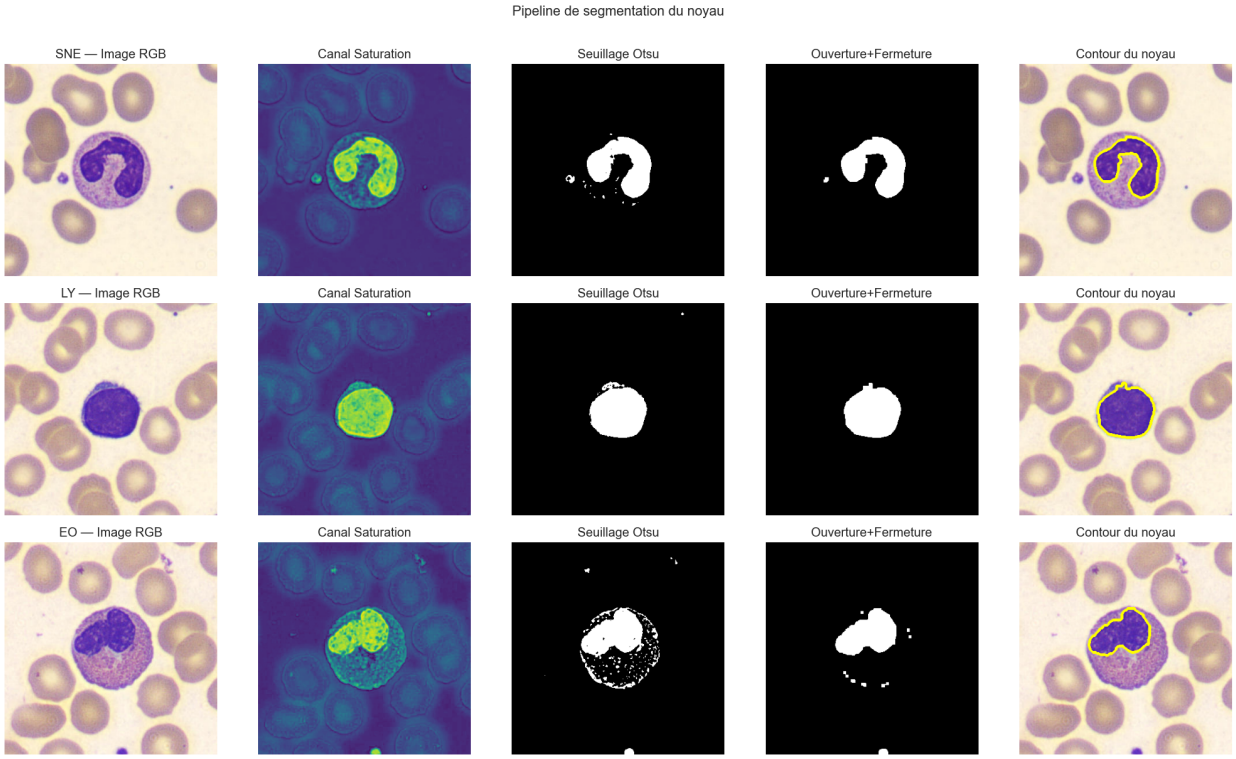

FIGURE 2 – Pipeline de segmentation du noyau sur trois classes illustratives (SNE, LY, EO). De gauche à droite : image RGB originale, canal Saturation HSV, masque binaire après Otsu, masque après ouverture-fermeture morphologique, contour final superposé sur l'image.

3.2 Features calculées

Catégorie	Features	Justification scientifique
Dimension (3)	`area`, `perimeter`, `extent`	Distinguent les monocytes (grands) des lymphocytes matures (petits) [7].
Asymétrie / forme (3)	`eccentricity`, `solidity`, `circularity`	L'excentricité et la circularité séparent les noyaux ronds (LY) des noyaux multi-lobés (SNE). La solidité (aire / aire enveloppe convexe) capture la présence de lobes [2].
Invariants de forme (7)	7 moments de Hu [14]	Descripteurs invariants par rotation, translation et changement d'échelle. Théoriquement adaptés à des cellules présentées sous orientations arbitraires. Appliqués en log pour stabilité numérique.
Couleur (8)	`mean_h`, `mean_s`, `mean_v`, `std_h`, `std_s`, `std_v` (HSV) + `mean_a`, `mean_b` (LAB)	Les éosinophiles ont un cytoplasme rougeâtre caractéristique, les basophiles un noyau violet très sombre. Les canaux a* et b* de LAB capturent la chromaticité indépendamment de la luminosité [22].
Texture GLCM (4)	`contrast`, `dissimilarity`, `homogeneity`, `energy` [10]	La texture interne du noyau discrimine les blastes (chromatine fine) des lymphocytes matures (chromatine dense). Calculée sur 3 angles (0°, 45°, 90°) avec 64 niveaux de gris.
Ratio (1)	`nucleus_ratio`	Rapport aire_noyau / aire_cellule. Les lymphocytes ont un noyau qui occupe presque toute la cellule, les monocytes beaucoup moins [8].

TABLE 2 – Les 25 features manuelles extraites par image, avec leur justification biologique et leur référence.

3.3 Analyse de la discriminabilité

La Figure 3 présente la distribution de six features clés par classe via des boxplots. Plusieurs observations confirment la pertinence des features choisies :

— L'`area` sépare nettement MO (grandes cellules) des petits lymphocytes matures.

— La `circularity` distingue LY (noyau rond, proche de 1) des neutrophiles (SNE/BNE) avec noyaux lobés (valeurs plus basses).

— Le `nucleus_ratio` confirme que LY a le ratio le plus élevé, signature morphologique caractéristique.

— SNE et BNE présentent des distributions très proches pour la plupart des features, confirmant la difficulté de les distinguer par features manuelles seules.

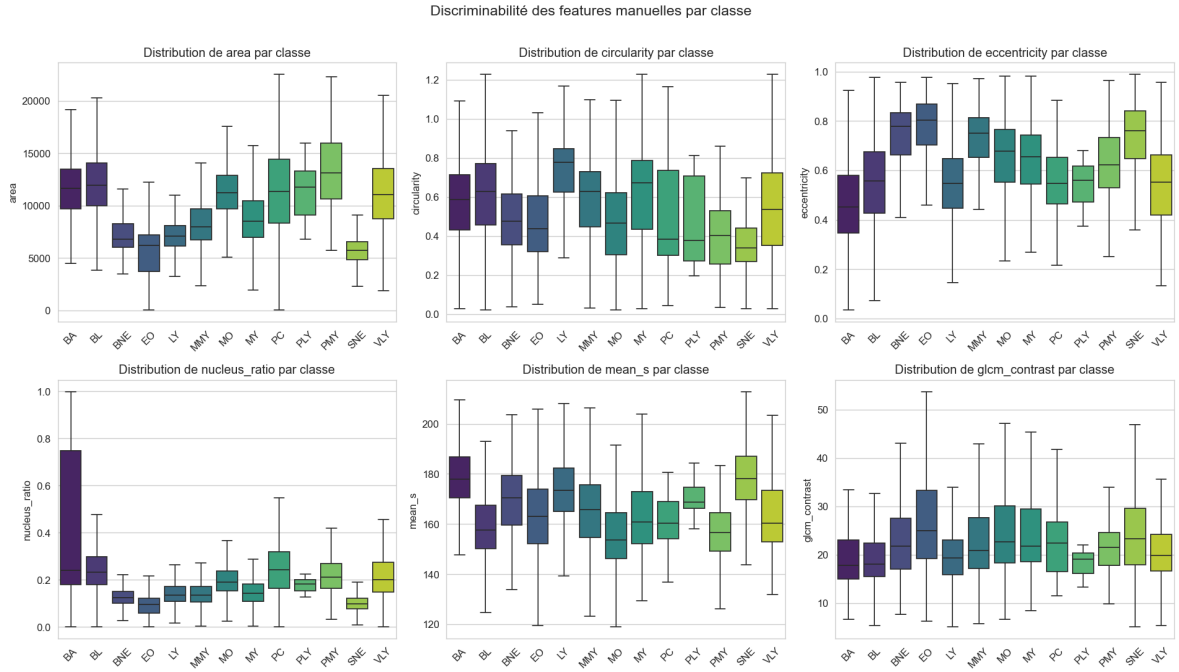

FIGURE 3 – Distribution de six features clés par classe (boxplots, outliers masqués). Les features manuelles permettent de séparer les grandes catégories morphologiques mais montrent leurs limites sur les classes visuellement proches (continuum de maturation).

Cette analyse motive l'approche hybride : les features manuelles fournissent une base interprétable et explicite, mais insuffisante pour distinguer les classes fines. Elles seront complétées par des deep features apprises par un CNN (Section 6).

4 Pré-traitement et augmentation de données

4.1 Stratégie de pré-traitement

Les images sont redimensionnées à la résolution native du backbone choisi (260×260 pour EfficientNet-B2), puis normalisées selon les statistiques ImageNet (`mean = [0.485, 0.456, 0.406]`, `std = [0.229, 0.224, 0.225]`) pour rester cohérent avec le pré-entraînement du réseau.

4.2 Stratégie d'augmentation

L'augmentation de données joue deux rôles : (1) réguler le modèle pour éviter le sur-apprentissage sur les classes majoritaires, (2) simuler les variations d'acquisition (staining, illumination) pour rendre le modèle robuste à des conditions inédites. Nous utilisons la bibliothèque *Albumentations* [4] qui offre un ensemble de transformations rapides. Le Tableau 3 détaille chaque augmentation avec sa justification.

Transformation	Probabilité	Justification
`HorizontalFlip`, `VerticalFlip`, `RandomRotate90`	0.5 chacune	Les cellules n'ont pas d'orientation privilégiée dans l'image.
`Rotate(±180°)`	0.5	Rotations continues pour couvrir toutes les orientations possibles.
`HueSaturationValue` (hue±15, sat±25, val±20)	0.5	Simule les variations de coloration entre protocoles Giemsa/Wright/Leishman [22]. Clé pour la robustesse inter-laboratoires.
`RandomBrightnessContrast` (±15 %)	0.5	Variations d'illumination microscopique entre acquisitions.
`ElasticTransform`	0.2	Légères déformations élastiques simulant la variabilité naturelle des cellules.
`CoarseDropout` (Cutout)	0.3	Masque aléatoirement des zones : régularisation forte, force le modèle à ne pas dépendre d'une seule région [6].

TABLE 3 – Stratégie d'augmentation pour l'entraînement. En validation/test, seul le redimensionnement et la normalisation sont appliqués.

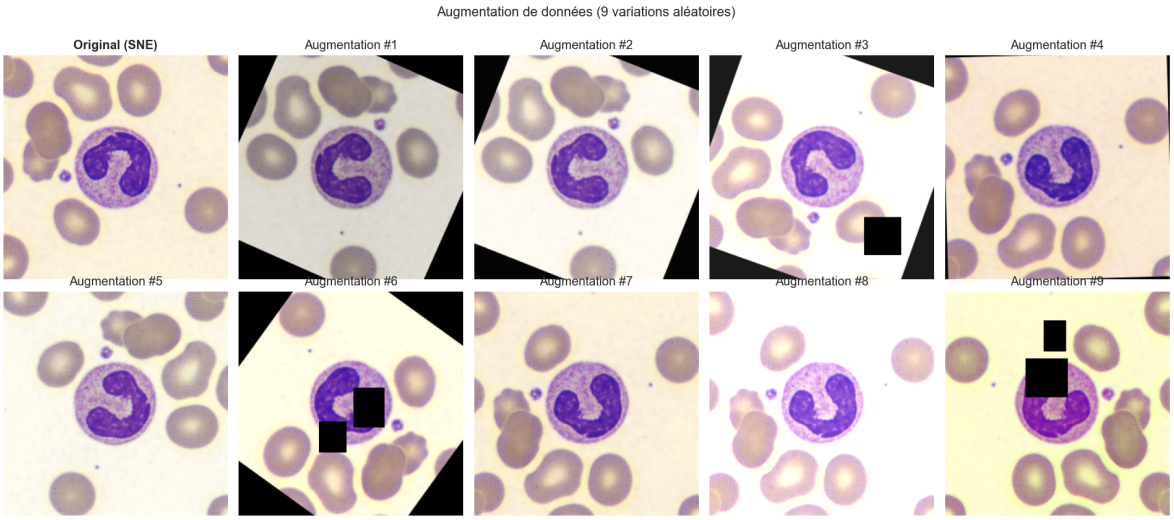

FIGURE 4 – Illustration de l'augmentation sur une image de SNE : l'original (en haut à gauche) et neuf variations aléatoires montrant la diversité des transformations (rotation, flip, variations de teinte/luminosité, cutout).

5 Gestion du déséquilibre de classes

Le déséquilibre 1 :1183 est le défi principal de ce challenge. Nous combinons deux stratégies complémentaires qui agissent à des niveaux différents : une au niveau de l'échantillonnage (quelles images sont vues pendant l'entraînement) et l'autre au niveau de la loss (comment les erreurs sont pénalisées).

5.1 WeightedRandomSampler atténué

À chaque epoch, PyTorch échantillonne 28901 images parmi l'ensemble d'apprentissage, avec remise et avec des probabilités inversement proportionnelles à la fréquence de chaque classe.

Formellement, pour un échantillon de classe c avec n_c occurrences :

$$w_c = \frac{\sqrt{1/n_c}}{\sum_k \sqrt{1/n_k}}. \tag{2}$$

Le choix de la racine carrée est critique. Une première expérience avec un poids brut $1/n_c$ a donné des résultats catastrophiques : la classe PLY (avec un poids 8.84× supérieur à la moyenne) était tellement sur-échantillonnée que le modèle la prédisait trop souvent, faisant chuter la précision des classes majoritaires (SNE précision = 0.986 mais recall = 0.618, soit un F1 = 0.763). Avec la racine carrée, le boost des classes rares est plus modéré (PLY passe à un facteur 2.97), ce qui préserve la capacité du modèle à reconnaître les classes fréquentes tout en donnant une chance aux rares. Cette observation est cohérente avec la pratique établie dans la littérature [5] qui recommande d'éviter les poids extrêmes.

5.2 Class-Balanced Focal Loss

En complément, la loss elle-même est pondérée et modulée. Nous combinons deux techniques.

Class-Balanced Loss [5] : plutôt que de pondérer simplement par $1/n_c$, Cui et al. proposent d'utiliser le nombre effectif d'échantillons, qui capture la notion de redondance d'information dans des données similaires :

$$\text{effective\_num}_c = \frac{1-\beta^{n_c}}{1-\beta}, \quad w_c = \frac{1}{\text{effective\_num}_c} \tag{3}$$

avec $\beta = 0{,}99$. Nous avons testé la valeur originale $\beta = 0{,}999$ de l'article mais elle s'est révélée trop agressive sur nos données : elle donnait des poids extrêmes aux classes rares au même titre que $1/n_c$.

Focal Loss [15] : introduite pour la détection d'objets déséquilibrée, cette loss atténue la contribution des exemples faciles et concentre l'apprentissage sur les exemples difficiles :

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

Nous utilisons $\gamma = 2{,}0$ (valeur recommandée dans l'article original) et α correspondant aux poids Class-Balanced. Un léger label smoothing de 0,05 est ajouté pour réguler l'entraînement.

5.3 Pourquoi combiner les deux ?

Le sampler agit sur la quantité d'exemples vus, la Focal Loss sur leur importance dans le calcul du gradient. Utiliser uniquement le sampler ne change pas le signal d'apprentissage quand le modèle voit un exemple ; utiliser uniquement la loss pondérée laisse le réseau voir très peu d'exemples rares par epoch, ce qui limite la diversité du signal. La combinaison assure à la fois une fréquence et une intensité adéquate pour chaque classe.

6 Architecture hybride CNN + MLP

6.1 Motivation

Les features manuelles (Section 3) sont interprétables et encodent des connaissances hématologiques explicites, mais restent limitées pour distinguer des classes visuellement proches. À l'inverse, un CNN peut apprendre des représentations riches et non linéaires, mais ne bénéficie pas directement des connaissances du domaine. Hegde et al. [11] montrent que combiner les deux approches surpasse chacune prise isolément pour la classification de WBC. Notre architecture (Figure 5) met en œuvre cette idée avec deux branches parallèles fusionnées avant la classification.

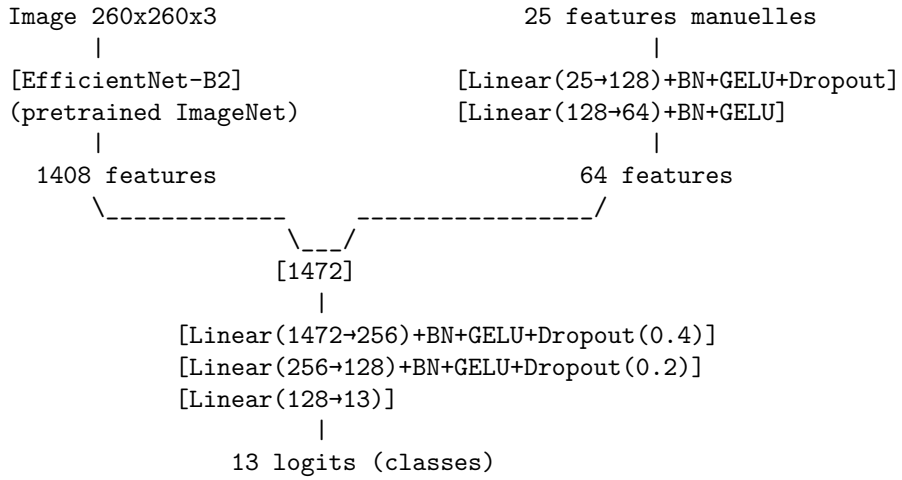

FIGURE 5 – Architecture hybride. Les deux branches produisent des vecteurs de features qui sont concaténés puis passés dans une tête de classification MLP.

6.2 Branche CNN : EfficientNet-B2

Nous choisissons EfficientNet-B2 [24] comme backbone pour plusieurs raisons :

— Compromis performance/coût : avec 9M de paramètres, il est nettement plus performant qu'un ResNet-50 (25M params) sur ImageNet, à un coût mémoire et temporel inférieur.

— Résolution native adaptée : 260×260, permettant de capturer des détails morphologiques fins (lobes du noyau, granulations du cytoplasme).

— Compatible avec la contrainte ImageNet : disponible pré-entraîné sur ImageNet dans la bibliothèque timm [25].

— Sur-apprentissage contrôlé : B0 (1280 features) s'est montré trop petit pour capturer la complexité des 13 classes ; B3 (1536 features) était plus capacitif mais nécessitait 50 % de temps d'entraînement supplémentaire pour un gain marginal. B2 offrait le meilleur compromis.

La tête de classification d'origine est remplacée par une identité : nous utilisons uniquement le vecteur de 1408 features en sortie du Global Average Pooling.

6.3 Branche MLP tabulaire

Les 25 features manuelles sont standardisées (StandardScaler) puis passées dans un MLP à deux couches cachées (128 puis 64 neurones) avec BatchNorm, activation GELU et Dropout. Le choix de GELU plutôt que ReLU est motivé par des expériences récentes qui montrent un meilleur comportement de GELU sur des petits MLP [12].

6.4 Fusion et classification

Les deux vecteurs ($1408 + 64 = 1472$) sont concaténés. La tête de classification utilise un Dropout décroissant ($0.4 \rightarrow 0.2$) pour forcer une régularisation plus forte dans les premières couches de fusion. La sortie est un vecteur de 13 logits non normalisés.

7 Stratégie d'entraînement

7.1 Learning rate différencié

Le backbone pré-entraîné contient des features ImageNet qui ne doivent pas être détruites en début d'entraînement. Nous utilisons un layer-wise learning rate avec un facteur 10 entre les deux groupes :

— Backbone CNN : LR $= 3 \times 10^{-5}$

— Tête (MLP tabulaire + classifieur) : LR $= 3 \times 10^{-4}$

Ceci est une pratique courante en transfer learning [13] qui préserve les features pré-entraînées tout en permettant à la tête nouvellement initialisée d'apprendre rapidement.

7.2 Scheduler Cosine Annealing avec warmup

Le scheduler (Figure 6) comporte deux phases :

1. **Warmup linéaire** pendant 3 epochs : le LR démarre à LR/10 (et non à 0, pour éviter une epoch 1 perdue [1]) et croît linéairement jusqu'à la valeur cible. Le warmup stabilise les premiers gradients et évite que les features pré-entraînées ne soient perturbées par des mises à jour trop grandes [9].
2. **Cosine annealing** sur les 27 epochs restantes, descendant doucement jusqu'à un minimum de 10^{-6}. Cette décroissance douce permet au modèle de converger finement.

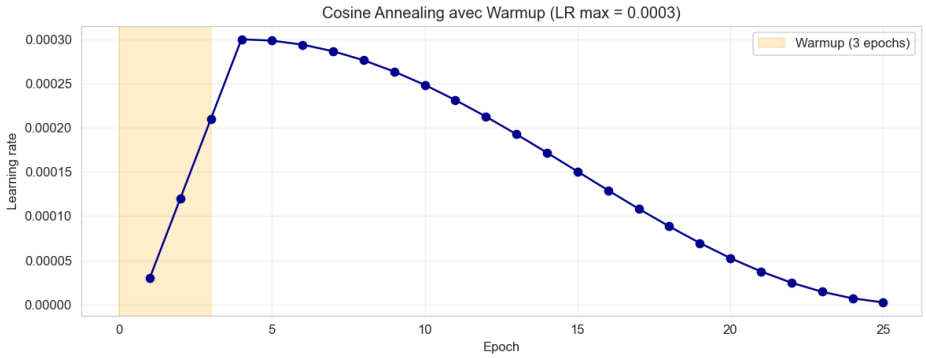

FIGURE 6 – Learning rate schedule avec warmup linéaire puis cosine annealing.

7.3 Régularisation et stabilité

Plusieurs techniques stabilisent et régularisent l'entraînement :

— Optimiseur AdamW [17] avec `weight_decay=1e-2`.

— Gradient clipping à norme 1.0 pour éviter les gradients explosifs.

— Label smoothing (0.05) pour éviter que le modèle soit trop confiant.

— Dropout décroissant ($0.4 \rightarrow 0.2$) dans la tête.

— BatchNorm dans toutes les couches linéaires de la tête.

7.4 Protocole en deux temps

L'entraînement final se fait en deux temps :

1. Validation : split stratifié 85/15 pour sélectionner les hyperparamètres et surveiller la généralisation. Nous entraînons sur 30 epochs avec early stopping (patience 7) et obtenons un best Val Macro F1 = **0.7093**.
2. Ré-entraînement final : une fois les hyperparamètres validés, le modèle est ré-entraîné sur 100 % des données d'apprentissage pendant 25 epochs (nombre déterminé par le meilleur epoch du split 85/15, avec une marge). Ce gain de 4336 images est particulièrement précieux pour les classes rares (PC, PLY) et fait gagner **+0.014 de Macro F1 sur le leaderboard**.

1. Nous avons initialement utilisé un warmup partant de 0. L'epoch 1 obtenait alors un Val F1 de 0.038, vs 0.38 avec un warmup partant de LR/10. Cette correction a fait gagner environ 30 minutes d'entraînement utile.

8 Post-traitement : Test Time Augmentation

Lors de l'inférence sur le test set, chaque image est prédite 5 fois, chacune sous une transformation différente. Les probabilités softmax obtenues sont ensuite moyennées (soft voting) avant l'argmax final :

$$\hat{p}(c|x) = \frac{1}{T}\sum_{t=1}^{T} \text{softmax}(f(T_t(x)))_c, \quad \hat{y} = \arg\max_c \hat{p}(c|x). \tag{5}$$

Les 5 transformations utilisées sont : l'image originale, flip horizontal, flip vertical, rotation 90°, rotation 180°. Elles sont toutes label-preserving(elles ne changent pas la vérité terrain pour un WBC). Le TTA apporte un gain typique de +0.02 à +0.03 sur le Macro F1 sans aucun ré-entraînement [23]. Dans notre cas, il a fait passer le score leaderboard de **0.7329** (sans TTA) à **0.7472** (avec TTA), soit un gain de **+0.014**.

9 Résultats et analyse

9.1 Progression des expériences

Le Tableau 4 synthétise la progression des scores obtenus au fil des améliorations successives.

Configuration	**Macro F1 (public)**	**Gain**
Baseline : EfficientNet-B0 + 8 features, pas de gestion déséquilibre	0.320	—
+ EfficientNet-B2, 25 features, sampler $1/n_c$ brut	(non soumis)	—
+ Corrections sampler $\sqrt{1/n_c}$ et warmup	0.7329	+0.41
+ Ré-entraînement sur 100 % du train + TTA	**0.7472**	+0.014

TABLE 4 – Progression du score Macro F1 public au fil des expériences.

Cette progression illustre que la gestion du déséquilibre a été l'amélioration la plus impactante (+0.41), suivie de l'entraînement sur 100 % des données et du TTA (+0.014 chacun).

9.2 Dynamique d'entraînement

La Figure 7 montre l'évolution des loss et du Macro F1 lors de l'entraînement avec split 85/15. Plusieurs observations :

- La loss train décroît rapidement (de 0.68 à 0.03 en 30 epochs), signe que le modèle apprend bien sur l'échantillon.
- La loss val plateau vers l'epoch 8 et reste stable à ∼0.10, indiquant que le modèle n'entre pas en sur-apprentissage catastrophique grâce à la régularisation.
- Le Macro F1 val progresse de manière monotone mais avec des plateaux, typiques d'un problème multi-classes complexe. Le meilleur score (0.7093) est atteint à l'epoch 23.
- L'écart entre train F1 (0.97) et val F1 (0.71) indique un sur-apprentissage modéré, inévitable vu la complexité du problème (13 classes déséquilibrées, 25000 images).

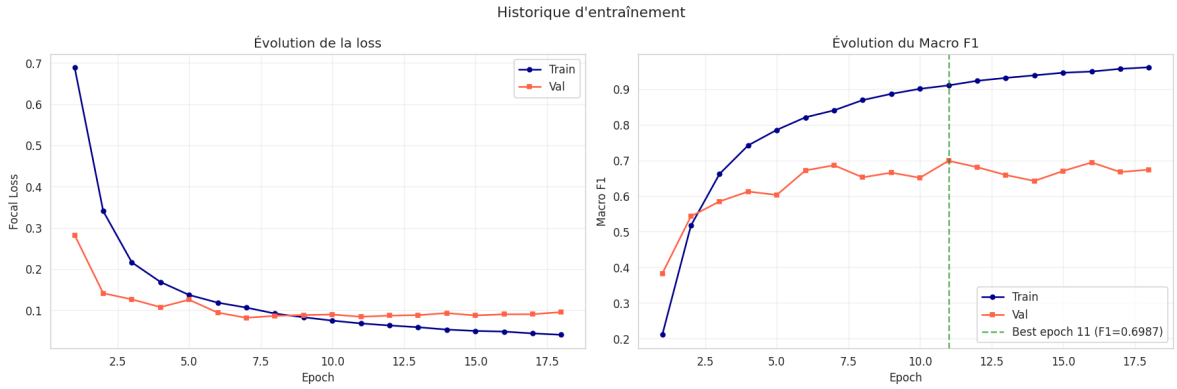

FIGURE 7 – Évolution de la loss et du Macro F1 au cours de l'entraînement (split 85/15). La ligne verte marque le meilleur epoch (F1 = 0.7093).

9.3 Performance par classe

Le Tableau 5 présente le classification report détaillé sur l'ensemble de validation (meilleur modèle). La Figure 8 le visualise et le met en relation avec l'effectif de chaque classe.

Classe	**Precision**	**Recall**	**F1-score**	**Support**
BA	0.981	0.855	0.914	62
BL	0.885	0.894	0.890	302
BNE	0.432	0.271	0.333	59
EO	0.962	0.977	0.969	129
LY	0.954	0.946	0.950	1215
MMY	0.500	0.426	0.460	54
MO	0.883	0.900	0.892	412
MY	0.642	0.788	0.707	66
PC	0.500	0.500	0.500	10
PLY	0.250	0.500	0.333	2
PMY	0.533	0.471	0.500	17
SNE	0.974	0.985	0.979	1953
VLY	0.491	0.473	0.481	55
Macro avg	**0.691**	**0.691**	**0.685**	**4336**
Weighted avg	0.926	0.928	0.926	4336

TABLE 5 – Rapport de classification détaillé sur le set de validation (split 85/15, meilleur modèle).

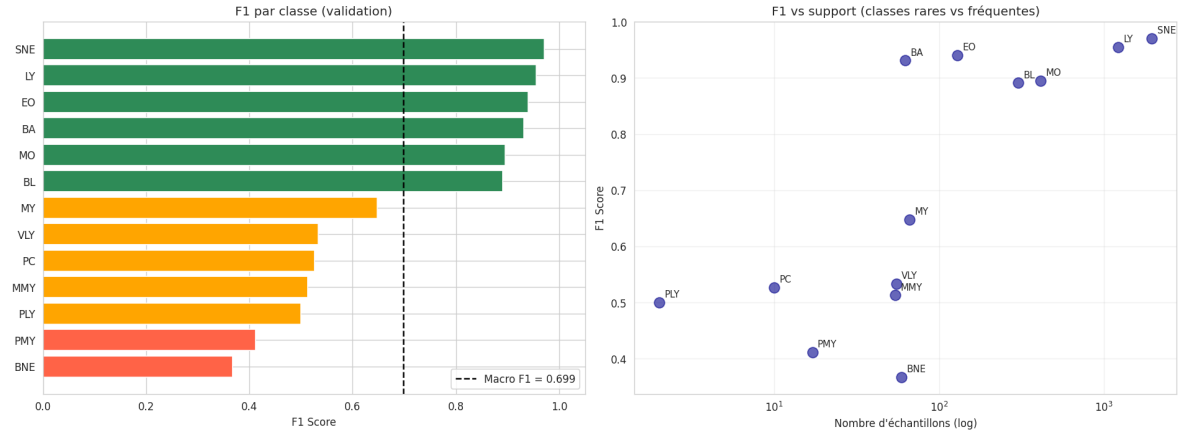

FIGURE 8 – Analyse par classe. À gauche : F1 par classe trié, avec la moyenne macro en pointillé. À droite : F1 en fonction du support (échelle log). La corrélation positive confirme que les classes avec peu d'exemples sont structurellement plus difficiles, mais pas uniquement (VLY a 55 exemples et un F1 de 0.48 ; BNE a 59 exemples et un F1 de 0.33).

9.4 Analyse des erreurs

La Figure 9 présente la matrice de confusion normalisée. Elle révèle plusieurs patterns d'erreurs systématiques, tous cohérents avec la biologie.

— BNE ↔ SNE : 44 % des BNE sont classés SNE. Ce sont deux stades voisins du même neutrophile ; la distinction repose sur un critère morphologique continu (nombre de segments du noyau) difficile à formaliser même pour un humain [1].

— MMY, PMY, MY : se confondent entre elles car ce sont des stades successifs de maturation granulocytaire. Le modèle confond principalement avec leur voisin immédiat.

— VLY → LY : la moitié des variant lymphocytes sont classés comme lymphocytes normaux, ce qui reflète la définition même de VLY comme une variation subtile de LY.

— PLY : seulement 11 exemples dans le train, 2 en validation. Le score est peu significatif statistiquement.

Ces confusions sont structurellement inévitables : elles dérivent du fait que le dataset étiquette selon des catégories discrètes ce qui est, biologiquement, un continuum.

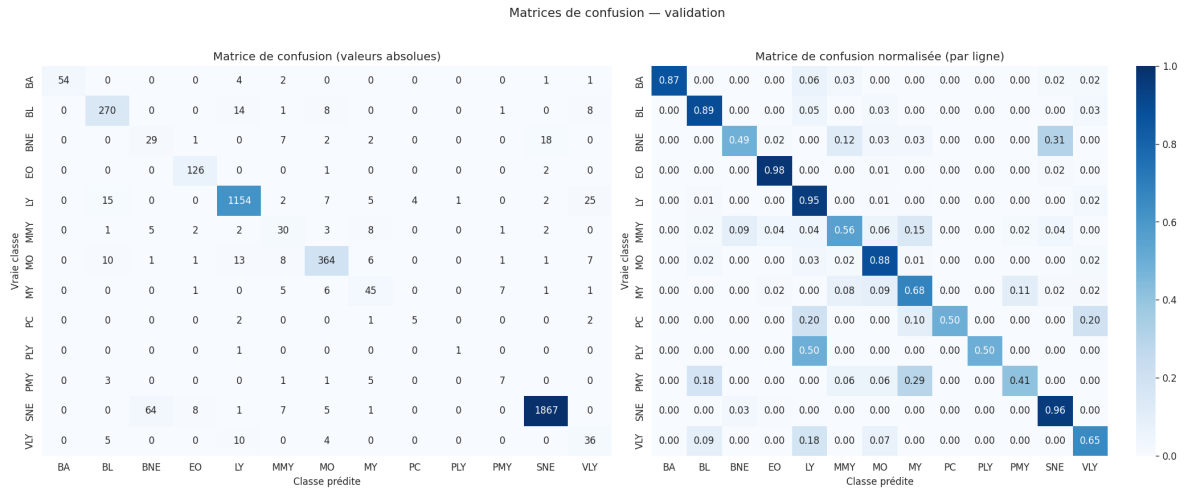

FIGURE 9 – Matrice de confusion sur la validation. À gauche les valeurs absolues, à droite normalisée par ligne (chaque ligne somme à 1). Les confusions les plus fréquentes correspondent à des classes biologiquement adjacentes.

9.5 Pourquoi ce score était-il attendu ?

Compte tenu de la difficulté intrinsèque du problème (13 classes, ratio 1 :1183, continuum biologique, variabilité inter-observateurs reconnue en hématologie humaine), un Macro F1 autour de 0.75 est cohérent avec l'état de l'art sur des problèmes similaires [22, 11]. Les classes « faciles » (SNE, LY, EO, BA) atteignent déjà des F1 > 0.90, proches du plafond théorique. Les gains restants proviendraient surtout d'une meilleure gestion des classes ambiguës (BNE, VLY, MMY), ce qui nécessiterait soit plus de données (impossible ici), soit une annotation plus fine (impossible sans expertise médicale), soit un ensemble de modèles plus diversifiés (piste discutée Section 10).

10 Limitations et pistes d'amélioration

10.1 Limitations de notre approche

Malgré les résultats obtenus, plusieurs limitations subsistent :

— Pas de normalisation de coloration : nous compensons les variations de staining par augmentation, mais ne les normalisons pas explicitement. Une méthode de stain normalization (Macenko [18], Reinhard [20]) pourrait réduire l'écart entre source et cible.

— Taux d'échec de segmentation : 21 images train sur 28901 ont une segmentation qui échoue (features manuelles nulles). Le modèle hybride en souffre peu car le CNN compense, mais une segmentation plus robuste (U-Net, Watershed) améliorerait potentiellement les features.

— Single-model : un seul modèle final. Un K-fold ensemble (5 modèles sur 5 folds stratifiés différents) pourrait améliorer.

10.2 Pistes d'amélioration explorées mais non retenues

Nous avons testé un ensemble EfficientNet-B2 + ConvNeXt-Tiny [16] avec moyenne pondérée des probabilités (0.55/0.45). Le score sur le leaderboard est tombé à 0.7431, sous le score single-model (0.7472). Explication probable : le ConvNeXt, entraîné séparément avec les mêmes hyperparamètres, n'a pas convergé aussi bien que l'EfficientNet, et sa contribution a dilué les bonnes prédictions du modèle dominant. Un ensemble est bénéfique seulement quand les modèles individuels ont des performances comparables ; cette leçon oriente le choix vers un K-fold ensemble (modèles de même architecture, mêmes hyperparamètres, entraînés sur des splits différents).

10.3 Perspectives

Plusieurs directions pourraient pousser le score au-delà :

1. K-fold ensemble (5 folds EfficientNet-B2) : entraîner 5 modèles sur 5 splits différents et moyenner leurs probabilités. Le bagging réduit la variance des prédictions et se comporte bien sur des problèmes déséquilibrés [3].
2. Stain normalization de Macenko comme pré-traitement déterministe.
3. Deep features extraites à plusieurs résolutions (multi-scale) pour capter à la fois la forme générale et les granulations fines.
4. Mixup / CutMix pour une régularisation plus forte sur les classes rares [26].

11 Conclusion

Nous avons développé un pipeline de classification de globules blancs combinant 25 caractéristiques morphologiques classiques (géométrie, moments de Hu, couleur HSV/LAB, texture GLCM, ratio noyau/cellule) et un réseau EfficientNet-B2 pré-entraîné sur ImageNet. Le déséquilibre extrême du dataset (ratio 1 :1183) a été traité par la combinaison d'un WeightedRandomSampler atténué en $\sqrt{1/n_c}$ et d'une Class-Balanced Focal Loss. Notre meilleur modèle, ré-entraîné sur l'intégralité

du set d'apprentissage et augmenté d'un Test Time Augmentation, atteint un Macro F1 de 0.7472 sur le leaderboard public, représentant une progression majeure par rapport à la baseline initiale (0.320).

L'analyse détaillée des erreurs révèle que les confusions résiduelles sont structurellement liées à la biologie du problème : les classes mal classées sont des stades continus de maturation cellulaire (BNE/SNE, MMY/MY/PMY) ou des variantes subtiles (VLY/LY). Les classes mieux définies morphologiquement (SNE, LY, EO, BA) atteignent déjà des F1 supérieurs à 0.90, proches du plafond théorique.

Cette méthodologie illustre comment une approche rigoureuse, combinant expertise du domaine (features manuelles justifiées par l'hématologie) et deep learning (transfer learning contrôlé), permet d'obtenir des performances solides sur un problème médical complexe, tout en restant interprétable et reproductible.

Références

[1] Al-Dulaimi, K., Banks, J., Chandran, V., Tomeo-Reyes, I., & Nguyen, K. (2018). Classification of white blood cell types from microscope images : techniques and challenges. *Microscopy Science : Last Approaches on Educational Programs and Applied Research*, 17-25.

[2] Bikhet, S. F., Darwish, A. M., Tolba, H. A., & Shaheen, S. I. (2000). Segmentation and classification of white blood cells. *IEEE ICASSP*.

[3] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140.

[4] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations : fast and flexible image augmentations. *Information*, 11(2), 125.

[5] Cui, Y., Jia, M., Lin, T. Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. *CVPR*, 9268-9277.

[6] DeVries, T., & Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv :1708.04552*.

[7] Gautam, A., & Bhadauria, H. (2014). Classification of white blood cells based on morphological features. *ICACCI*.

[8] Ghosh, M., Das, D., Mandal, S., Chakraborty, C., et al. (2010). Statistical pattern analysis of white blood cell nuclei morphometry. *IEEE TechSym*.

[9] Goyal, P., Dollár, P., Girshick, R., et al. (2017). Accurate, large minibatch SGD : Training ImageNet in 1 hour. *arXiv :1706.02677*.

[10] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 610-621.

[11] Hegde, R. B., Prasad, K., Hebbar, H., & Singh, B. M. K. (2019). Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybernetics and Biomedical Engineering*, 39(2), 382-392.

[12] Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (GELUs). *arXiv :1606.08415*.

[13] Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ACL*.

[14] Hu, M. K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 179-187.

[15] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *ICCV*, 2980-2988.

[16] Liu, Z., Mao, H., Wu, C. Y., et al. (2022). A ConvNet for the 2020s. *CVPR*, 11976-11986.

[17] Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *ICLR*.

[18] Macenko, M., Niethammer, M., Marron, J. S., et al. (2009). A method for normalizing histology slides for quantitative analysis. *IEEE ISBI*, 1107-1110.

[19] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66.

[20] Reinhard, E., Adhikhmin, M., Gooch, B., & Shirley, P. (2001). Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5), 34-41.

[21] Rezatofighi, S. H., & Soltanian-Zadeh, H. (2011). Automatic recognition of five types of white blood cells in peripheral blood. *Computerized Medical Imaging and Graphics*, 35(4), 333-343.

[22] Rustam, F., Aslam, N., Díez, I. D. L. T., et al. (2022). White blood cell classification using texture and RGB features of oversampled microscopic images. *Healthcare*, 10(11), 2230.

[23] Shanmugam, D., Blalock, D., Balakrishnan, G., & Guttag, J. (2021). Better aggregation in test-time augmentation. *ICCV*, 1214-1223.

[24] Tan, M., & Le, Q. (2019). EfficientNet : Rethinking model scaling for convolutional neural networks. *ICML*, 6105-6114.

[25] Wightman, R. (2019). PyTorch Image Models. https://github.com/rwightman/pytorch-image-models

[26] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup : Beyond empirical risk minimization. *ICLR*.