

# RAPPORT DE GROUPE EN SCIENCES DES DONNÉES 4

COULIBALY Maïmouna, MEKKI Ryan



Département MIASHS, UFR 6 Informatique, Mathématique et Statistique  
Université Paul Valéry, Montpellier 3

Avril 2025

SOUMIS COMME CONTRIBUTION PARTIELLE  
POUR LE COURS SCIENCE DES DONNÉES 4

---

## Déclaration de non plagiat

---

Nous déclarons que ce rapport est le fruit de notre seul travail, à part lorsque cela est indiqué explicitement.

Nous acceptons que la personne évaluant ce rapport puisse, pour les besoins de cette évaluation:

- la reproduire et en fournir une copie à un autre membre de l'université; et/ou,
- en communiquer une copie à un service en ligne de détection de plagiat (qui pourra en retenir une copie pour les besoins d'évaluation future).

Nous certifions que nous avons lu et compris les règles ci-dessus.

En signant cette déclaration, nous acceptons ce qui précède.

Signature:



Date: 11/04/2025

Signature:



Date: 11/04/2025

---

## Remerciements

---

Nos plus sincères remerciements vont à nos encadrants pédagogiques pour les conseils avisés sur notre travail.

Nous remercions aussi Madame BRINGAY Sandra, Madame PATEL Namrata et Monsieur MICHEL-PICQUE Théodor.

16/04/2025.

---

## Résumé

---

La classification des images par reconnaissance d'images est un domaine clé en science des données car elle peut être appliquée de l'industriel au biomédical et de la théorie à la pratique. Les dernières années ont introduit une nouvelle ère de progrès dans ce domaine, encourageant les avancées dans les approches d'apprentissage profond, particulièrement illustrées dans le domaine de l'imagerie médicale, en permettant la découverte de motifs complexes invisibles à l'œil humain. Dans ce sens, l'analyse automatique des cellules sanguines par imagerie s'avère être un défi clé pour obtenir un diagnostic rapide et efficace dans l'analyse médicale. Par exemple, les globules blancs peuvent appartenir à plusieurs catégories, où leur répartition est reconnue comme un signal clé de l'état du patient. Comme de nombreuses publications ont déjà mis en œuvre des réseaux de convolution pour cette tâche, sachant qu'ils gèrent cette tâche efficacement, le contraste rigoureux entre les types de réseaux et l'analyse complète des considérations derrière la mauvaise classification dans de nombreux cas manquent. Par conséquent, notre objectif est de combler cette lacune en réalisant la tâche de classification supervisée organisée avec un ensemble de données majoritaire de cellules et d'effectuer les réseaux de convolution conçus en comparaison avec l'architecture bien connue ResNet-18 et les conceptions classiques, les arbres aléatoires et les perceptrons multicouches à comparer également. Des modifications spéciales pour arrêter automatiquement l'entraînement lorsque la perte cesse de diminuer ont permis d'équilibrer efficacité et bonne généralisation. En conséquence, nous atteignons une précision de 84,76 % aux tests pour notre réseau entraîné à partir de zéro et légèrement meilleure, 85,65 % pour la combinaison de ResNet-18 et de la classification SVM. La précision de validation est de 100 %, mais l'exactitude auto-test, légèrement moins bonne, confirme cette précision. Cette découverte est prouvée par la représentation graphique de la matrice de confusion, les travaux d'apprentissage et l'estimation de la classification. Ces conclusions illustrent comment même les modèles les plus simples peuvent rivaliser efficacement avec des conceptions préétablies plus complexes lorsque les capacités de l'ensemble de données sont limitées, révélant la stratégie potentiellement développée de combinaison des concepts d'apprentissage automatique. Ce projet ouvre donc un outil simple mais efficace pour aider à analyser les micrographies par les biologistes ou les médecins tout en fournissant des pistes pour des explications et une analyse de précision plus approfondies.

---

## Table des matières

---

Chapitre 1	Introduction	1
Chapitre 2	Gestion de projet	2
2.1	Introduction	2
2.2	Planification	2
2.3	Exploration et préparation des données	2
2.4	Sélection des modèles et entraînement	3
2.5	Évaluation des modèles et analyse des résultats	3
2.6	Suivi et gestion des ajustements	3
2.7	Contribution	3
Chapitre 3	Présentation des données et expérimentations	5
3.1	Introduction	5
3.2	prétraitement	5
3.2.1	Augmentation de données	6
3.2.2	Gestion du déséquilibre des classes	6
3.2.3	Création d'un ensemble de validation	7
3.2.4	Préparation pour les modèles MLP et CNN	7
3.3	Conclusion	7
Chapitre 4	Analyse et interprétation	8
4.1	Introduction	8
4.2	Modèles	9
4.2.1	CNN	9
4.2.2	RESNET18	10
4.2.3	MLP	12
4.2.4	Conclusion	13
4.3	Analyse des classificateurs	14
4.3.1	Classificateurs du CNN	14
4.3.2	Classificateurs du RESNET18	17
4.3.3	Classificateurs du MLP	20
4.3.4	Conclusion	22
Chapitre 5	Conclusions et perspectives	24
	Bibliographie	26

---

# CHAPITRE 1

## Introduction

---

Dans le cadre de ce projet, nous avons exploré et mis en œuvre des modèles de machine learning et de deep learning pour la classification des cellules sanguines. Nous avons choisi de travailler avec un jeu de données trouvé sur Kaggle, qui contient des images de cellules sanguines augmentées, accompagnées d'étiquettes indiquant quatre types de cellules différentes : éosinophiles, lymphocytes, monocytes et neutrophiles. En utilisant cet ensemble de données, nous avons exploré différentes techniques pour entraîner des modèles capables de classer les cellules sanguines. Ce processus a impliqué l'utilisation d'outils d'analyse, tels que les matrices de confusion, et l'interprétation approfondie des résultats obtenus, en analysant les performances du modèle ainsi que les facteurs influençant ses prédictions. Le projet vise à appliquer des méthodes d'intelligence artificielle à la classification d'images médicales. Il a pour but de démontrer notre capacité à réaliser une analyse approfondie des données en utilisant des techniques de machine learning et de deep learning pour classer des images de cellules sanguines, faire des prédictions et interpréter les résultats obtenus à l'aide d'outils d'évaluation comme les matrices de confusion. Ce projet est principalement destiné à un cadre académique et n'a pas pour objectif immédiat de servir directement aux professionnels de la santé ou aux chercheurs. L'originalité de ce projet réside dans l'application de techniques d'intelligence artificielle, notamment l'apprentissage profond, pour automatiser la classification des cellules sanguines à partir d'images numériques. En testant divers modèles de deep learning, tels que les réseaux de neurones convolutifs (CNN) et ResNet, nous avons cherché à optimiser cette classification. Par ailleurs, l'utilisation d'un jeu de données augmentées a permis d'améliorer les performances des modèles tout en réduisant le risque de sur-apprentissage, grâce à des techniques comme la normalisation et l'augmentation des données. Les vidéos de démonstration et de soutenance sont disponibles à cette URL :[https://pod.univ-montp3.fr/media/videos/9c51dbf59cb2bb2a34d7b1c38016a3a9ca545355e3f8e99fb831f8da453c0bf3/projet\\_final.mp4](https://pod.univ-montp3.fr/media/videos/9c51dbf59cb2bb2a34d7b1c38016a3a9ca545355e3f8e99fb831f8da453c0bf3/projet_final.mp4)

---

## CHAPITRE 2

### Gestion de projet

---

#### 2.1 Introduction

Ce chapitre décrit la gestion du projet dans son ensemble, en détaillant les différentes étapes que nous avons suivies pour mener à bien ce travail. Nous y aborderons la méthodologie que nous avons choisie, ainsi que les différentes contributions qui ont structuré notre démarche. Nous avons mis en place un planning rigoureux pour nous assurer du respect des délais tout en garantissant la qualité de nos résultats. Ce projet, de nature académique, a nécessité une gestion minutieuse des ressources, des outils et du temps. Nous commencerons par expliquer la méthodologie adoptée, puis nous passerons à une présentation des différentes contributions réalisées par chacun des membres du binôme. Le planning du projet, essentiel pour une progression harmonieuse, sera également détaillé. Enfin, ce chapitre se conclura par un bilan de la gestion du projet, soulignant les ajustements apportés au fur et à mesure de son avancement et les leçons tirées de l'expérience. Méthodologie : Dans le cadre de ce projet, la méthodologie adoptée a été divisée en plusieurs étapes, permettant d'assurer une approche structurée pour atteindre les objectifs définis. Ces étapes comprennent la planification du projet, l'exploration et le prétraitement des données (détaillés dans le chapitre dédié à la base de données), la sélection des modèles, l'entraînement des modèles et de leurs classificateurs, et l'évaluation des résultats obtenus.

#### 2.2 Planification

La gestion du projet a été essentielle pour garantir le respect des délais et une bonne organisation du travail. Un planning détaillé a été élaboré dès le début, avec des marques définissant les étapes majeures du projet : collecte des données, traitement des données, sélection et entraînement des modèles, et enfin, l'évaluation des performances des modèles. Un suivi régulier a permis d'ajuster le plan en fonction des défis rencontrés.

#### 2.3 Exploration et préparation des données

La base de données utilisée a été attentivement explorée pour comprendre sa structure et déterminer les actions nécessaires avant de pouvoir entraîner les modèles. Cela inclut la gestion de la qualité des données, le nettoyage des données, et le prétraitement des images. Pour cette étape, les tâches de prétraitement (détaillées dans le chapitre consacré à la base de données) ont été cruciales pour rendre les données prêtes pour l'entraînement, telles que la normalisation des images et l'augmentation des données afin d'éviter le risque de sur-apprentissage.

## 2.4 Sélection des modèles et entraînement

Une fois les données prêtes, plusieurs modèles de machine learning et deep learning ont été sélectionnés. En particulier, des réseaux de neurones convolutifs (CNN) et le modèle ResNet ont été utilisés pour la classification des cellules sanguines. L'entraînement des modèles s'est fait de manière itérative, avec des ajustements constants des hyperparamètres et une validation croisée pour optimiser les performances.

## 2.5 Évaluation des modèles et analyse des résultats

Après l'entraînement des modèles de deep learning tels que CNN et ResNet, une approche complémentaire a été mise en place pour évaluer la performance des modèles. Nous avons extrait les caractéristiques des images à partir des couches internes des réseaux, en utilisant les représentations apprises par les modèles après l'entraînement. Ces caractéristiques ont ensuite été utilisées comme entrée pour plusieurs classificateurs classiques, à savoir le k-Nearest Neighbors (k-NN), les Support Vector Machines (SVM), et la Random Forest. Cette méthode, basée sur l'extraction de caractéristiques et l'utilisation de classificateurs classiques, nous a permis d'évaluer si l'ajout de ces classificateurs pouvait améliorer la performance de la classification des cellules sanguines. Pour analyser les résultats, des métriques classiques de classification, telles que la précision et les matrices de confusion, ont été utilisées. Ces analyses ont permis de mieux comprendre les forces et les faiblesses de chaque modèle en fonction de la manière dont ils ont traité les caractéristiques extraites, et ont guidé les ajustements nécessaires pour améliorer les performances globales du système.

## 2.6 Suivi et gestion des ajustements

Tout au long du projet, un suivi constant a permis de s'assurer que les objectifs étaient atteints. Afin de rester organisés et de suivre l'avancement des différentes étapes, nous avons principalement échangé via WhatsApp, où nous avons partagé des idées et résolu des problèmes en temps réel. De plus, nous avons utilisé un onglet partagé sur nos Macs pour collaborer efficacement et organiser notre travail. Nous avons également discuté régulièrement du projet, tant pendant les heures de cours du module que lors de discussions informelles en dehors des cours, notamment pour traiter les défis rencontrés et ajuster notre approche en fonction des résultats obtenus. Ces échanges ont été essentiels pour assurer une bonne coordination et pour adapter notre travail au fur et à mesure de l'avancement du projet. En conclusion, cette méthodologie s'est appuyée sur une approche rigoureuse et structurée, intégrant des étapes d'exploration, de prétraitement, de modélisation et d'évaluation des résultats. Chaque phase a été conduite avec soin pour garantir une analyse précise et des modèles de qualité, et le suivi de l'avancement du projet a permis de répondre efficacement aux objectifs fixés.

## 2.7 Contribution

Ce projet a été réalisé en binôme, avec une répartition équilibrée des tâches et une collaboration continue à chaque étape. La recherche du jeu de données a été



effectuée conjointement, après plusieurs échanges et vérifications pour s'assurer de la pertinence du dataset. Afin de mieux appréhender les notions de deep learning, que nous n'avions pas encore étudiées en cours, nous avons également consacré du temps à consulter ensemble des ressources pédagogiques, notamment des vidéos explicatives sur YouTube.

- Le développement des modèles a été réparti comme suit :
  1. Ryan a conçu et implémenté le modèle CNN personnalisé.
  2. Maïmouna a développé les modèles ResNet18 et MLP.

L'idée d'exploiter des classificateurs traditionnels (kNN, SVM, Random Forest) sur les représentations extraites des modèles profonds a émergé au fil du projet, et a été mise en œuvre dans un second temps. Maïmouna a également proposé et intégré l'affichage de prédictions visuelles sur des images tests pour enrichir l'analyse finale.

- Pour la rédaction du rapport :
  1. Maïmouna a rédigé l'introduction (chapitre 1), la gestion de projet (chapitre 2) et l'analyse des résultats (chapitre 4).
  2. Ryan s'est chargé du résumé, de la présentation de la base de données et des modèles (chapitre 3), ainsi que de la conclusion (chapitre 5).

Chaque section a été relue et validée à deux, dans un esprit de relecture croisée et d'amélioration continue. L'ensemble du rapport a d'abord été rédigé sur Word, puis mis en forme en LaTeX par Ryan. Enfin, la présentation PowerPoint destinée à la soutenance a été conçue par Maïmouna. Cette collaboration a été marquée par une forte implication des deux membres, dans un souci permanent d'entraide et de cohérence du travail rendu.

---

## CHAPITRE 3

### Présentation des données et expérimentations

---

#### 3.1 Introduction

Dans le cadre de notre projet, nous avons travaillé sur la classification de différents types de cellules sanguines à partir d'images microscopiques, en utilisant un jeu de données riche et structuré provenant de la plateforme Kaggle. Le lien vers notre jeu de données : <https://www.kaggle.com/datasets/paultimothymooney/blood-cells>. Ce jeu de données, annoté et organisé de manière hiérarchique, contient des images de globules blancs réparties en quatre classes : éosinophiles, monocytes, neutrophiles et lymphocytes. Les images sont enregistrées au format .jpeg, avec une taille moyenne d'environ 100 à 150 Ko, ce qui garantit un bon équilibre entre qualité visuelle et rapidité de chargement. L'organisation des fichiers dans des sous-dossiers par classe rend ce dataset directement exploitable avec des outils comme ImageFolder de PyTorch, facilitant ainsi l'automatisation de l'étiquetage. Pour extraire les caractéristiques visuelles, nous avons utilisé trois types de modèles : un premier réseau de neurones convolutif (CNN) que nous avons construit spécifiquement pour notre tâche, le modèle pré-entraîné ResNet18, reconnu pour sa capacité à extraire des représentations profondes robustes, et un perceptron multi-couche (MLP) basé uniquement sur des couches entièrement connectées, afin d'explorer une approche non convolutive. Les caractéristiques extraites par ces modèles ont ensuite été exploitées par trois classificateurs classiques : k-Nearest Neighbors (kNN), Support Vector Machine (SVM) et Random Forest. Chacun a été choisi pour ses spécificités : kNN pour explorer la proximité dans l'espace des features (caractéristiques), SVM pour sa capacité à optimiser la séparation entre classes, et Random Forest pour sa robustesse aux données bruitées et sa capacité à gérer des relations complexes non linéaires. Cette combinaison de techniques issues à la fois du deep learning et du machine learning classique nous a permis d'explorer différentes stratégies de classification et d'analyser leur efficacité dans le cadre d'une tâche de reconnaissance d'images biomédicales.

#### 3.2 prétraitement

Avant l'importation de nos données, aucun nettoyage spécifique n'a été nécessaire. Le jeu de données fourni était déjà complet, bien structuré et organisé par classe dans des dossiers distincts (éosinophiles, lymphocytes, monocytes et neutrophiles). Il ne contenait ni doublons, ni valeurs manquantes, ni incohérences textuelles, car il s'agissait exclusivement d'images classées dans un format exploitable. Cependant, des prétraitements essentiels ont été réalisés après l'importation, afin de préparer

les données pour l'apprentissage automatique, en particulier pour les modèles CNN, MLP et ResNet18. Ces étapes ont été essentielles pour améliorer la performance des modèles et prévenir des problèmes courants comme le sur-apprentissage.

### 3.2.1 *Augmentation de données*

Afin d'améliorer la robustesse des modèles et simuler des conditions réelles plus variées, plusieurs transformations ont été appliquées aux images via la bibliothèque `torchvision.transforms`. Ces transformations incluent :

- le redimensionnement des images :
  1. À 128x128 pixels pour les modèles CNN et MLP.
  2. À 224x224 pixels pour ResNet18, conformément aux spécifications du modèle pré-entraîné.
- les rotations aléatoires (jusqu'à plus ou moins 15° ou plus ou moins 20° selon le modèle) pour simuler des variations d'angles.
- les flips horizontaux aléatoires, introduisant de la symétrie.
- le jitter de couleur (uniquement pour ResNet18) : variations aléatoires de la luminosité, contraste, saturation et teinte.
- la conversion en tenseurs pour le traitement par PyTorch.
- la normalisation des images :
  1. Moyenne = 0.5 et écart-type = 0.5 (CNN, MLP).
  2. Moyenne = [0.485, 0.456, 0.406] et écart-type = [0.229, 0.224, 0.225] (ResNet18, comme dans ImageNet).

Ces transformations sont appliquées dynamiquement à chaque chargement d'image via les `DataLoader`, ce qui permet d'introduire une variabilité à chaque époque.

### 3.2.2 *Gestion du déséquilibre des classes*

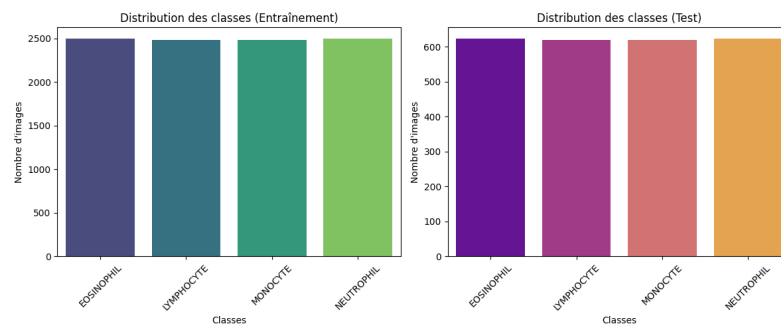


Figure 3.1: distribution des tâches

Une analyse préliminaire a révélé un déséquilibre entre les classes (certaines étant sur-représentées). Cela aurait pu conduire à un biais de prédiction. Pour y remédier : Dans le cas de ResNet18, nous avons utilisé des poids inversement proportionnels à la fréquence des classes dans la fonction de perte (`CrossEntropyLoss`), ce qui donne plus d'importance aux classes sous-représentées. Cette stratégie n'a pas été utilisée dans les autres modèles, car elle est naturellement compensée par le shuffle du `DataLoader`, les transformations, et la taille globale du dataset.

### 3.2.3 *Création d'un ensemble de validation*

Lors du premier entraînement du modèle ResNet18, nous avons constaté que la validation n'avait pas été correctement réalisée. Cela a conduit à un sur-apprentissage (overfitting), avec des performances élevées en apprentissage mais médiocres en test. Pour corriger cela : Nous avons séparé 20% des données d'entraînement pour constituer un ensemble de validation, permettant de surveiller la performance du modèle pendant l'apprentissage. Des métriques comme la perte, la précision, le rappel et la F1-mesure ont été suivies en temps réel à chaque époque.

### 3.2.4 *Préparation pour les modèles MLP et CNN*

Pour le MLP, les images ont été transformées en vecteurs unidimensionnels (flatten), tandis que pour le CNN, l'image conserve ses dimensions pour être traitée par les couches convolutives. Dans tous les cas : Les DataLoader ont été configurés avec un batch\_size de 32 pour accélérer l'entraînement. Le paramètre shuffle=True a été activé pour le dataset d'entraînement afin d'éviter l'apprentissage d'un ordre fixe.

Ces étapes de prétraitement ont été cruciales pour assurer la qualité de l'entraînement, la généralisation du modèle et une meilleure robustesse face aux variations dans les données. Elles ont permis d'améliorer significativement la performance de nos modèles sur le jeu de test final.

## 3.3 Conclusion

La mise en place de l'environnement de travail et de la base de données image a demandé environ une semaine de préparation, répartie entre la structuration du dossier dataset2 et l'implémentation du code de traitement dans le notebook. Une attention particulière a été portée à la cohérence des images et à leur répartition dans les dossiers TRAIN et TEST, car la réussite de la classification dépend de la qualité de l'annotation initiale. L'utilisation de ImageFolder a facilité cette étape, mais a nécessité de s'assurer que toutes les classes correspondaient à un nom de dossier unique. Parmi les principales difficultés rencontrées, on peut citer les lenteurs de traitement dans Google Colab, notamment lors des phases d'entraînement du modèle. Le réglage du nombre d'époques, combiné à un arrêt conditionné par la précision du modèle, a demandé du temps. Enfin, même si les images sont bien annotées, l'ajout de transformations (rotation, redimensionnement, normalisation) a demandé plusieurs essais progressifs pour éviter que le modèle n'apprenne à reconnaître que des images "trop propres". Malgré plusieurs obstacles, le projet a permis de mettre en place une chaîne complète de traitement d'images supervisé. Le système mis en place est bon, réutilisable et adaptable à d'autres jeux d'images.

---

## CHAPITRE 4

### Analyse et interprétation

---

#### 4.1 Introduction

Dans ce chapitre, nous présentons une analyse détaillée des performances des modèles développés pour la classification des cellules sanguines à partir d'images microscopiques. Notre objectif principal est d'évaluer l'efficacité de chaque approche, non seulement à travers la précision globale, mais également via des indicateurs plus fins tels que la perte (loss), le rappel (recall), la F1-mesure (F1-score) et la capacité de chaque modèle à distinguer correctement les différentes classes. Nous avons étudié trois familles de modèles complémentaires : un réseau de neurones convolutionnel (CNN) conçu manuellement, le modèle pré-entraîné ResNet18, réputé pour ses bonnes performances en extraction de caractéristiques profondes, ainsi qu'un perceptron multicouche (MLP), qui repose uniquement sur des couches entièrement connectées. À travers cette diversité, nous cherchons à comparer différentes approches d'apprentissage : de la conception sur mesure à l'usage de représentations issues de réseaux profonds, en passant par une approche plus classique fondée sur des vecteurs plats d'entrée. Nous avons complété cette démarche par l'utilisation de classificateurs traditionnels tels que k-Nearest Neighbors (kNN), Support Vector Machine (SVM) et Random Forest, que nous avons appliqués sur les représentations intermédiaires (features) extraites de nos modèles. Cela permet d'évaluer la qualité intrinsèque des représentations produites par chaque réseau, indépendamment de leur couche de sortie initiale. Avant d'analyser les résultats, nos attentes étaient les suivantes : nous anticipions de meilleures performances du modèle ResNet18, en raison de sa profondeur et de son entraînement préalable sur de grandes bases d'images. Le CNN personnalisé, bien que plus simple, devait offrir des performances acceptables grâce à une architecture adaptée à notre jeu de données. Enfin, le MLP, privé d'informations spatiales, risquait d'être moins performant, sauf s'il bénéficiait de bonnes représentations en entrée. Concernant les classificateurs, nous nous attendions à ce que le SVM et la Random Forest tirent parti de structures de données riches, tandis que le kNN, sensible à la densité locale, pouvait être limité si les représentations n'étaient pas bien séparables. L'ensemble de cette analyse repose sur des courbes d'apprentissage, des matrices de confusion, des rapports de classification et des métriques quantitatives, permettant de mettre en évidence les points forts et les limites de chaque approche, tant sur les performances brutes que sur leur capacité à généraliser sur des données jamais vues.

## 4.2 Modèles

### 4.2.1 CNN

#### *Courbe de perte (loss)*

La perte diminue très rapidement pendant les premières époques, puis ralentit et se stabilise autour de 0.05. Cela indique que le modèle a bien appris à minimiser l'erreur sur les données d'entraînement. Aucun signe clair de surapprentissage dans la perte, mais on le confirmera avec les résultats sur les données de test.

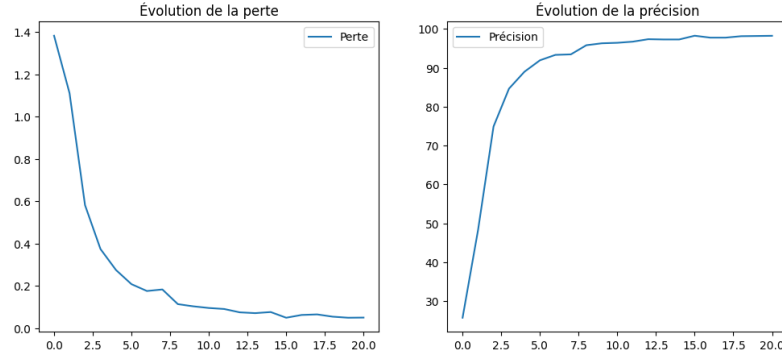


Figure 4.1: courbe de perte et de précision

#### *Courbe de précision (accuracy)*

Très belle montée de la précision, qui passe de 25% à 98%. La stabilisation qui est également visible vers les dernières époques montre que l'entraînement semble avoir convergé correctement. Alors, nous pouvons dire que l'évolution de la courbe de perte montre une diminution rapide au début de l'entraînement ce qui est un signe que le modèle apprend efficacement. De plus, l'augmentation progressive de la précision, atteignant une stabilité autour de 98%, confirme la convergence du modèle. Par ailleurs, d'après le rapport de classification, les Lymphocytes constituent la classe la mieux reconnue avec des scores quasi parfaits, les Monocytes et Eosinophiles présentent de bons résultats en précision mais des rappels plus faibles, indiquant que plusieurs exemples de ces classes ne sont pas détectés correctement et les Neutrophiles ont un bon rappel (0,88), mais la précision faible (0,67) suggère que d'autres classes sont parfois classées à tort comme Neutrophile. Ainsi, le modèle CNN entraîné sur les images de cellules sanguines a atteint une précision de 98.22% sur les données d'entraînement et une précision de 84.76% sur les données de test. Cette différence indique un léger surapprentissage, le modèle ayant très bien mémorisé les exemples d'entraînement mais généralisant un peu moins bien sur les données inédites. En somme, le modèle montre de bonnes capacités de classification, en particulier sur les lymphocytes. Toutefois, il confond certaines classes, notamment les monocytes et les neutrophiles, probablement en raison de similarités visuelles dans les images.

#### 4.2.2 RESNET18

Le modèle ResNet18 pré-entraîné a été adapté à la classification des cellules sanguines. Il a été entraîné pendant 21 époques, avec une stratégie d'augmentation de données, Dropout, et pondération des classes pour éviter le sur-apprentissage et gérer le déséquilibre. Nous avons observé une hausse rapide de la précision dès les premières époques :

- Epoch 1 : 80.9
- Epoch 6 : 97.2
- Epoch 10 : 99.1
- Epoch 15 - 21 : 100

Cela montre un apprentissage très efficace, une très bonne régularisation, notamment grâce au Dropout, à la normalisation des données, et à l'utilisation d'un modèle pré-entraîné. D'où les résultats finaux suivant :

- Précision d'entraînement finale : 99.82
- Précision de validation finale : 100
- Score F1 validation : 1.00

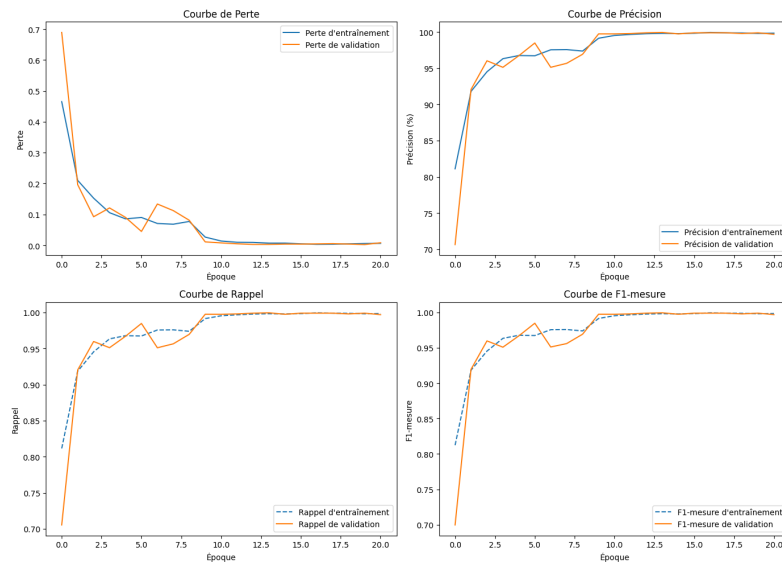


Figure 4.2: courbe de perte, de précision, de rappel et de F-mesure

##### *Courbe de perte (loss)*

La perte diminue rapidement dès les premières époques (epochs) et atteint presque zéro après 10 époques. Cela montre que : le modèle apprend très efficacement, la perte sur l'ensemble de validation rejoint celle d'entraînement sans divergence, ce qui indique peu ou pas de surapprentissage apparent au niveau de la perte. Toutefois, comme la perte est extrêmement faible, cela peut aussi signifier que le modèle s'est trop bien adapté au jeu de validation.

### *Courbe de précision (accuracy)*

La précision d'entraînement et de validation montent rapidement à environ 99%. Donc nous pouvons dire que le modèle atteint une quasi-perfection sur l'ensemble de validation dès l'époque 10. Cela renforce l'idée d'un risque de sur-adaptation, d'autant plus que la précision test (85.65%) est bien plus basse.

### *Courbe de rappel*

Le rappel de validation est légèrement fluctuant au début, puis se stabilise vers 0.99. Cela indique que le modèle rate peu de vrais positifs, ce qui est cohérent avec le haut rappel sur la validation. Cependant, sur le test, certaines classes (comme Monocyte) ont un rappel bien plus faible (0.72) ce qui veut dire que le modèle ne généralise pas aussi bien.

### *Courbe de F-mesure*

Il est similaire à la courbe de rappel c'est-à-dire que les courbes convergent rapidement vers 1.0. Cela indique que le modèle est très équilibré entre rappel et précision sur la validation. Mais sur le test, la F1-mesure est plus basse (0.86), confirmant une performance surévaluée sur validation. Alors, les courbes montrent que ResNet18 apprend très rapidement et efficacement. De plus, les métriques presque parfaites sur l'ensemble de validation, peut masquer un sur-apprentissage subtil. Mais, la différence avec les performances du test révèle que le modèle ne généralise pas aussi bien qu'il n'y paraît. Par ailleurs, d'après le rapport de classification, les Lymphocytes ont parfaitement reconnu, presque 100%, sur toutes les métriques; les Monocytes ont bien prédit (100% précision), mais le modèle rate beaucoup de vrais monocytes (rappel = 0.72), donc beaucoup de faux négatifs; les Neutrophiles, inversement, le rappel est très élevé (0.94), donc peu de faux négatifs, mais faible précision (0.67) donc beaucoup de faux positifs et les Eosinophiles ont des résultats corrects mais moins bons que les lymphocytes. Ainsi, le score de 85,65 % sur le jeu de test est proche de celui obtenu par le modèle CNN personnalisé (84.76%), ce qui est un peu surprenant étant donné que ResNet18 obtient des performances quasi parfaites sur le jeu de validation (100%). Cela peut indiquer une légère suradaptation à l'ensemble de validation ou simplement une complexité particulière dans les images du test. Le modèle ResNet18 s'est montré particulièrement performant sur l'ensemble d'entraînement et de validation, atteignant des taux de précision, de rappel et de F1-mesure très élevés (proches de 99 %). Les courbes d'apprentissage confirment une convergence rapide et stable, avec une faible perte et une absence apparente de surapprentissage sur la validation. Cependant, les résultats sur le jeu de test révèlent une baisse notable de performance avec une précision globale de 85.65 %. Certaines classes, comme les monocytes (rappel = 0.72) et les neutrophiles (précision = 0.67), sont moins bien reconnues, indiquant une difficulté de généralisation du modèle face à de nouvelles données. En somme, ResNet18 est un modèle puissant et efficace pour la classification des globules blancs, mais il semble suradapter légèrement aux données de validation.



### 4.2.3 MLP

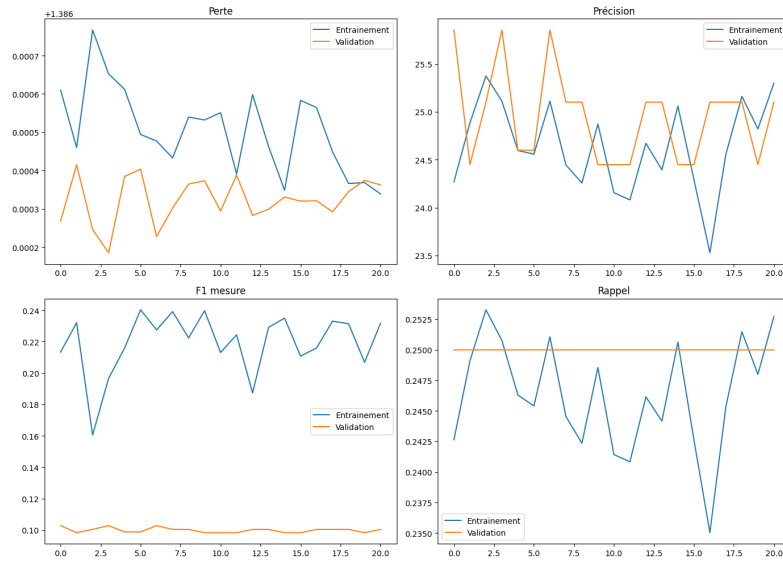


Figure 4.3: courbe de perte, de précision, de rappel et de F-mesure

#### *Courbe de perte (loss)*

La perte est globalement faible aussi bien pour l'entraînement que pour la validation (0.0003), et relativement stable. Cela peut nous tromper car une faible perte ne signifie pas nécessairement que le modèle apprend bien : ici, cela reflète surtout une faible sortie du modèle, avec des prédictions peu tranchées, sans réel lien avec les bonnes classes. De plus, le fait que la perte de validation soit inférieure à celle d'entraînement peut indiquer que le modèle ne généralise pas, mais fait des prédictions très incertaines, proches de la moyenne.

#### *Courbe de précision (accuracy)*

La précision d'entraînement varie autour de 24 - 25 %, avec de fortes fluctuations. La précision de validation oscille également sans progression représentative. Sachant que nous avons 4 classes, une précision de 25 % correspond à un modèle qui prédit au hasard. Nous pouvons dire que le MLP n'extraît pas certaines informations. C'est-à-dire que les couches entièrement connectées ne permettent pas de capturer la structure spatiale des images (formes, textures, etc.).

#### *Courbe de rappel*

Le rappel d'entraînement oscille autour de 0.24 - 0.25, tandis que celui de validation est parfaitement constant à 0.25. Cela confirme l'idée d'un modèle qui prédit toujours la même classe (ou de manière uniforme), ce qui donnerait un rappel de 25 % si chaque classe est prédite de manière égale sans discernement.

#### *Courbe de F-mesure*

Le F1- mesure est bas : entre 0.10 et 0.24 pour l'entraînement, autour de 0.10 constant en validation. Il combine précision et rappel, ce qui le rend très sensible aux

déséquilibres. Sa stagnation indique que le modèle n'apprend pas à bien identifier les différentes classes, ni à équilibrer les faux positifs et faux négatifs.

Ainsi, le score de 25.01 % sur le jeu de test confirme toutes les observations précédentes : le MLP n'apprend rien de pertinent sur les données. Le test ne montre aucune amélioration par rapport à la validation, ce qui indique une incapacité du modèle à généraliser. Donc, le modèle Multi-Layer Perceptron (MLP), bien que simple à implémenter, s'est révélé inadapté à la tâche de classification des images de globules blancs. Cela peut s'expliquer par plusieurs facteurs :

- L'absence d'apprentissage spatial. En effet, le modèle ne capte pas les relations locales entre pixels or ces relations sont essentielles pour détecter des motifs dans les images médicales.
- Un sur-apprentissage localisé ou prédiction biaisée. En effet, le modèle semble converger vers une stratégie naïve (comme prédire toujours la même classe) pour minimiser la perte sans chercher la performance.
- La complexité visuelle des images. En effet, les globules blancs ont des textures et formes subtiles que le modèle ne peut pas exploiter. Elle rend le problème trop difficile pour un réseau purement dense, même avec plusieurs couches. En somme, le MLP est un point de départ intéressant pour comprendre les réseaux de neurones, mais totalement inadapté à des images médicales complexes comme celles des globules blancs. Il souffre d'un manque de capacité à exploiter l'information spatiale, essentielle pour cette tâche.

#### 4.2.4 Conclusion

Le MLP affiche des performances très faibles. Cela s'explique par le fait qu'il ne prend pas en compte la structure spatiale des images. Chaque pixel est traité indépendamment, ce qui limite fortement la capacité du modèle à extraire des motifs visuels pertinents. Le CNN classique, composé de quelques couches convolutionnelles suivies de couches fully connected, améliore drastiquement les résultats (près de +60 % de précision par rapport au MLP). Il parvient à extraire des informations locales utiles comme les textures, formes ou bordures caractéristiques des différentes classes de cellules. Le modèle ResNet18, pré-entraîné sur ImageNet, donne les meilleures performances, dépassant légèrement le CNN personnalisé. Cette légère amélioration s'explique par :

- L'utilisation de blocs résiduels, facilitant l'entraînement des réseaux profonds.
- Le transfert d'apprentissage, permettant de réutiliser des caractéristiques visuelles génériques apprises sur un très large corpus (ImageNet).
- Une meilleure généralisation, même avec un dataset de taille limitée.

### 4.3 Analyse des classificateurs

#### 4.3.1 Classificateurs du CNN

KNN

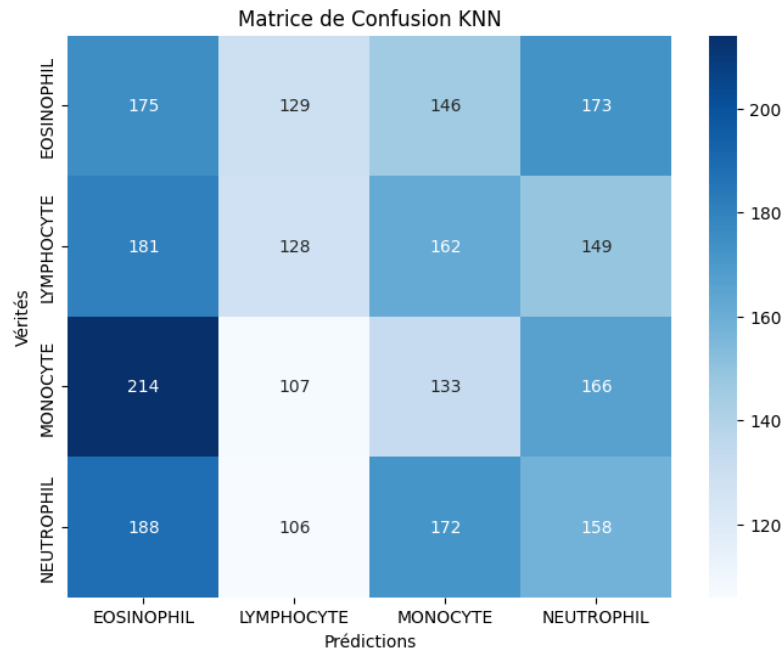


Figure 4.4: matrice de confusion KNN

Avec le classificateur kNN, les performances sont globalement faibles. Par exemple, pour la classe EOSINOPHIL, on obtient une précision de 0,23, un rappel de 0,28 et une F1-mesure de 0,25. Les autres classes suivent une tendance similaire : LYMPHOCYTE (précision 0,27, rappel 0,21), MONOCYTE (précision 0,22, rappel 0,21) et NEUTROPHIL (précision 0,24, rappel 0,25). L'accuracy globale atteint seulement 24%, ce qui est très faible. La matrice de confusion montre une forte confusion entre toutes les classes, sans dominance claire. Ces résultats peuvent s'expliquer par le fait que les caractéristiques extraites par le CNN ne sont pas bien séparées dans l'espace euclidien, ce qui rend la tâche difficile pour un kNN. De plus, ce modèle souffre souvent de la malédiction de la dimension lorsque le nombre de caractéristiques (features) est élevé, ce qui affecte directement sa capacité à distinguer les classes.

## SVM

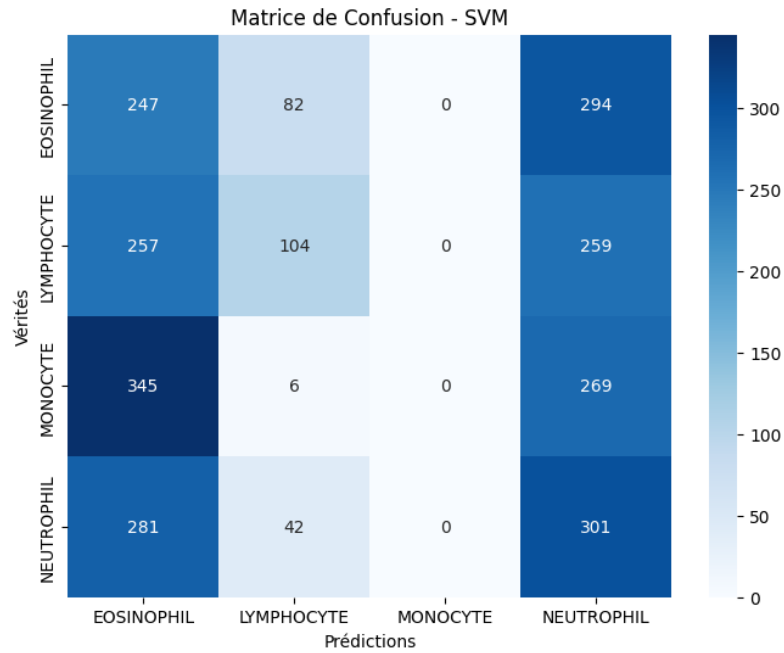


Figure 4.5: matrice de confusion SVM

Le SVM avec noyau linéaire donne des résultats légèrement meilleurs que le kNN, mais reste globalement médiocre. On remarque un comportement très déséquilibré : la classe EOSINOPHIL atteint un rappel de 0,40 (précision 0,22), et NEUTROPHIL un rappel de 0,48 (précision 0,27). À l'inverse, MONOCYTE n'est jamais prédit (0% dans toutes les métriques). L'accuracy atteint 26%, mais la matrice de confusion révèle une forte prédiction vers certaines classes (comme EOSINOPHIL et NEUTROPHIL) au détriment d'autres. Cette inégalité peut s'expliquer par le fait que le SVM linéaire n'est pas capable de capturer les frontières de décision complexes entre les classes, ce qui est souvent le cas avec des caractéristiques profondes issus de CNN. L'utilisation d'un noyau non linéaire pourrait potentiellement améliorer les performances.

## RANDOM FOREST

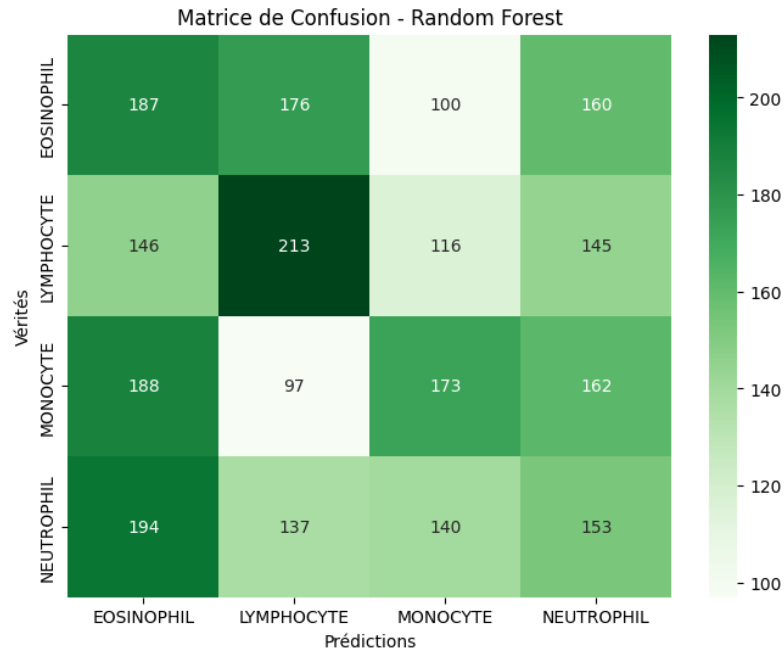


Figure 4.6: matrice de confusion random forest

Le classificateur Random Forest donne les résultats les plus équilibrés parmi les trois. Toutes les classes sont raisonnablement bien représentées : EOSINOPHIL (précision 0,26, rappel 0,30), LYMPHOCYTE (0,34, 0,34), MONOCYTE (0,33, 0,28), et NEUTROPHIL (0,25, 0,25). L'accuracy globale atteint 29%, et la matrice de confusion montre une répartition plus homogène des prédictions, bien que la confusion reste présente. Ce bon équilibre s'explique par la capacité du Random Forest à gérer les relations non linéaires et les interactions complexes entre caractéristiques tout en étant moins sensible à la dimensionnalité. Il semble mieux exploiter les caractéristiques issues du CNN par rapport aux deux autres modèles testés.

### 4.3.2 Classificateurs du RESNET18

#### KNN

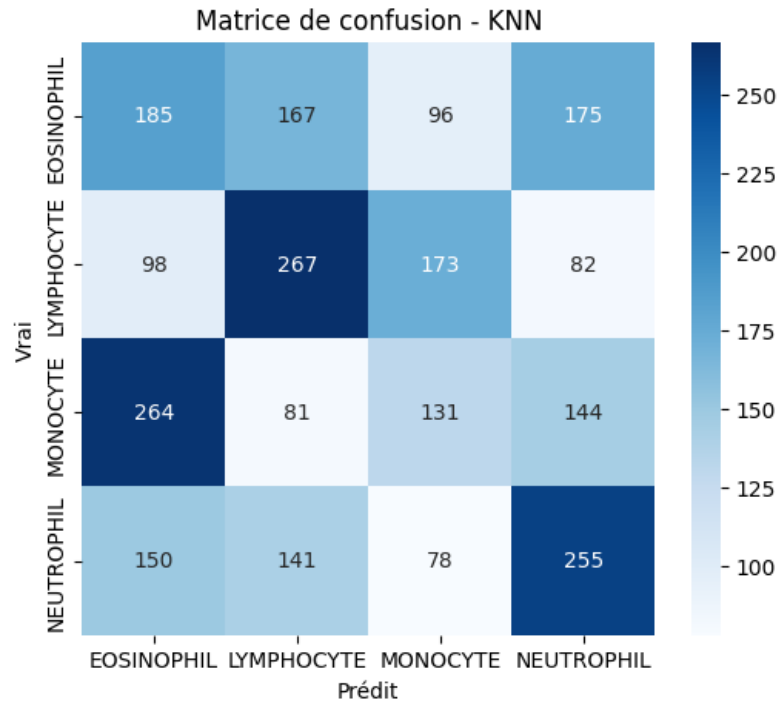


Figure 4.7: matrice de confusion kNN

Avec les caractéristiques extraites de ResNet18, le classificateur kNN atteint une précision ( accuracy ) globale de 34%, ce qui marque une nette amélioration par rapport à l'utilisation directe des features (caractéristiques) CNN précédents. La matrice de confusion montre une certaine confusion persistante entre EOSINOPHIL, MONOCYTE et NEUTROPHIL, mais les classes LYMPHOCYTE et NEUTROPHIL sont mieux différenciées que précédemment. Cela montre que les représentations extraites par ResNet18 sont plus informatives dans l'espace euclidien, ce qui bénéficie à kNN. Néanmoins, la proximité entre certaines classes reste problématique, comme en témoignent les nombreux cas de MONOCYTE classés en EOSINOPHIL (264 erreurs).

SVM

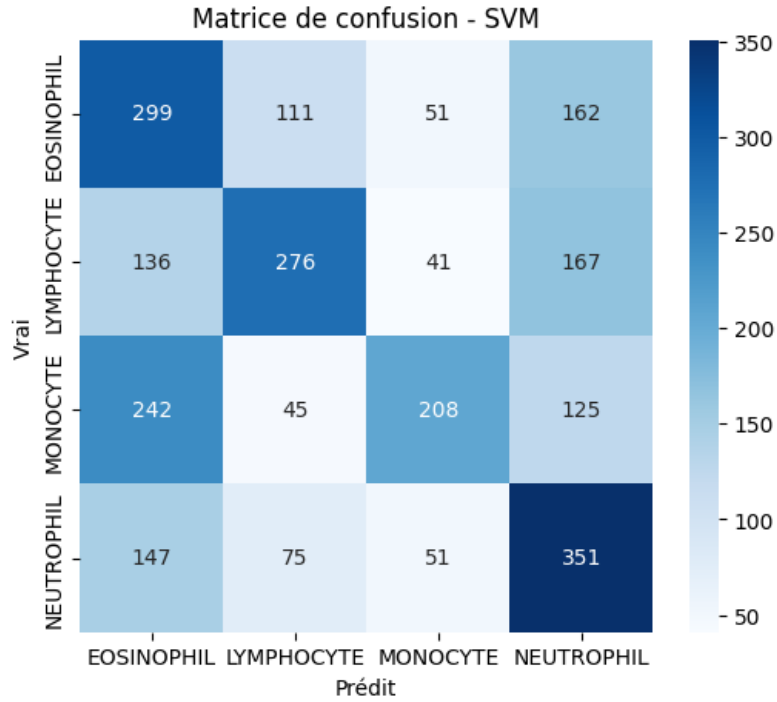


Figure 4.8: matrice de confusion SVM

Le SVM linéaire montre ici la meilleure performance globale avec 46% de précision, ce qui confirme que les représentations extraites par ResNet18 sont beaucoup plus discriminantes. La matrice de confusion reflète un meilleur équilibre des prédictions. Par exemple, NEUTROPHIL est bien prédit (351 bonnes prédictions sur 624), et LYMPHOCYTE également (276/620). Même les classes difficiles comme MONOCYTE sont mieux identifiées (208 prédictions correctes). Ce résultat s'explique par la capacité du SVM à tirer parti d'un espace de représentation linéairement plus séparable, ce qui est souvent le cas avec des features profonds bien appris. On observe néanmoins quelques confusions persistantes entre EOSINOPHIL et MONOCYTE.

## RANDOM FOREST

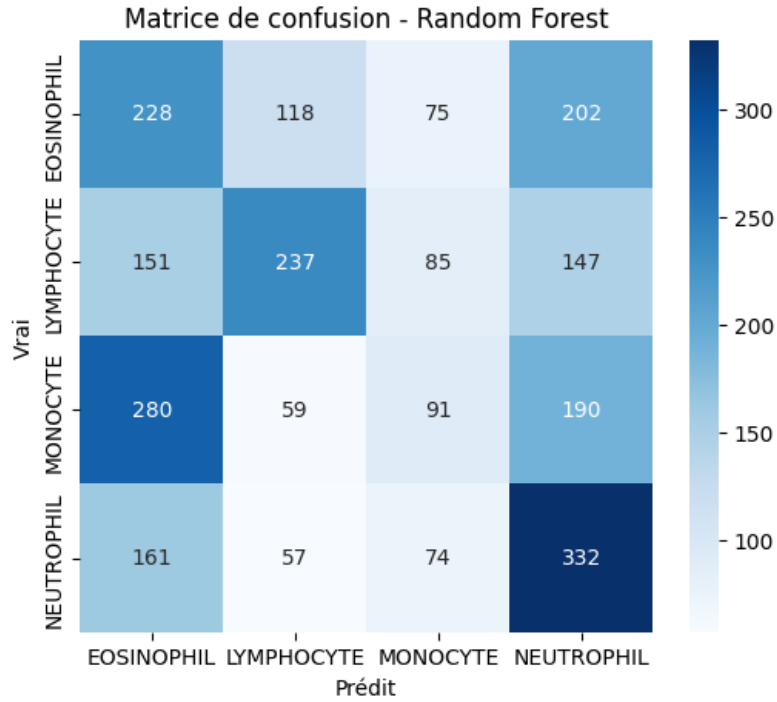


Figure 4.9: matrice de confusion random forest

Le Random Forest obtient une accuracy de 36%, légèrement inférieure au SVM, mais supérieure au kNN. Sa matrice de confusion montre une répartition relativement équilibrée, avec des prédictions correctes sur toutes les classes, en particulier NEUTROPHIL (332), LYMPHOCYTE (237), et EOSINOPHIL (228). Toutefois, MONOCYTE reste la classe la plus confuse, avec 280 erreurs de classification en EOSINOPHIL. Ce résultat reste encourageant car Random Forest est capable d'exploiter les interactions complexes entre les features extraits par ResNet, mais peut être affecté d'une sensibilité au bruit et au déséquilibre des classes, malgré l'utilisation d'un `class_weight`.



### 4.3.3 Classificateurs du MLP

#### KNN

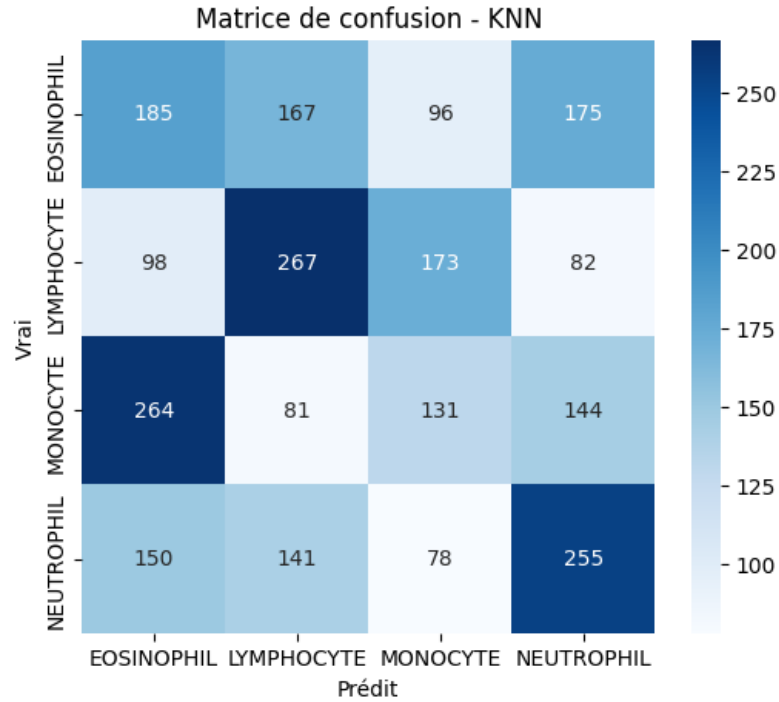


Figure 4.10: matrice de confusion KNN

Avec les caractéristiques extraites du MLP, le classificateur kNN atteint une accuracy globale de 34%, ce qui montre une performance équivalente à celle obtenue avec les features de ResNet18. La matrice de confusion révèle toutefois une forte confusion entre les classes MONOCYTE, EOSINOPHIL et NEUTROPHIL. Par exemple, 264 instances de MONOCYTE ont été incorrectement classées comme EOSINOPHIL. Cette confusion suggère que, malgré la transformation non linéaire offerte par le MLP, la distance euclidienne utilisée par le kNN ne suffit pas à séparer les classes dans ce nouvel espace de représentation. Les caractéristiques extraites restent donc insuffisamment discriminantes pour ce type de classifieur.

SVM

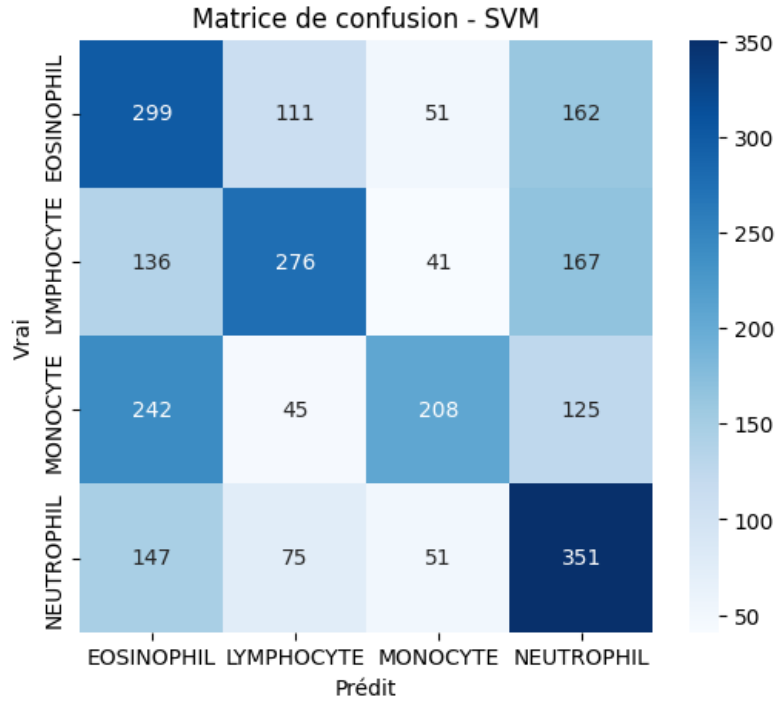


Figure 4.11: matrice de confusion SVM

Le classificateur SVM appliqué aux features extraits du MLP atteint une accuracy de 46%, ce qui en fait l'un des meilleurs modèles testés dans ce contexte. La matrice de confusion indique une meilleure séparation des classes, en particulier pour NEUTROPHIL (351 prédictions correctes) et LYMPHOCYTE (276). Toutefois, certaines confusions subsistent entre MONOCYTE et EOSINOPHIL (242 erreurs), ce qui montre que la séparation linéaire du SVM, bien qu'efficace, atteint ses limites lorsque les représentations restent partiellement entremêlées. Cela dit, le MLP semble produire des features plus utiles que les précédents pour un modèle linéaire comme le SVM.

## RANDOM FOREST

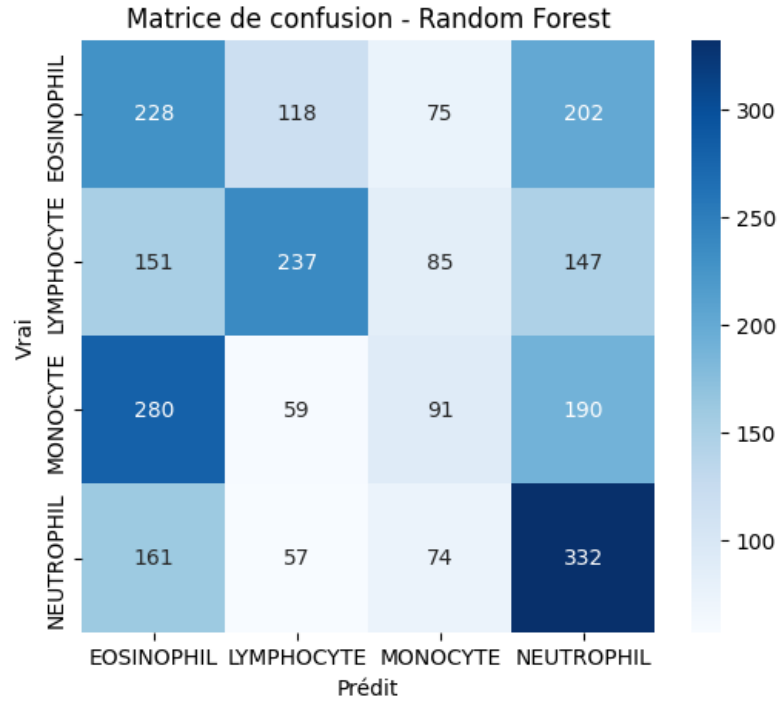


Figure 4.12: matrice de confusion random forest

Avec les features fournis par le MLP, Random Forest atteint une précision globale de 36%, ce qui reste en dessous des performances du SVM mais comparable à celles de kNN. Le modèle parvient à bien identifier certaines classes comme NEUTROPHIL (332 prédictions correctes) et LYMPHOCYTE (237), mais se heurte à des confusions marquées, notamment 280 MONOCYTE prédits en EOSINOPHIL. Ce comportement laisse penser que, bien que Random Forest capte mieux les interactions non linéaires entre features, certaines classes restent difficilement séparables, probablement à cause de leur similarité morphologique. Le MLP améliore donc globalement la qualité des représentations, mais ne suffit pas à éliminer toutes les ambiguïtés entre classes.

### 4.3.4 Conclusion

Analyse comparative des modèles			
Modèle	Type	Précision test	Observation
MLP	Réseau de neurones dense	25,01%	Apprentissage très limité et son comportement est proche du hasard
CNN	Réseau convolutif personnalisé	84,76%	Bon apprentissage des caractéristiques visuelles
Resnet18	CNN profond pré-entraîné	85,65%	Meilleur score et généralisation plus robuste grâce au transfert d'apprentissage

Figure 4.13: Analyse comparative des modèles

Comparaison Synthétique			
Critères	MLP	CNN	Resnet18
Capacité à traiter les images	Faible	Bonne	Excellente
Précision ( test)	25,01%	84,76%	85,65%
Apprentissage de motifs visuels	Non	Oui	Oui
Complexité	Faible	Moyenne	Élevé
Temps d'entrainement	Rapide	Moyen	Plus long

Figure 4.14: Comparaison synthétique

Cette étude a comparé plusieurs modèles de classification pour la reconnaissance des cellules sanguines : CNN, ResNet18 et MLP, ainsi que des classificateurs classiques appliqués aux caractéristiques extraites de ces réseaux. Le CNN a montré de bonnes performances avec une précision d'entraînement de 98,22 % et une précision de test de 84,76 %. Toutefois, un léger surapprentissage a été observé, le modèle généralisant moins bien sur les nouvelles données. Il a bien distingué certaines classes, notamment les lymphocytes, mais a eu du mal à différencier d'autres classes similaires. Le modèle ResNet18, pré-entraîné, a surpassé le CNN avec une précision d'entraînement de 99,82 % et une précision de validation de 100 %. Cependant, la précision de test est tombée à 85,65 %, ce qui suggère un surapprentissage ou des difficultés à généraliser. Bien que les performances sur la validation aient été excellentes, certaines classes, comme les monocytes, ont montré un faible rappel sur les données de test. Le MLP a obtenu des résultats très faibles, avec une précision de test de seulement 25,01 %. Ce modèle a échoué à capter les informations visuelles locales essentielles pour ce type de tâche, étant limité par son incapacité à prendre en compte la structure spatiale des images. Quant aux classificateurs classiques appliqués aux caractéristiques extraites des modèles, les performances étaient largement dépendantes de la qualité des représentations produites par les réseaux. Le Random Forest a montré les résultats les plus équilibrés, tandis que les classificateurs kNN et SVM ont eu des performances inférieures. En résumé, les modèles basés sur des réseaux de neurones profonds, comme le CNN et le ResNet18, ont prouvé leur efficacité pour la classification des cellules sanguines, bien qu'il faille prendre en compte le risque de surapprentissage. Le MLP, en revanche, s'est révélé inadéquat pour ce problème en raison de son incapacité à capturer les relations spatiales dans les images. Les classificateurs classiques ont montré des performances limitées, bien que les caractéristiques extraites des modèles profonds aient amélioré leurs résultats.

---

## CHAPITRE 5

### Conclusions et perspectives

---

Au cours de ce projet, nous avons conçu plusieurs modèles d'apprentissage supervisé pour la classification d'images de cellules sanguines en quatre classes. Nous avons d'abord développé un modèle CNN à partir de zéro, que nous avons entraîné avec des stratégies d'augmentation de données et d'early stopping pour l'efficacité du modèle. Nous avons ensuite testé différentes combinaisons en extrayant les features de ce modèle pour les passer à des classifieurs traditionnels (KNN, SVM, Random Forest), ce qui nous a permis de mieux comprendre la qualité des représentations internes. Par la suite, nous avons utilisé un modèle pré-entraîné (ResNet18) comme extracteur de caractéristiques, ce qui a significativement amélioré les performances. Enfin, nous avons testé une architecture MLP, avec des résultats plus modestes et plus intéressants. Ces expérimentations ont été accompagnées de courbes, de matrices de confusion et de rapports de classification détaillés. Notre approche a été systématique, nous avons testé différents modèles, mais avons également cherché à les interpréter et à identifier leurs points forts et leurs limites, notamment en ce qui concerne la séparation des classes complexes comme les monocytes. En outre, nous avons appris à préparer et transformer un dataset image pour l'entraînement sur les modèles. Nous avons entraîné et évalué plusieurs types de modèles. Nous avons analysé les résultats retournés à travers des métriques variées comme la précision, le rappel, la F1-score et la matrice de confusion. Finalement, nous avons comparé les approches pour en tirer des enseignements concrets et utiles. Nous avons mis en place un système d'arrêt automatique de l'entraînement modulable si notre modèle est utilisé pour un autre dataset. Nos perspectives à court terme seraient d'ajouter une interface graphique comme Streamlit pour permettre de tester d'autres images, ce qui pourrait être intéressant si des laboratoires de recherches veulent un modèle de machine learning pour faciliter et accélérer la labellisation d'image de cellules. D'autres perspectives seraient de tester sur d'autres images pour voir comment le modèle réagit face à des données différentes mais semblables. On pourrait par ailleurs créer des affichages automatiques des prédictions des images ou mêmes des affichages en général. Nos perspectives à long terme qui pourraient durer 1 à 2 mois seraient de créer des évaluations plus précises c'est-à-dire mesurer les performances par classe en fonction de l'origine des images pour détecter des biais ou autres variabilités. Le stockage des résultats en ajoutant un enregistrement automatique des prédictions et des scores dans un fichier csv ainsi on pourrait faire appel à des statistiques pour évaluer mathématiquement les résultats. Nous pourrions réaliser des comparaisons automatiques ainsi les modèles pourraient être comparés entre eux à l'aide d'un tableau de synthèse des scores pour mieux faciliter l'analyse. Le plus essentiel serait de déployer notre projet sur une application ou site web afin

d'avoir une interface plus accessible pour des personnes extérieures, par exemple pour un chercheur dans un laboratoire biologique. Nous avons fait en sorte que le projet soit fait de part égale pour chaque membre afin de prendre la main sur plusieurs compétences. De ce fait, nous avons acquis de bonnes compétences et revu différents outils. Nous avons acquis une bonne autonomie dans notre projet, notamment dans l'entraînement et l'évaluation de modèles d'apprentissage profond sur des images. Nous avons identifié des axes d'amélioration faciles à implémenter. Si ce projet était poursuivi en stage nous aurions un bon point de départ et la capacité d'avancer rapidement.

---

## Bibliographie

---

Nos ressources proviennent de nos cours de sciences des données, nous avons utilisé chatgpt pour optimiser et corriger des fautes dans nos codes cependant la base de nos codes ne proviennent en aucun cas de chatgpt ou autre IA.