



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET
POPULAIRE**

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Kheider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N°d'ordre : Startup_IVA05/M2 16/06/2025

Mémoire

Pour l'obtention du diplôme Master académique en Informatique

Spécialité

Spécialité : système d'information optimisation et décision

Spécialité : Images et vie Artificielle

Développement d'une application mobile pour la classification du cancer du sang à partir d'images médicales en utilisant des algorithmes d'apprentissage profond

Présenté par :

Adaika Insaf

Ouamane Rayhana

Supervisé par :

Encadrant : Pr. Abdealmalik Ouamane

Co-encadrant : Dr. Abdelmoumen Zerari

Soutenu le 16/06/2025, devant le jury composé de :

Président	Babahenini Djihane	Grade
Rapporteur	Ouamane Abdeal Malik	Grade
Examinateur	Chighoub Rabia	Grade

Année Universitaire 2024/2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Résumé

L'application mobile pour médecins de classification des leucémies repose sur des techniques d'intelligence artificielle et d'apprentissage automatique, notamment le traitement d'images médicales, pour analyser les images sanguines et diagnostiquer les cas avec précision et efficacité. Des algorithmes avancés sont utilisés pour détecter et classer les cellules cancéreuses selon des critères médicaux spécifiques, offrant ainsi un diagnostic rapide et précis qui aide les médecins à prendre des décisions thérapeutiques appropriées.

L'application apporte une valeur ajoutée au secteur médical en réduisant le temps nécessaire à l'analyse manuelle, en améliorant la précision du diagnostic et en minimisant les erreurs humaines. De plus, elle contribue à l'amélioration des soins de santé en permettant aux médecins d'obtenir des résultats instantanés et fiables, en particulier dans les régions manquant d'expertise médicale spécialisée.

Dans ce contexte, le développement de cette application représente un défi technologique majeur et une opportunité stratégique pour renforcer les soins de santé grâce aux technologies modernes. L'application permet aux utilisateurs (médecins et professionnels de santé) de télécharger des images médicales, d'obtenir une analyse détaillée et de stocker les résultats pour un suivi ultérieur. Elle propose également des options de personnalisation pour l'affichage des rapports et des notifications, en faisant un outil efficace dans la lutte contre les leucémies.

Mots-clés : Intelligence artificielle, apprentissage automatique, leucémie, imagerie médicale, application médicale.

Abstract

The mobile application for leukemia classification designed for physicians relies on artificial intelligence and machine learning techniques—particularly medical image processing—to analyze blood images and diagnose cases with accuracy and efficiency. Advanced algorithms are employed to detect and classify cancerous cells based on specific medical criteria, providing rapid and precise diagnostics that assist doctors in making appropriate therapeutic decisions.

This application brings added value to the medical field by reducing the time required for manual analysis, improving diagnostic accuracy, and minimizing human error. Moreover, it contributes to better healthcare by allowing physicians to obtain instant and reliable results, especially in regions lacking specialized medical expertise.

In this context, the development of this application represents a major technological challenge and a strategic opportunity to enhance healthcare services through modern technologies. The app enables users (physicians and healthcare professionals) to upload medical images, receive detailed analyses, and store results for follow-up. It also offers customization options for report display and notifications, making it an effective tool in the fight against leukemia.

Keywords : Artificial intelligence, machine learning, leukemia, medical imaging, medical application.

Remerciements

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.
Louange à Allah qui nous a permis, à moi et à mon amie,
d'achever ce travail.

Nous adressons nos sincères remerciements, notre profonde reconnaissance et tout notre amour à l'ensemble de nos éminents professeurs. Nous tenons à exprimer toute notre gratitude au Professeur Abdelmalik Ouamane, qu'Allah le protège et lui accorde longue vie, pour la supervision généreuse de ce mémoire et ses orientations précieuses qui nous ont permis de le mener à bien.

Nous exprimons également notre reconnaissance à tous nos professeurs estimés qui nous ont accompagnées tout au long de notre parcours universitaire.

À tous nos collègues et amis qui nous ont soutenues et accompagnées, nous vous adressons nos remerciements les plus sincères.

Enfin, nous tenons à exprimer notre profonde gratitude envers nous-mêmes pour notre persévérance, notre détermination et nos efforts continus afin de donner le meilleur de nous-mêmes

Dédicace

Louange à Dieu, le Tout-Puissant, le Miséricordieux, Celui qui m'a guidé, soutenu et comblé de Ses bienfaits tout au long de mon parcours. À Lui reviennent mes premières pensées de gratitude et de reconnaissance.

À ma famille bien-aimée, source de ma joie et mon soutien dans cette vie... À mon père, mon roc inébranlable, et ma mère, dont l'amour est infini, merci pour toutes les valeurs et la tendresse que vous m'avez transmises. Mon petit frère, ma fierté et ma joie, ainsi que ma sœur Imane et sa fille Rital, votre lumière éclaire mon chemin. Ma chère cousine, toi qui as été mon soutien dans les moments les plus sombres... Aucun mot ne peut exprimer ma gratitude. Mon amour Sarah, toi qui fais battre mon cœur de bonheur, ma vie n'aurait aucun sens sans toi. Mes tantes adorées, qui m'ont donné l'affection maternelle, et mon oncle Aziz Said, merci pour tout ce que vous m'avez offert.

Mes merveilleuses amies (Celia, Rounak, Malak, Chaima), avec vous j'ai appris le vrai sens de l'amitié. Mes cousines et cousins, merci pour tous ces souvenirs inoubliables. Ma grand-mère Zakia (que Dieu ait son âme), toi qui nous as inculqué la foi et la bonté, puisses-tu reposer en paix. Et ma grand-mère Habiba, ta bénédiction continue de me protéger. Ma voisine chérie Habiba, toi qui partages mes joies et me soutiens dans mes peines, merci pour tout. Samia, tes efforts et tes encouragements ont été une lumière sur mon chemin, je n'oublierai jamais ta gentillesse.

À ma binôme : Ouamane Rayhana.

Enfin... à tous ceux qui ont contribué à mon bonheur ou séché mes larmes, ces mots ne sont qu'une infime partie de ma gratitude éternelle.

Avec tout mon amour et ma fidélité.

Insaf

Dédicace

Louange à Dieu, le Tout-Puissant, le Miséricordieux, Celui qui m'a guidé, soutenu et comblé de Ses bienfaits tout au long de mon parcours. À Lui reviennent mes premières pensées de gratitude et de reconnaissance. À ma chère mère et à mon cher père Mustapha, sources infinies d'amour, de patience et de courage. Merci pour vos prières, vos sacrifices et votre soutien inconditionnel. Ce travail est avant tout le fruit de votre présence dans ma vie.

À mon frère Abdelmalek, mon guide, mon encadrant et mon pilier tout au long de ce mémoire. Ta disponibilité, ta bienveillance et tes conseils ont été précieux. Je te remercie du fond du cœur.

À mes frères Oussama, Zakaria et Mohamed, merci pour votre amour, vos encouragements et vos mots qui m'ont portée dans les moments difficiles.

À notre petite étoile, Dania, ta joie de vivre et ton innocence ont adouci mes journées les plus chargées. À ma binôme : Adaika insaf.

À moi-même, pour toute la patience et la persévérance, malgré la fatigue et parfois la déception.

J'adresse cette dédicace à tous ceux qui ont contribué de près ou de loin à mon parcours, et à ceux qui ont cru en moi depuis le début. Merci infiniment.

Rayhana

Table des matières

Introduction Générale	13
.1 Problématique	15
.2 Objectifs et solution	16
.2.1 Objectifs du projet	16
.2.2 Solution proposée	16
Chapitre I: Classification du cancer du sang : concepts de base et état de l'art	17
I.1 Introduction	17
I.2 Le cancer du sang et ses types	18
I.2.1 Introduction au cancer du sang	18
I.2.2 Les principaux types de cancers du sang	19
I.2.3 Caractéristiques observables dans les frottis sanguins	22
I.3 Applications de l'IA en médecine	24
I.3.1 Convolutional neural network(CNN)	25
I.3.1.1 Introduction CNN	25
I.3.1.2 Convolutional neural network (CNN) classique	25
I.3.2 Apprentissage par transfert (Transfer Learning)	27
I.3.2.1 AlexNet	28
I.3.2.2 CaffNet	29
I.3.2.3 VGGNet	29
I.3.2.4 GoogLeNet	29
I.3.2.5 ResNet	30
I.3.2.6 MobileNet	31
I.3.2.7 XCEPTION	32
I.4 Conclusion	32

Chapitre II: Collecte et Prétraitement des Données	33
II.1 Introduction	33
II.2 Sources de données	34
II.2.1 Blood Cells Image Dataset	34
II.2.2 Lymphoblastic Leukemia (ALL)	37
II.3 Prétraitement des données	40
II.3.1 Augmentation des données	40
II.3.1.1 Explication du code	42
II.3.1.2 Répartition des données	42
II.4 Conclusion	44
Chapitre III: Développement des Modèles de Classification du cancer du sang	45
III.1 Introduction	45
III.2 Environnement de développement	46
III.2.1 Python	46
III.2.2 TensorFlow	46
III.2.2.1 Utilité de TensorFlow	48
III.2.3 Keras	48
III.2.3.1 Utilité de Keras	49
III.3 Notre système de classification du cancer du sang à l'aide de algorithmes d'apprentissage profond	49
III.4 Conception des modèles	51
III.4.1 Extraction de caractéristiques par CNN	51
III.4.2 Utilisation de modèles pré-entraînés (MobileNetV2 et ResNet101) pour le transfert d'apprentissage	53
III.4.2.1 MobileNetV2	53
III.4.2.2 ResNet101	54
III.5 Paramètres du modèle	55
III.6 Évaluation des performances	56
III.6.1 Fonction de perte (Loss Function)	56
III.6.2 Taux de reconnaissance (Accuracy)	57
III.6.3 Précision (Precision)	57
III.6.4 Rappel (Recall)	57
III.6.5 F1-score	57

III.6.6 Matrice de Confusion	57
III.7 Évaluation des performances des modèles	58
III.7.1 Résultats sur la base de données 'Blood Cells Image'	58
III.7.2 Résultats sur la base de données 'Lymphoblastic Leukemia ALL'	63
III.7.3 Comparaison générale des résultats	67
III.8 Conclusion	68
Chapitre IV: Développement de l'Application Mobile et Tests	69
IV.1 Introduction	69
IV.2 Diagrammes de formation de modèles	70
IV.2.1 Diagramme de cas d'utilisation	70
IV.2.2 Diagramme de Séquence	72
IV.2.2.1 Système d'Authentification et d'Enregistrement des Utilisateurs	72
IV.2.2.2 Gestion du Profil Médecin	72
IV.2.2.3 Processus de Diagnostic et de Sauvegarde — BloodDetect AI	73
IV.2.3 Diagramme de Classe	75
IV.3 Technologies pour le Développement Mobile	76
IV.3.1 Android Studio	76
IV.3.2 Utilité d'Android Studio	76
IV.4 Conception de l'interface utilisateur	77
IV.4.1 BloodDetect AI – Connexion ou Inscription	77
IV.4.2 Login et Sign Up	77
IV.4.3 Écran d'accueil de l'application Doctar	78
IV.4.4 Page de diagnostic sanguin	79
IV.4.5 Historique des analyses sanguines (History)	79
IV.4.6 Gestion des consultations	80
IV.4.7 Paramètres de compte (Settings) et modification du profil	80
IV.5 Intégration du modèle	81
IV.5.1 Conversion du modèle	81
IV.5.2 Optimisation des performances	82
IV.6 Tests de l'application	82
IV.6.1 Tests utilisateurs	82
IV.7 Conclusion	83
Conclusion et Perspectives	84

Liste des figures

I.1	Les cinq principaux types de globules blancs : (a) basophile, (b) éosinophile, (c) lymphocyte, (d) monocyte, (e) neutrophile.	20
I.2	Constituants du sang et principaux types de leucémie	21
I.3	Architecture CNN conventionnelle	26
I.4	Architecture AlexNet [1]	28
I.5	Blocs de convolution : (a) Bloc avec deux couches de convolution 3×3 , (b) Bloc avec trois couches de convolution 3×3	29
I.6	Module de démarrage.	30
I.7	Bloc d'apprentissage résiduel.	30
I.8	Convolution séparable en profondeur [2], [3].	31
II.1	Blood Cells Image Dataset ; (a) Basophils, (b)Eosinophils, (c)Erythroblasts , (d) Immature Granulocytes , (e) Lymphocytes, (f)Monocytes , (g)Neutrophils , (h)Platelets ,	36
II.2	La base de données ALL-B, (a) Benign (hematogone) , (b) Early Pre-B ALL , (c) Pre-B , (d)Pro-B ALL	39
II.3	La rotation, le retournement horizontal et le retournement vertical	40
III.1	Python	47
III.2	TensorFlow	47
III.3	keras	48
III.4	Notre système de classification du cancer du sang	50
III.5	(a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée	61
III.6	(a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée	62
III.7	(a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée	65

III.8 (a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée	66
IV.1 Diagramme de cas d'utilisation de l'application BloodDetect AI	71
IV.2 Diagramme de séquence — Système d'authentification et d'enregistrement des utilisateurs	72
IV.3 Diagramme de séquence — Gestion du profil médecin	73
IV.4 Diagramme de séquence — Processus de diagnostic et de sauvegarde dans BloodDetect AI	74
IV.5 Diagramme de classe du système	75
IV.6 Environnement Android Studio	77
IV.7 BloodDetect AI – Connexion ou Inscription	77
IV.8 Pages de connexion (Login) et d'inscription (Sign Up)	78
IV.9 Écran d'accueil de l'application Doctar – Diagnostic rapide du cancer du sang	78
IV.10Page de diagnostic sanguin : enregistrement du patient et affichage des résultats détaillés	79
IV.11Historique des analyses sanguines	79
IV.12Gestion des consultations	80
IV.13Paramètres de compte et modification du profil	81

Liste des tableaux

II.1	Types et nombre de cellules dans chaque groupe (Blood Cells Image Dataset)	35
II.2	Distribution des images dans la base de données Acute Lymphoblastic Leukemia (ALL)	38
II.3	Un nombre total d'images d'entraînement, de validation et de test sont utilisées pour les ensembles de données Blood Cells Image dataset, Lymphoblastic Leukemia (ALL) et Blood Cells Cancer (ALL) dataset, avec et sans augmentation de données.	43
III.1	Spécifications du système	46
III.2	Mesures de performance des différents modèles de CNN sur la base de données 'Blood Cells Image' pour la classification du cancer du sang. Par. d'ap. : Paramètres d'apprentissage, Esp. Sto. :Espace de stockage.	60
III.3	Mesures de performance des différents modèles de CNN sur la base de données ' Lymphoblastic Leukemia ALL' pour la classification du cancer du sang. Par. d'ap. : Paramètres d'apprentissage, Esp. Sto. :Espace de stockage.	64

Introduction Générale

Le cancer du sang, englobant des maladies telles que la leucémie, le lymphome et le myélome, constitue un défi majeur en hématologie. Son diagnostic repose essentiellement sur l'analyse microscopique des frottis sanguins, une tâche exigeante, chronophage et susceptible d'erreurs humaines. L'automatisation de cette analyse à l'aide des techniques d'intelligence artificielle, en particulier l'apprentissage profond (Deep Learning), offre une opportunité prometteuse d'améliorer à la fois la précision et la rapidité du diagnostic.

Ce projet de master a pour objectif de développer une application mobile destinée à la détection et à la classification du cancer du sang à partir d'images médicales de frottis sanguins. L'ambition est de fournir aux professionnels de santé un outil intelligent, fiable et efficace, capable d'analyser et de catégoriser les cellules sanguines anormales.

L'application repose sur des modèles de réseaux neuronaux convolutifs (CNN), qui ont démontré leur efficacité dans le traitement des images biomédicales. Ces modèles seront implémentés à l'aide des bibliothèques TensorFlow et Keras, en s'appuyant sur des techniques avancées telles que le transfert d'apprentissage afin d'améliorer les performances, même avec des ensembles de données restreints. L'interface utilisateur, développée sous Android Studio, offrira une expérience fluide et intuitive, permettant aux professionnels de capturer ou d'importer des images microscopiques et d'obtenir les résultats d'analyse en temps réel.

Le processus de classification repose sur plusieurs étapes clés :

1. Prétraitement des images pour améliorer la qualité visuelle et réduire le bruit.
2. Extraction et sélection des caractéristiques pour identifier les indicateurs morphologiques pertinents.
3. Classification à l'aide d'un modèle de Deep Learning, en comparant les performances des réseaux CNN.

L'un des principaux défis de ce projet réside dans l'obtention d'une précision élevée tout en assurant un temps de traitement rapide, indispensable pour une utilisation clinique efficace.

Ce travail s'inscrit dans la dynamique de transformation numérique du secteur médical, en rendant l'intelligence artificielle accessible aux professionnels de santé. En facilitant le

dépistage précoce et en réduisant les erreurs humaines, cette application mobile pourrait jouer un rôle essentiel dans l'amélioration de la qualité des soins apportés aux patients atteints de cancers du sang.

.1 Problématique

Le cancer du sang, notamment la leucémie, est une maladie grave nécessitant un diagnostic rapide et précis pour maximiser les chances de succès du traitement. Actuellement, l'analyse des frottis sanguins repose principalement sur l'expertise des hématologues, qui doivent examiner manuellement les cellules au microscope. Ce processus, long et complexe, est sujet à des erreurs humaines pouvant entraîner des retards de diagnostic et des interprétations subjectives.

Grâce aux avancées récentes en intelligence artificielle et en apprentissage profond (Deep Learning), il devient envisageable d'automatiser la détection et la classification des cellules anormales à partir d'images médicales. Toutefois, plusieurs défis subsistent :

- Comment concevoir une application mobile capable de classifier avec précision et rapidité les cancers du sang à partir d'images de frottis sanguins ?
- Comment optimiser l'utilisation des réseaux neuronaux convolutifs (CNN) pour garantir de bonnes performances malgré des bases de données limitées ?
- Comment proposer une interface intuitive et accessible aux professionnels de santé, tout en garantissant des résultats interprétables ?

.2 Objectifs et solution

.2.1 Objectifs du projet

L'objectif principal de ce projet est de développer une application mobile intelligente capable de détecter et de classifier automatiquement les cancers du sang à partir d'images de frottis sanguins. Cette solution vise à aider les professionnels de santé à poser un diagnostic rapide et précis, tout en réduisant le temps d'analyse et les risques d'erreurs humaines.

Les objectifs spécifiques sont les suivants :

- Automatiser la classification des cellules leucémiques à l'aide d'algorithmes de Deep Learning.
- Développer une application mobile sous Android Studio permettant de capturer, importer et analyser en temps réel des images de frottis sanguins.
- Optimiser les performances des modèles CNN en utilisant le transfert d'apprentissage et des techniques de régularisation pour améliorer la précision, même avec des jeux de données limités.
- Renforcer l'interprétabilité des résultats en fournissant des visualisations explicites et des scores de confiance pour chaque prédiction.
- Concevoir une interface ergonomique et intuitive adaptée aux exigences du personnel médical.

.2.2 Solution proposée

La solution proposée repose sur une approche combinant intelligence artificielle et traitement d'images biomédicales. Elle débute par l'acquisition et le prétraitement des images, incluant la collecte d'images de frottis sanguins, la normalisation, la réduction du bruit et l'amélioration de la qualité visuelle. Ensuite, des modèles de réseaux neuronaux convolutifs (CNN) sont utilisés pour l'apprentissage et la classification des cellules normales et cancéreuses. Cette phase est suivie par l'optimisation du modèle à travers la comparaison de différentes architectures CNN et l'exploitation du transfert d'apprentissage pour améliorer les performances. Parallèlement, une application mobile est développée sous Android Studio, avec une interface simple et intuitive intégrant les algorithmes d'analyse et facilitant l'interaction avec les utilisateurs. Enfin, l'évaluation et la validation de la solution sont réalisées à l'aide de bases de données réelles, avec des comparaisons face aux méthodes traditionnelles et des validations menées par des experts médicaux. Cette démarche vise à fournir aux médecins et aux laboratoires d'hématologie un outil performant et rapide pour le diagnostic précoce du cancer du sang, contribuant ainsi à améliorer la prise en charge des patients.

I

Classification du cancer du sang : concepts de base et état de l'art

I.1 Introduction

Les progrès récents dans le domaine de l'intelligence artificielle (IA) et de ses sous-domaines, notamment l'apprentissage automatique et l'apprentissage profond, ont transformé le paysage de la médecine moderne. Ces technologies offrent des opportunités sans précédent pour améliorer les pratiques médicales, allant de la recherche pharmaceutique à la prise de décision clinique en passant par le diagnostic assisté par ordinateur. Avec l'avènement des dossiers médicaux électroniques (DME), les données médicales sont désormais plus accessibles et exploitables que jamais. Entre 2007 et 2012, l'utilisation des DME aux États-Unis a quadruplé, passant de 11,8 % à 39,6 % chez les médecins exerçant en cabinet [4]. Parmi ces données figurent les images médicales, qui jouent un rôle essentiel dans le diagnostic et le suivi des patients [5, 6].

Cependant, l'analyse traditionnelle des images médicales repose principalement sur l'expertise humaine, ce qui présente plusieurs limitations [7]. La fatigue, la variabilité interindi-

viduelle et le coût élevé de formation des radiologues qualifiés peuvent influencer la précision et la rapidité des diagnostics. De plus, certains systèmes de santé externalisent les analyses radiologiques vers des pays à moindre coût, comme l'Inde, via la téléradiologie, ce qui peut introduire des délais ou des erreurs. Dans ce contexte, l'automatisation de l'analyse des images médicales grâce à des algorithmes d'apprentissage automatique représente une avancée majeure [5, 6]. Ces outils promettent non seulement d'améliorer la précision et l'efficacité des diagnostics, mais aussi de réduire les risques liés aux erreurs humaines, tout en optimisant les coûts et les délais de traitement.

I.2 Le cancer du sang et ses types

I.2.1 Introduction au cancer du sang

Le cancer du sang, bien qu'invisible à l'œil nu, est une maladie complexe qui touche des millions de personnes dans le monde. Parmi ses différentes formes, la leucémie se distingue comme l'une des pathologies les plus répandues, particulièrement chez les enfants. Cette maladie résulte d'une prolifération anormale et incontrôlée de cellules immatures du sang, perturbant ainsi le fonctionnement normal de la moelle osseuse, des globules rouges et du système immunitaire. Selon les données épidémiologiques, la leucémie représente environ 3,5 % des nouveaux cas de cancers aux États-Unis, avec plus de 60 000 nouveaux cas diagnostiqués en 2018 seulement. Ces cellules malignes, appelées lymphoblastes, peuvent migrer vers d'autres organes vitaux tels que la rate, le foie, le cerveau et les reins, entraînant des métastases et aggravant l'état du patient [8, 9].

Cependant les avancées médicales, le diagnostic précoce de la leucémie reste un défi majeur. Les symptômes initiaux, tels que la fièvre, la pâleur, la perte de poids ou l'hypertrophie des ganglions lymphatiques, sont souvent non spécifiques et peuvent être confondus avec d'autres affections. En outre, la nature insidieuse de la maladie, caractérisée par des signes légers dans ses stades précoce, complique encore davantage son identification rapide. Traditionnellement, le diagnostic repose sur une évaluation microscopique des frottis sanguins périphériques (PBS), tandis que l'analyse de la moelle osseuse reste l'étalon-or pour confirmer la présence de la leucémie [9, 10]. Cependant, ces méthodes nécessitent une expertise considérable et peuvent être sujettes à des erreurs humaines.

Face à ces défis, les progrès technologiques ont ouvert de nouvelles perspectives dans le domaine du diagnostic médical. Au cours des deux dernières décennies, l'apprentissage automatique (Machine Learning, ML) et les techniques d'analyse d'images assistées par ordinateur ont révolutionné la manière dont les médecins abordent les maladies hématologiques. Ces méthodes permettent non seulement de détecter la leucémie à un stade précoce, mais aussi de différencier ses sous-types avec une précision accrue. Par exemple, des études récentes ont

exploré l'utilisation d'algorithmes ML pour analyser des images de frottis sanguins, identifier les anomalies cellulaires et classer les différents types de leucémies [11, 12].

L'intelligence artificielle, et plus spécifiquement le Machine Learning (ML), repose sur des algorithmes capables d'apprendre à partir de grandes quantités de données sans instructions explicites. Ces outils ont montré des résultats extraordinaires dans l'extraction et l'analyse des caractéristiques des images médicales, facilitant ainsi les processus décisionnels complexes en médecine [13, 14]. Avec la multiplication des outils de diagnostic et la production croissante de données de haute qualité, il est désormais essentiel de développer des méthodes d'analyse plus avancées pour répondre aux besoins cliniques modernes. Les approches traditionnelles, limitées par leur incapacité à traiter de vastes volumes de données, laissent place à des solutions innovantes qui combinent science des données, imagerie médicale et intelligence artificielle pour transformer la prise en charge des patients atteints de leucémie [15, 16].

En somme, la lutte contre la leucémie et d'autres formes de cancer du sang repose aujourd'hui sur une synergie entre la biologie humaine et la technologie. Grâce à des innovations telles que l'apprentissage automatique, nous sommes sur le point de franchir une nouvelle étape dans la détection précoce, la classification précise et le traitement personnalisé de ces maladies complexes.

I.2.2 Les principaux types de cancers du sang

La moelle osseuse est responsable de la production des cellules sanguines. Sans aucun doute, l'un des organes les plus essentiels du corps est le système circulatoire. Le rôle du sang est de transporter l'oxygène et les nutriments vers les différentes parties du corps. Il contribue également à réguler la température corporelle et protège le corps grâce à la production d'anticorps. Le sang humain se compose de trois composants principaux : les globules rouges (GR), les globules blancs (GB) et les plaquettes (thrombocytes). Les GR transportent l'oxygène vers les tissus du corps et éliminent les déchets produits par ceux-ci. Les GB constituent une partie vitale du système immunitaire humain en défendant le corps contre les maladies et les envahisseurs étrangers. Les plaquettes sont un groupe de cellules sanguines qui empêchent les saignements [17]. Parmi les GB, il existe deux grands groupes fonctionnels : les lymphocytes B et T [18]. Les GB se divisent également en plusieurs catégories principales, telles que les basophiles, les éosinophiles, les lymphocytes, les monocytes, les neutrophiles, etc. [19–21]. La Figure I.1 Les cinq principaux types de globules blancs. Les composants du sang et les principaux types de GB sont illustrés dans la Figure I.2.

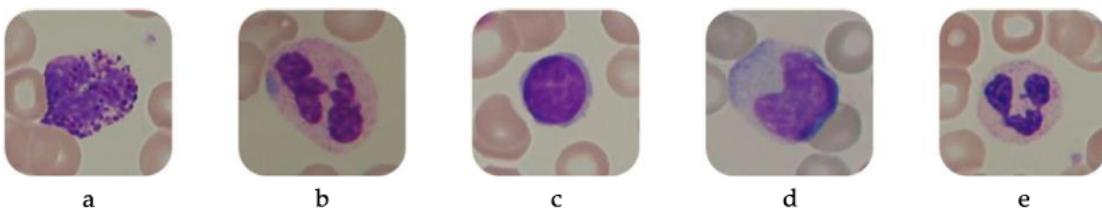


FIGURE I.1 – Les cinq principaux types de globules blancs : (a) basophile, (b) éosinophile, (c) lymphocyte, (d) monocyte, (e) neutrophile.

La leucémie est un type de cancer du sang qui commence généralement dans la moelle osseuse et provoque la production anormale de GB appelés blastes ou cellules leucémiques. Ces cellules perturbent la production normale des GB et diminuent la résistance du corps [22]. Les différents types de leucémies sont classifiés en fonction du type de GB affecté ainsi que selon la vitesse de progression de la maladie. La leucémie aiguë débute brutalement et progresse rapidement sur quelques jours ou semaines. En conséquence, l'état du patient s'aggrave rapidement et nécessite un traitement immédiat. Une leucémie aiguë non traitée peut entraîner une issue fatale en quelques mois. La leucémie chronique, quant à elle, évolue lentement et peut prendre des mois voire des années pour progresser [17, 23].

Il existe différents types de leucémies sanguines, et ici nous décrivons les quatre principaux types [24] :

1 : Acute Lymphoblastic Leukemia (ALL) :

La maladie commence particulièrement avec les lymphoblastes B. Ses principaux symptômes sont l'augmentation du nombre de cellules blastiques dans la moelle osseuse et la diminution des cellules sanguines normales. La prolifération rapide de lymphocytes anormaux est appelée lymphoblaste. Lorsque la leucémie lymphoblastique aiguë (LLA) survient, les globules blancs ne parviennent pas à mûrir complètement ; ce type de leucémie est plus fréquent chez les enfants. Le manque de traitement de cette maladie provoque la production excessive de lymphoblastes dans le corps, ce qui peut entraîner la mort [17, 25, 26]. La LLA survient plus fréquemment dans les lymphocytes B que dans les lymphocytes T. Pour la première fois, le groupe Franco-Américain-Britannique (FAB) a classifié la LLA selon des critères morphologiques, en la divisant en trois sous-groupes : L1, L2 et L3 [27-30].

2 : Acute Myeloid Leukemia (AML)

C'est un type de leucémie qui se développe dans les cellules de la moelle osseuse ou les myélocytes et progresse rapidement. Les myélocytes produisent des globules rouges (GR), des globules blancs (GB) (à l'exception des granulocytes et des lymphocytes), ainsi que des plaquettes. Dans cette maladie, la moelle osseuse produit des myéloblastes, des globules rouges

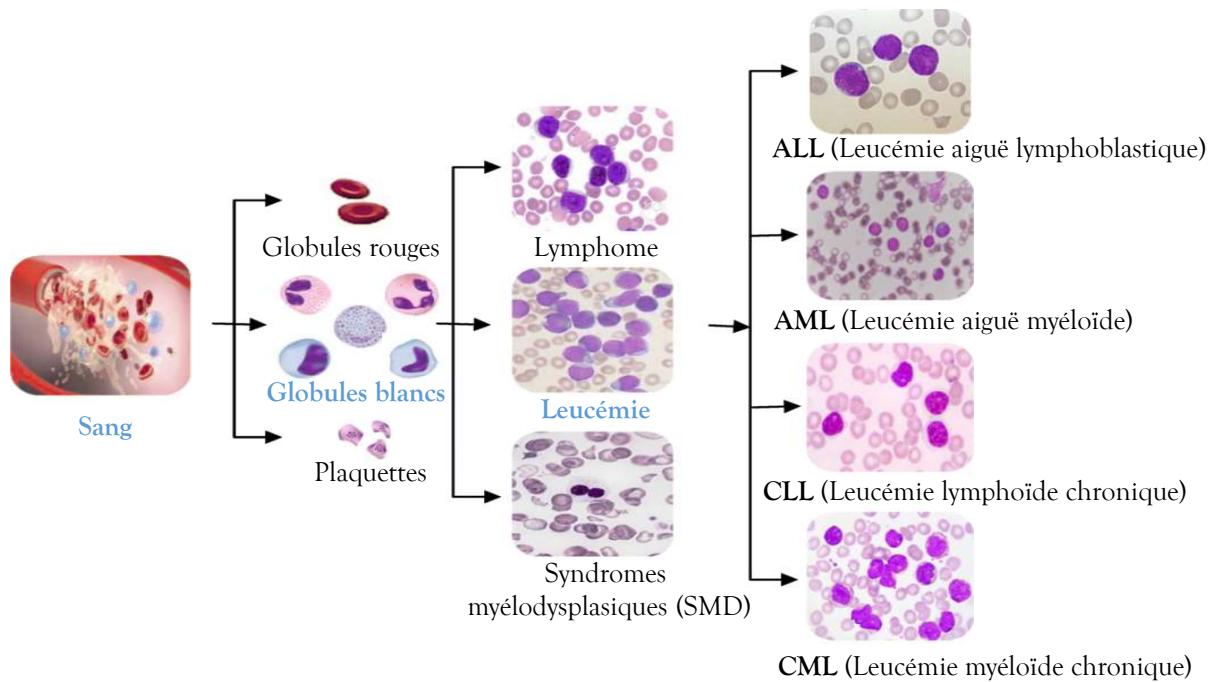


FIGURE I.2 – Constituants du sang et principaux types de leucémie

anormaux ou des plaquettes, ce qui en fait le trouble myéloïde malin le plus courant chez les adultes. La LAM est classée en huit sous-types, incluant M0, M1, M2, M3, M4, M5, M6 et M7 [31].

3 :Chronic Lymphoblastic Leukemia (CLL)

C'est le type de leucémie le plus courant, qui est plus fréquent chez les personnes âgées. Dans cette maladie, les cellules sanguines semblent matures mais ne sont pas complètement normales et sont incapables de combattre les cellules envahissantes. Elle peut également se propager aux ganglions lymphatiques et à des organes tels que le foie et la rate. La LLC progresse lorsqu'un nombre excessif de lymphocytes anormaux se développent, ce qui évince les cellules sanguines normales et affaiblit le système immunitaire [17].

4 : Chronic Myeloid Leukemia (CML)

C'est une maladie causée par une augmentation de la croissance et de la concentration des cellules myéloïdes de la moelle osseuse, ainsi que par une augmentation de leur densité dans le sang. La progression de la maladie est lente et contrôlable. Les patients atteints de LMC mènent une vie normale et sont souvent asymptomatiques [17].

Le diagramme en bloc des composants du sang et des principaux types de leucémie est illustré dans la Figure I.2

I.2.3 Caractéristiques observables dans les frottis sanguins

Les frottis sanguins sont des préparations microscopiques réalisées à partir d'une goutte de sang étalée sur une lame de verre. Ils constituent un outil diagnostique fondamental en hématologie, permettant l'observation directe des cellules sanguines et la détection d'anomalies potentielles. Voici une analyse détaillée des caractéristiques observables dans les frottis sanguins, basée sur des sources scientifiques et techniques [32, 33].

1 : Globules rouges (Érythrocytes)

Les globules rouges (GR) sont les cellules les plus abondantes dans le sang. Leur observation dans un frottis sanguin permet d'évaluer plusieurs caractéristiques importantes [32] :

- **Taille** : Normalement, les GR ont un diamètre moyen de 7 à 8 μm . Des variations de taille (anisocytose) peuvent indiquer des pathologies comme l'anémie.
- **Forme** : Les GR doivent être régulièrement circulaires avec une zone centrale claire (due à leur forme biconcave). Des anomalies morphologiques, telles que les cellules en "cible" ou les elliptocytes, peuvent signaler des troubles spécifiques [34].
- **Coloration** : La coloration appropriée (par exemple, avec la méthode de May-Grünwald-

Giems) révèle une teinte rose pâle uniforme. Une coloration anormale peut indiquer des défauts d'hémoglobine, comme dans les cas de thalassémie ou d'anémie ferri-prive [32].

2 : Globules blancs (Leucocytes)

Les globules blancs jouent un rôle central dans le système immunitaire. Leur observation permet de détecter des infections, des inflammations ou des maladies hématologiques graves [33, 35] :

Types cellulaires :

- **Neutrophiles** : Présents en grande quantité, ils ont un noyau segmenté et un cytoplasme granuleux. Une augmentation peut indiquer une infection bactérienne.
- **Lymphocytes** : Ils possèdent un noyau dense et un cytoplasme clair. Leur augmentation est souvent associée à des infections virales.
- **Monocytes** : Ce sont les plus grandes cellules sanguines, avec un noyau en forme de rein.
- **Éosinophiles et basophiles** : Ces cellules contiennent des granulations spécifiques dans leur cytoplasme. Les éosinophiles augmentent dans les allergies et les parasitoses, tandis que les basophiles sont impliqués dans les réactions allergiques sévères.

Anomalies morphologiques : La présence de blastes (cellules immatures) ou de noyaux atypiques peut être un signe de leucémie ou d'autres troubles hématologiques [36].

3 : Plaquettes

Les plaquettes sont visibles sous forme de petites structures rondes ou allongées, généralement regroupées en agrégats [32] :

- **Nombre** : Une diminution du nombre de plaquettes (thrombopénie) peut être liée à des troubles de la coagulation.
- **Morphologie** : Des plaquettes géantes ou anormalement petites peuvent indiquer des anomalies de production ou de fonctionnement [34].

4 : Parasites et autres éléments

Dans certaines situations cliniques, les frottis sanguins peuvent également révéler la présence de parasites ou d'autres anomalies [33] :

- **Parasites** : Par exemple, Plasmodium (responsable du paludisme) peut être observé dans les GR sous forme de corps intracellulaires.
- **Corps inclus** : Certains inclusions, comme les corps de Howell-Jolly ou les anneaux de Cabot, peuvent être présents dans des conditions pathologiques spécifiques.

5 : Aspects techniques

Pour garantir une observation précise des caractéristiques ci-dessus, plusieurs facteurs techniques doivent être pris en compte [34] :

- **Qualité du frottis** : Un frottis bien réalisé doit être uniformément étalé et séché pour éviter les déformations cellulaires.
- **Coloration** : Les méthodes de coloration standardisées, comme Wright et Giemsa, permettent une meilleure visualisation des cellules.
- **Microscopie** :
 - À faible grossissement : Permet de vérifier la qualité générale du frottis et la distribution des cellules.
 - À fort grossissement : Utilisé pour examiner les détails morphologiques spécifiques des cellules.

Applications cliniques :

L'analyse des caractéristiques observables dans les frottis sanguins a des implications diagnostiques majeures :

- Détection des infections : Augmentation des neutrophiles en cas d'infections bactériennes, ou des lymphocytes en cas d'infections virales [32].
- Diagnostic des maladies hématologiques : Identification des anomalies cellulaires dans les leucémies, les anémies ou les thrombopénies [36].
- Suivi des traitements : Évaluation de l'efficacité des thérapies par chimiothérapie ou immunosupresseurs [35].

I.3 Applications de l'IA en médecine

L'intégration de l'intelligence artificielle (IA) dans le domaine médical marque une révolution technologique aux impacts considérables. Grâce à ses sous-domaines, tels que l'apprentissage automatique et l'apprentissage profond, l'IA ouvre de nouvelles perspectives pour optimiser les soins de santé, accélérer les diagnostics et appuyer la prise de décision clinique. La multiplication des données issues des dossiers médicaux électroniques (DME) et des dispositifs d'imagerie médicale permet désormais de concevoir des systèmes intelligents capables d'analyser, d'interpréter et d'anticiper les pathologies avec une précision remarquable.

En particulier, l'analyse des images médicales, historiquement confiée à l'expertise humaine, bénéficie aujourd'hui d'outils automatisés capables de réduire la variabilité des diagnostics et de pallier les contraintes liées au coût et à la disponibilité des spécialistes. Ces technologies s'avèrent précieuses face aux défis actuels du secteur médical, notamment l'augmentation des volumes de données et la nécessité de décisions cliniques rapides et fiables.

Ainsi, les applications de l'IA en médecine ne cessent de s'étendre : du diagnostic assisté par ordinateur à la personnalisation des traitements, en passant par le suivi prédictif des patients et la recherche biomédicale. Toutefois, malgré ces avancées, des enjeux subsistent, tels

que l'interprétabilité des résultats des modèles et la nécessité de bases de données médicales volumineuses et diversifiées pour garantir la fiabilité des systèmes.

I.3.1 Convolutional neural network(CNN)

I.3.1.1 *Introduction CNN*

Le CNN (Convolutional Neural Network) est le schéma d'apprentissage profond le plus populaire. C'est un réseau neuronal composé de multiples couches, qui ressemble au cortex visuel. Il fonctionne mieux pour les données multidimensionnelles corrélées, comme les images, et aide ainsi à extraire les caractéristiques les plus pertinentes sans corrélation. Dans un CNN, les poids sont partagés de manière adaptative pour effectuer des opérations de convolution sur une image, contrairement au perceptron multicouche (Multi-layer Perceptron, MLP) [37].

I.3.1.2 *Convolutional neural network (CNN) classique*

Dans un CNN classique, les couches de convolution extraient des caractéristiques génériques, tandis que les couches plus profondes sont intégrées avec des caractéristiques spécifiques à la cible. Ainsi, les caractéristiques sont progressivement transformées de génériques à spécifiques à la cible [38]. Il se compose de différentes couches : une couche d'entrée, une couche convulsive, une couche de pooling, une couche entièrement connectée et une couche de classification, comme illustré à la Figure I.3.

a : COUCHE D'ENTRÉE

Dans cette couche d'un modèle CNN, les entrées (images) sont fournies. La taille des entrées est également définie ici.

b : COUCHE CONVOLUTIVE

Comme son nom l'indique, une opération de convolution est effectuée dans cette couche. Elle est réalisée en effectuant une multiplication scalaire entre les pixels du noyau (kernel) et les pixels respectifs de l'image sur lesquels le noyau est actuellement positionné, puis en additionnant tous ces résultats pour évaluer une sortie de convolution à un pixel particulier. Ensuite, le noyau se déplace sur toute l'image selon le pas (stride) prédéfini. Son objectif est d'extraire diverses caractéristiques de la couche précédente [39].

c : COUCHE ReLu

ReLU signifie Rectified Linear Unit (Unité Linéaire Rectifiée). Cette couche est utilisée pour améliorer la non-linéarité dans le modèle CNN. Ici, des fonctions d'activation ReLU sont

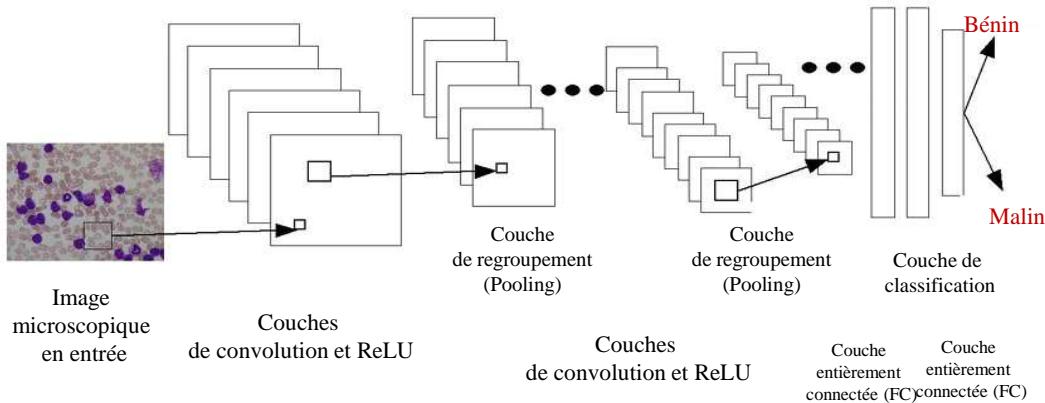


FIGURE I.3 – Architecture CNN conventionnelle

employées pour ajouter de la non-linéarité, au lieu des fonctions d'activation traditionnelles comme tanh ou sigmoïde [1]. Contrairement à ces deux fonctions, ReLU ne sature pas près de 1. Plus important encore, elle accélère le réseau tout en maintenant une précision similaire en améliorant la vitesse d'apprentissage [39], [1]. C'est une fonction d'activation populaire car elle est facile à utiliser et réussit à éliminer les inconvénients d'autres fonctions d'activation autrefois populaires.

d : COUCHE DE POOLING

Le pooling réduit la dimensionnalité des images pour minimiser les coûts de calcul et accélérer l'apprentissage. Il est également utilisé pour atténuer les problèmes de sur-apprentissage (overfitting). Dans un CNN, Max Pooling et Average Pooling sont deux méthodes populaires de pooling [39].

- (i) **Max Pooling** : Il renvoie la valeur maximale de la région de l'image sur laquelle le noyau de pooling est appliqué. Son rôle est de supprimer le bruit en rejetant toutes les activations bruyantes, tout en réduisant la dimensionnalité [39].
- (ii) **Average Pooling** : Il renvoie la moyenne des valeurs de la région de l'image sur laquelle le noyau de pooling est appliqué [39].

e : COUCHE ENTIEREMENT CONNECTEE

Une ou plusieurs couches entièrement connectées (fully connected layers) sont ajoutées après les couches convolutives et de pooling dans le modèle CNN. Comme son nom l'indique, chaque neurone de cette couche est connecté à tous les neurones de la couche précédente. Cette couche intègre toutes les informations apprises par les couches précédentes pour détecter des motifs plus larges. La dernière couche entièrement connectée intègre les informations (caractéristiques) pour la classification d'images. Ainsi, la taille de sortie de la dernière couche entièrement connectée est choisie comme étant égale au nombre de classes que nous souhaitons classifier dans les données d'entrée. Une activation SoftMax est généralement utilisée dans les dernières couches entièrement connectées pour déterminer les probabilités d'appartenance à une classe [39].

f : COUCHE DE CLASSIFICATION

C'est la dernière couche du modèle CNN, qui exploite les probabilités estimées par la fonction d'activation SoftMax pour chaque entrée afin de prédire efficacement la classe [39].

Contributions spécifiques

Banik et al. [40] ont proposé une méthode de classification de la LLA (leucémie lymphoblastique aiguë) basée sur un CNN. Dans cette méthode, les caractéristiques des premières et dernières couches sont combinées. Une couche de dropout est utilisée pour éviter le surapprentissage.

En 2020, **Jha et Dutta** [41] ont présenté un modèle de CNN profond basé sur un algorithme chronologique Sinus-Cosinus (SCA) pour classer avec succès la LLA.

Shahin et al. [42] ont proposé un nouveau cadre basé sur un CNN (WBCsNet) pour classifier efficacement les globules blancs (WBC).

Claro et al. [43] ont suggéré un réseau d'apprentissage profond (Alert-Net). Il se compose de 5 couches convolutives, deux couches entièrement connectées et une couche SoftMax.

Limites des CNN classiques

Un CNN conventionnel nécessite un grand ensemble de données pour atteindre des performances exceptionnelles. Cependant, les ensembles de données médicaux standard disponibles publiquement sont souvent de petite taille, ce qui empêche un ajustement précis des poids pour extraire efficacement des caractéristiques ciblées importantes. Par conséquent, cela entraîne des performances relativement médiocres.

I.3.2 Apprentissage par transfert (Transfer Learning)

Récemment, l'apprentissage par transfert est apparu comme une approche en pleine expansion dans le domaine de l'imagerie médicale en raison de ses excellentes performances [44],

[37], [42], [43]. L'apprentissage par transfert présente l'avantage de ne pas nécessiter un grand ensemble de données pour l'entraînement, car il utilise un réseau pré-entraîné et transfère ses connaissances ou ses poids vers des tâches spécifiques au domaine cible (tâches d'imagerie médicale). Par conséquent, dans le domaine cible, seul un ajustement fin (fine-tuning) de ces réseaux pré-entraînés est nécessaire. Ainsi, il offre des performances exceptionnelles, même pour les petits ensembles de données [44], [37], [42], [43]. Dans ce contexte, nous discutons ici de certaines approches populaires d'apprentissage par transfert : AlexNet [1], CaffNet [45], VGGNet [46], GoogLeNet [47], ResNet [48], MobileNet [2] et Xception [49].

I.3.2.1 AlexNet

AlexNet est une technique d'apprentissage par transfert très connue. Elle a été développée par Krizhevsky et al. [1] en 2012. Ce modèle comprend cinq couches convolutives et trois couches entièrement connectées, comme illustré à la Figure I.4. Chaque couche convulsive est suivie par des activations ReLU (pour ajouter de la non-linéarité) et une couche de max-pooling (pour réduire le surapprentissage, ou overfitting) [44], [50], [51], [1]

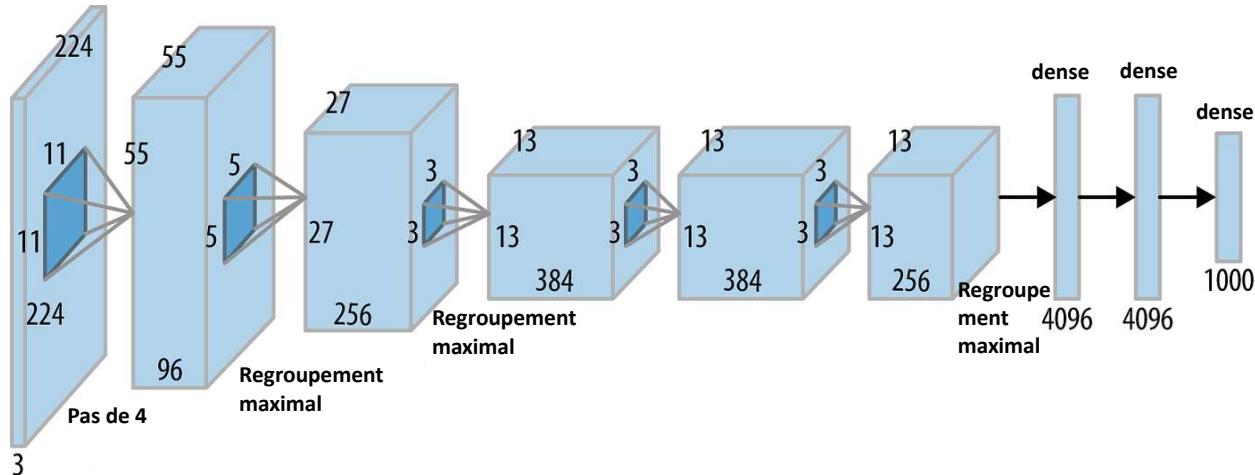


FIGURE I.4 – Architecture AlexNet [1]

I.3.2.2 CaffNet

CaffNet a été créé par le Berkeley Vision and Learning Center (BVLC). Similaire à AlexNet, il comprend cinq couches convolutives et trois couches entièrement connectées. Cependant, il diffère d'AlexNet en termes de l'ordre des couches de pooling et de normalisation [44], [45].

I.3.2.3 VGGNet

VGGNet est devenu populaire en 2014 grâce à son architecture uniforme (il utilise exclusivement des filtres de taille 3×3). Dans ce modèle, un bloc de filtres est utilisé à la place d'une seule couche convective. Cette approche augmente les non-linéarités et optimise le champ récepteur en appliquant des convolutions 3×3 successives, comme illustré à la Figure I.5. La succession de deux couches de convolution 3×3 et trois couches de convolution 3×3 produit respectivement des champs récepteurs équivalents à des convolutions 5×5 et 7×7 . VGGNet gère environ 138 millions de paramètres. Les variantes VGG16 et VGG19 comportent respectivement 16 et 19 couches pondérées [46].

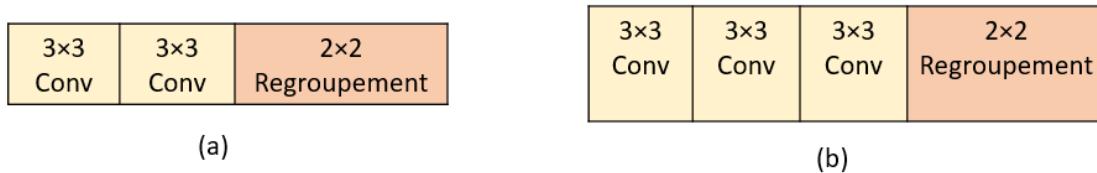


FIGURE I.5 – Blocs de convolution : (a) Bloc avec deux couches de convolution 3×3 , (b) Bloc avec trois couches de convolution 3×3 .

I.3.2.4 GoogLeNet

Szegedy et al. [47] ont présenté une nouvelle architecture appelée GoogLeNet, qui utilise un module inception pour la réduction de la dimensionnalité. Le module inception utilise un bloc convolutif contenant des filtres de différentes tailles. Le module inception est présenté à la Figure I.6 (b), un module inception efficace est proposé dans l'architecture GoogLeNet [47]. Il devient computationnellement efficace en appliquant des convolutions 1×1 avant les convolutions 3×3 et 5×5 , ce qui donne lieu à une architecture de type bottleneck (goulot d'étranglement).

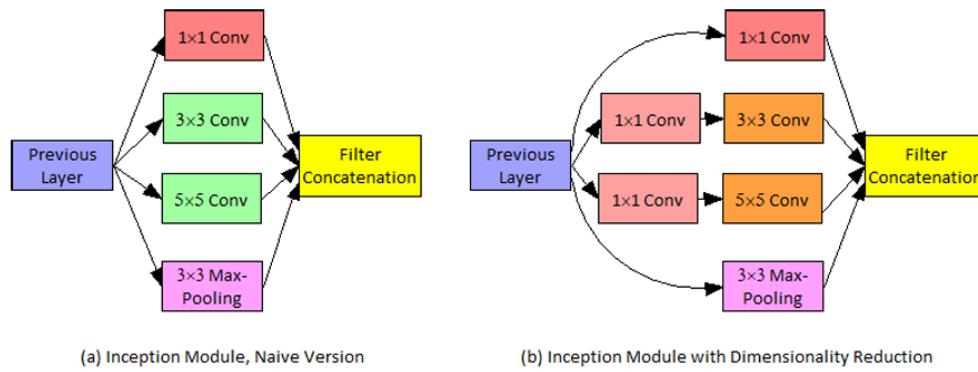


FIGURE I.6 – Module de démarrage.

I.3.2.5 ResNet

ResNet a été créé par He et al. [48] en 2015, un réseau beaucoup plus profond en raison de son grand nombre de couches. Ils ont proposé l'utilisation de connexions de contournement (skip-connections) pour surmonter les problèmes de disparition du gradient, offrant ainsi une voie alternative pour la propagation du gradient, comme le montre la Figure I.7. Le facteur résiduel, $F(x)$, est présenté comme suit :

$$F(x) = H(x) - x \quad (I.1)$$

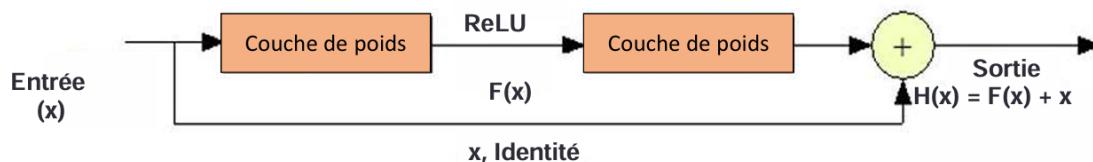


FIGURE I.7 – Bloc d'apprentissage résiduel.

où x et $H(x)$ symbolisent respectivement l'entrée et la sortie du bloc résiduel.

Plus important encore, dans ce modèle, ils ont recommandé d'apprendre le facteur résiduel $F(x)$ plutôt que la sortie du bloc résiduel $H(x)$, ce qui entraîne un apprentissage plus rapide et un système plus rapide, car $F(x)$ est très petit par rapport à $H(x)$.

I.3.2.6 MobileNet

Wang et al. [2] ont créé une architecture légère appelée MobileNet. Une convolution séparable en profondeur est réalisée en appliquant une convolution en profondeur avant une convolution ponctuelle (point-wise convolution), rendant ainsi le système computationnellement efficace, comme le montre la Figure I.8 [2, 3]. Dans la convolution en profondeur, une convolution est effectuée individuellement sur chaque canal d'entrée. La convolution ponctuelle est responsable de combiner linéairement les sorties de la convolution en profondeur. Ils ont introduit des muplicateurs de largeur et de résolution pour rendre le modèle plus rapide [2]. De plus, la largeur est utilisée pour produire efficacement un réseau plus épais ou plus fin [2].

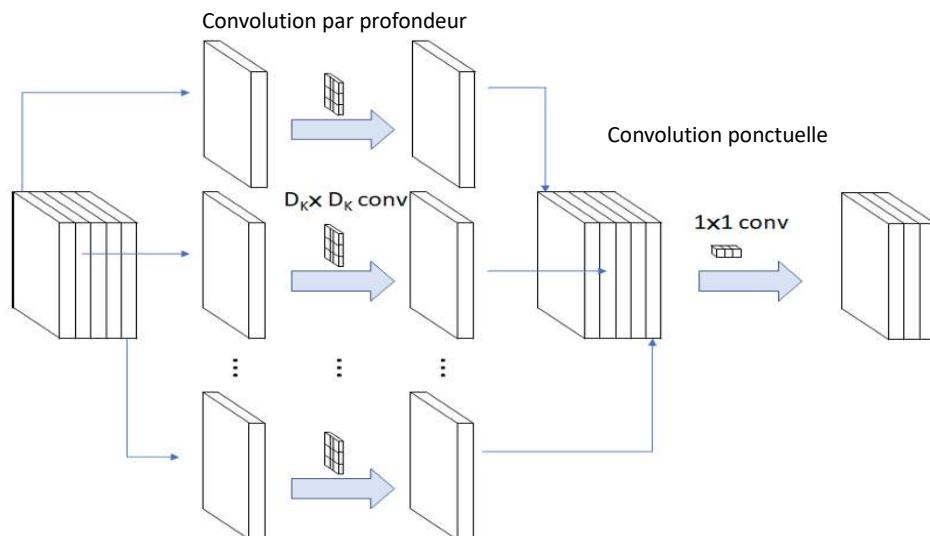


FIGURE I.8 – Convolution séparable en profondeur [2], [3].

I.3.2.7 XCEPTION

Xception est une méthode d'apprentissage par transfert basée sur des convolutions séparables en profondeur. Il se compose de 14 modules et contient au total 36 couches convolutives. Tous ces modules, à l'exception du premier et du dernier, ont des connexions résiduelles linéaires [49].

Contributions spécifiques :

Sha que et Tehsin [52] ont utilisé AlexNet [1] pour classer les globules blancs (WBC) en quatre classes : sains, L1, L2, L3. Ce modèle ne classe pas seulement les cas sains et la leucémie lymphoblastique aiguë (ALL), mais aussi efficacement les trois sous-types de l'ALL. Yu et al. [53] Ont présenté une approche de détection de l'ALL basée sur un ensemble d'apprentissage par transfert. Ils ont combiné VGG-16 [46], InceptionV3 [47], VGG-19 [46], ResNet50 [48] et Xception [49] pour classifier efficacement les WBC.

Shahin et al. [42] Ont appliqué des versions ajustées de LENet [54] et AlexNet [1] pour une classification efficace des WBC.

Mallick et al. [55] Ont développé un modèle CNN à cinq couches pour classer l'ALL et la leucémie myéloïde aiguë (AML) basé sur un CNN profond à cinq couches.

Roy et Ameer [51] Ont utilisé AlexNet [1] pour classifier efficacement les types de cellules sanguine leont présenté un réseau basé sur l'apprentissage profond appelé Alert-Net. Ils ont proposé deux réseaux hybrides : Alert-Net-R et Alert-Net-X, en combinant Alert-Net avec ResNet [48] et Xception [49], respectivement. Ils ont encore modifié ces deux réseaux en supprimant une couche de dropout, ce qui a donné naissance à deux nouveaux réseaux : Alert-Net-RWD et Alert-Net-XWD .

I.4 Conclusion

En conclusion, ce chapitre a exploré les bases de la classification des cancers du sang et les applications innovantes de l'IA pour améliorer le diagnostic médical. Les cancers sanguins ont été définis, et les techniques d'étude des frottis sanguins ont été détaillées afin de normaliser les méthodes de diagnostic. Le développement de l'IA, en particulier des réseaux neuronaux convolutifs (CNN), a permis de faire progresser le diagnostic des cancers du sang. Enfin, des outils technologiques innovants ont été présentés pour concevoir des applications intelligentes dans le domaine médical. Ces avancées offrent de nouvelles perspectives pour des soins de santé plus précis et efficaces, ouvrant ainsi la voie à une médecine plus personnalisée et performante.

II

Collecte et Prétraitement des Données

II.1 Introduction

Le succès des modèles d'apprentissage profond pour la classification d'images médicales dépend en grande partie de la qualité et de la représentativité des données utilisées lors de leur phase d'entraînement. Dans le contexte de la détection et de la classification des cancers du sang à partir de frottis sanguins, l'acquisition de données pertinentes ainsi que leur préparation rigoureuse constituent des étapes cruciales du processus de développement.

Ce chapitre présente les différentes sources de données exploitées dans ce projet, en mettant l'accent sur les bases de données publiques reconnues. Il décrit également les techniques de prétraitement appliquées pour améliorer la qualité des images et faciliter l'apprentissage automatique, telles que la normalisation, la réduction du bruit et l'augmentation des données. Enfin, une analyse exploratoire est réalisée afin d'évaluer la distribution et les caractéristiques des images collectées, dans le but d'orienter les choix méthodologiques pour l'entraînement du modèle.

II.2 Sources de données

Lors du développement d'une application mobile de classification des leucémies à l'aide d'algorithmes d'apprentissage profond, la sélection des sources de données est un facteur clé pour garantir l'exactitude et l'efficacité du modèle. Le succès d'un modèle de classification dépend de la qualité et de la variété des données utilisées pendant la phase de formation et de test. Nous nous sommes donc appuyés sur des ensembles de données publiques spécialisées en imagerie médicale des cellules sanguines, qui fournissent des images annotées de cellules saines et cancéreuses.

Ces données visent à fournir au modèle une quantité suffisante d'échantillons, ce qui contribue à améliorer sa capacité de généralisation lors de la rencontre de nouvelles images dans un environnement d'application réel. La sélection repose sur des critères tels que la résolution de l'image, la diversité des échantillons et la disponibilité de données annotées par des experts du domaine médical.

Parmi les bases de données les plus utilisées dans ce domaine, nous avons sélectionné des ensembles de données publiques spécialisées dans le diagnostic de la leucémie.

II.2.1 Blood Cells Image Dataset

Les ensembles de données d'images de cellules sanguines (WBC, White Blood Cells) jouent un rôle crucial dans l'étude du système immunitaire, car leur analyse permet de diagnostiquer des maladies graves telles que la leucémie. Grâce aux avancées de l'apprentissage profond, en particulier les réseaux de neurones convolutifs (CNN), il est désormais possible d'analyser automatiquement des images microscopiques de cellules sanguines.

Pour répondre à ces besoins, plusieurs jeux de données ont été développés, tels que BCCD, Raabin WBC et PBC, qui contiennent des images annotées selon les différents types cellulaires. Ces ensembles sont indispensables pour l'entraînement et l'évaluation de modèles performants de classification cellulaire, certains atteignant des taux de précision supérieurs à 99 %.

L'ensemble de données étudié s'inscrit dans le domaine de l'hématologie, plus précisément dans le cadre du développement d'outils informatiques pour le diagnostic hématologique à l'aide d'images microscopiques de cellules sanguines et de méthodes d'apprentissage automatique. Les données sont constituées d'images numériques de cellules sanguines périphériques normales, obtenues à partir d'échantillons collectés au laboratoire central de l'Hospital Clinic de Barcelone. Les analyses sanguines ont été effectuées avec l'instrument Advia 2120, et les frottis ont été préparés automatiquement à l'aide du Sysmex SP1000i en utilisant le colorant May Grünwald-Giemsa. Les images individuelles des cellules ont été capturées à l'aide de l'analyseur Cellavision DM96, au format JPG avec une résolution de 360 × 363

pixels. Ces données, brutes, proviennent d'individus présentant des paramètres de laboratoire normaux. La collecte s'est déroulée sur une période de quatre ans, de 2015 à 2019, dans le cadre de la routine hospitalière quotidienne. Toutes les images ont été anonymisées à l'aide de numéros aléatoires pour respecter la confidentialité des patients. La source de ces données est l'Hospital Clinic de Barcelone, situé en Catalogne, en Espagne. Le jeu de données est accessible via le dépôt Mendeley sous le nom : A dataset for microscopic peripheral blood cell images for development of automatic recognition systems, avec l'identifiant DOI : 10.17632/snkd93bnjr.1. Il peut être consulté directement via le lien suivant : <https://data.mendeley.com/datasets/snkd93bnjr/draft?a=d9582c71-9af0-4e59-9062-df30df05a121>.

Le jeu de données de sang périphérique normal contient un total de 17 092 images de cellules individuelles, qui ont été acquises à l'aide de l'analyseur Cellavision DM96. Toutes les images ont été obtenues dans l'espace de couleur RVB. Le format et la taille des images sont respectivement jpg et 360×363 pixels, et ont été étiquetées par des pathologistes cliniques de l'Hôpital Clinic.

Le jeu de données est organisé en huit groupes de différents types de cellules sanguines, comme indiqué dans le Tableau II.1. Bien que le groupe des granulocytes immatures inclue des myélocytes, métamyélocytes et promyélocytes, nous avons regroupé tous ces éléments dans un seul groupe pour deux raisons principales : (1) l'identification individuelle de sous-groupes spécifiques ne présente pas d'intérêt particulier pour le diagnostic ; et (2) les différences morphologiques entre ces groupes sont subjectives, même pour les pathologistes cliniques.

TABLE II.1 – Types et nombre de cellules dans chaque groupe (Blood Cells Image Dataset).

Type de cellule	Nombre total d'images	Pourcentage de distribution (%)	Dimensions de l'image
Neutrophiles	3329	19.48	360×363
Éosinophiles	3117	18.24	360×363
Basophiles	1218	7.13	360×363
Lymphocytes	1214	7.10	360×363
Monocytes	1420	8.31	360×363
Granulocytes immatures (métamyélocytes, myélocytes, promyélocytes)	2895	16.94	360×363
Érythroblastes	1551	9.07	360×363
Plaquettes (thrombocytes)	2348	13.74	360×363
Total	17,092	100	-

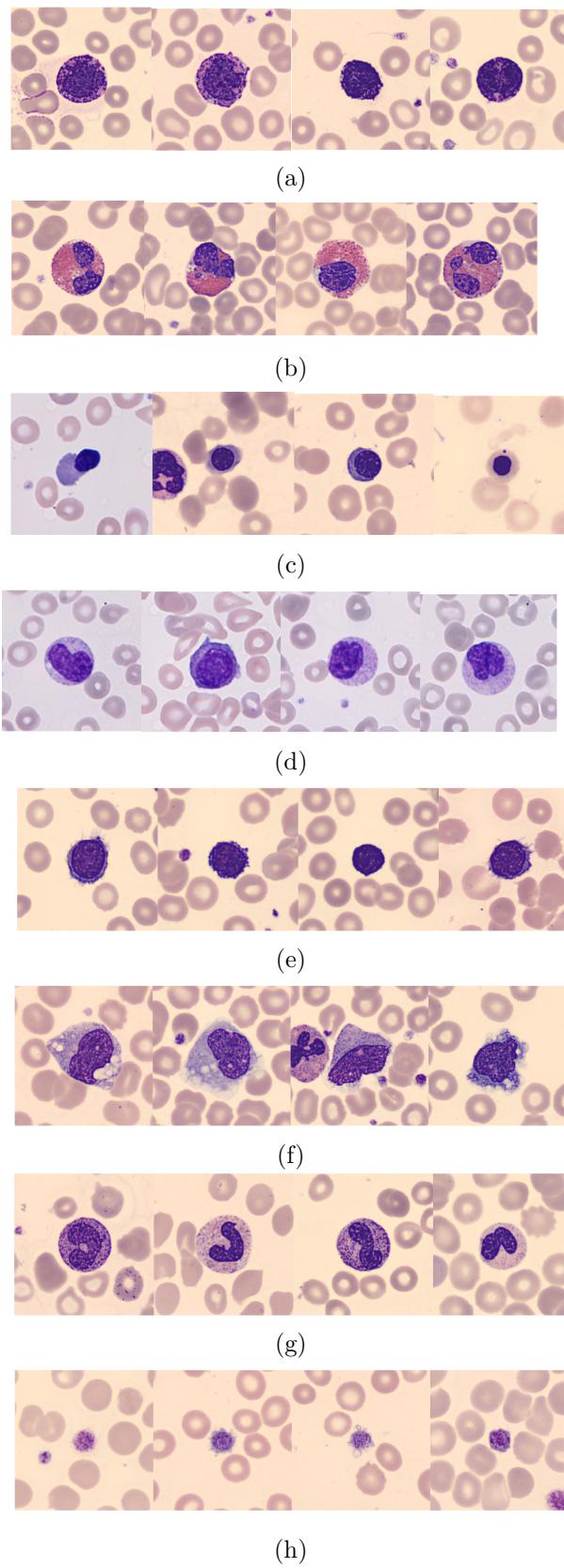


FIGURE II.1 – Blood Cells Image Dataset ; (a) Basophils, (b)Eosinophils, (c)Erythroblasts , (d) Immature Granulocytes , (e) Lymphocytes, (f)Monocytes , (g)Neutrophils , (h)Platelets ,

II.2.2 Lymphoblastic Leukemia (ALL)

La Acute Lymphoblastic Leukemia de type B (ALL-B) est l'un des types de leucémie les plus courants, touchant des individus de tous âges. Son diagnostic précis repose sur l'utilisation de plusieurs outils et méthodes, notamment l'aspiration de moelle osseuse (BMA) ou la biopsie médullaire (BMB), des procédures chirurgicales de laboratoire complexes et coûteuses.

Étant donné que la majorité des patients atteints de ALL-B sont des enfants, ces méthodes peuvent engendrer des complications et nécessitent souvent plusieurs jours avant l'obtention des résultats. Une analyse précoce des frottis sanguins au microscope par le personnel de laboratoire est donc essentielle. Toutefois, un grand nombre d'échantillons à traiter peut accroître le risque d'erreurs diagnostiques.

La détection de la ALL-B est également sujette à des imprécisions dues au temps nécessaire à l'analyse, à l'expérience du personnel médical et à la fatigue accumulée. Par ailleurs, certaines maladies immunitaires et inflammatoires présentent des symptômes similaires à ceux de la ALL-B, ce qui peut conduire à des confusions diagnostiques.

Les hématogones, cellules bénignes présentes dans la moelle osseuse, ont une apparence et une morphologie proches de celles des lymphoblastes cancéreux de la ALL-B, mais ne nécessitent pas de traitement spécifique. Il est souvent difficile de les distinguer uniquement sur la base de la morphologie cellulaire. C'est pourquoi la cytométrie en flux sur des échantillons de moelle osseuse est généralement utilisée comme méthode de référence pour confirmer le diagnostic.

Nous avons utilisé l'ensemble de données d'images de Lymphoblastic Leukemia (ALL). Les images de cet ensemble de données [56] [57] ont été préparées au laboratoire de moelle osseuse de l'hôpital Taleqani (Téhéran, Iran). Cet ensemble de données comprenait 3256 images de frottis sanguins périphériques (PBS) de 89 patients suspectés de ALL, dont les échantillons de sang ont été préparés et colorés par un personnel de laboratoire compétent. Parmi ces patients, 25 étaient en bonne santé, avec un diagnostic bénin (hématogone), et 64 avaient un diagnostic définitif de sous-types de ALL, à savoir ALL pré-B précoce, pré-B et pro-B. Cet ensemble de données est divisé en deux classes : bénignes et malignes. La première comprend les lymphoblastes hématogènes, et la seconde inclut le groupe ALL avec trois sous-types de lymphoblastes malins : ALL pré-B précoce, pré-B et pro-B. Toutes les images ont été prises à l'aide d'une caméra Zeiss dans un microscope avec un grossissement de 100x et enregistrées sous forme de fichiers JPG. Un spécialiste utilisant l'outil de cytométrie en flux a déterminé de manière définitive les types et sous-types de ces cellules. Après une segmentation basée sur le seuillage des couleurs dans l'espace colorimétrique HSV, les collecteurs de données fournissent également des images segmentées. La tableau II.2 illustre des exemples d'images de cellules bénignes (hématogones), ainsi que des cellules malignes des sous-types Early Pre-B

ALL, Pre-B ALL et Pro-B ALL.

Nous avons utilisé l'ensemble de données sur la Lymphoblastic Leukemia (ALL) pour former et valider notre modèle DNN. Il s'agit d'un ensemble de données multiclassées contenant des cellules bénignes et 3 sous-types de cellules blastiques appelées Early Pre-B, Pre-B et Pro-B. Un modèle DNN formé avec cet ensemble de données ALL peut aider à détecter les sous-types de Lymphoblastic Leukemia en classant ces différents sous-types de cellules.

Type	Classes	Nombre total d'images	Échantillons par classe	Dimensions de l'image
Bénin	Hématogones	504	25	224 × 224
Malin	Early Pre-B ALL	985	20	224 × 224
	Pre-B ALL	963	21	224 × 224
	Pro-B ALL	804	23	224 × 224
Total	-	3256	89	-

TABLE II.2 – Distribution des images dans la base de données Acute Lymphoblastic Leukemia (ALL).

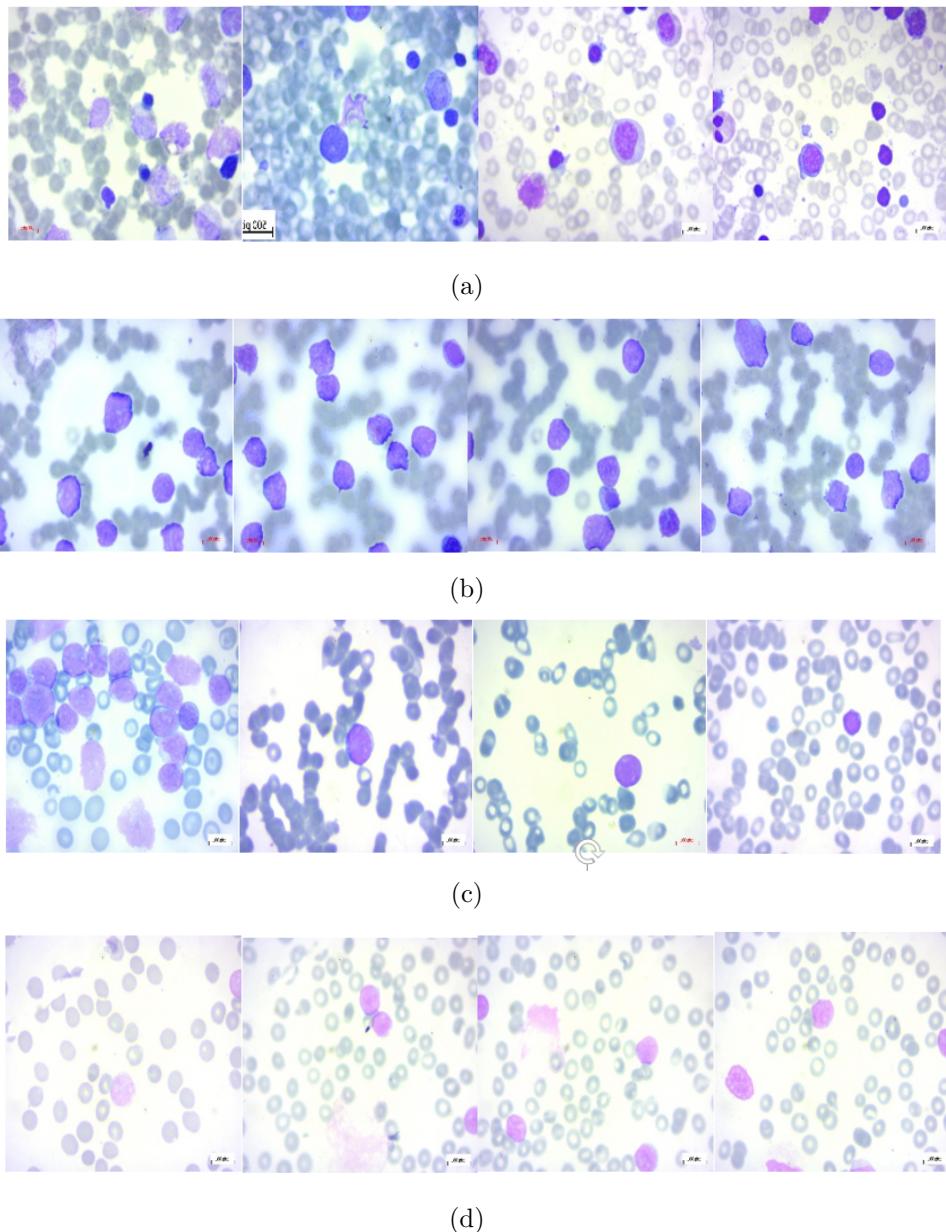


FIGURE II.2 – La base de données ALL-B, (a) Benign (hematogone) , (b) Early Pre-B ALL , (c) Pre-B , (d) Pro-B ALL .

II.3 Prétraitement des données

Le prétraitement des données constitue une étape cruciale dans le développement de modèles d'apprentissage profond, en particulier dans le domaine de la classification d'images médicales. Il vise à améliorer la qualité des données et à optimiser les performances des algorithmes de classification.

II.3.1 Augmentation des données

L'augmentation des données consiste à modifier les données d'entraînement d'origine afin de générer davantage de données artificielles, augmentant ainsi les chances d'exposer le modèle à un plus large éventail de caractéristiques pendant l'entraînement. Les techniques d'augmentation d'image se divisent généralement en deux catégories principales : les distorsions photométriques, qui modifient la teinte, la saturation, le contraste et la luminosité, et les distorsions géométriques, qui incluent la rotation, la translation, le redimensionnement, le cisaillement, le zoom avant ou arrière, ainsi que le retournement horizontal ou vertical des images.

Afin de mettre en œuvre ces techniques d'augmentation de données, nous avons utilisé la bibliothèque TensorFlow Keras, qui fournit des outils essentiels pour la prétraitement des images, notamment via la classe `ImageDataGenerator`. Le code ci-dessous montre comment ces transformations ont été intégrées dans notre pipeline de traitement des images, en appliquant rapidement des augmentations aux images, telles que le chargement des données et l'application des transformations. Cela permet non seulement de diversifier les données, mais aussi d'améliorer la robustesse du modèle face aux variations présentes dans les images réelles. La Figure II.3 illustre des exemples d'images pour augmentation des données.

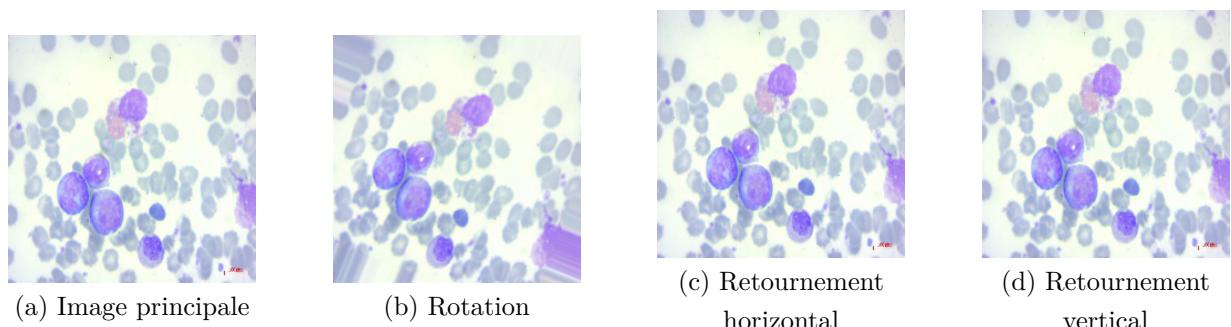


FIGURE II.3 – La rotation, le retournement horizontal et le retournement vertical

```

1 from tensorflow.keras.preprocessing.image import ImageDataGenerator
2
3 IMG_SIZE = (224, 224)
4 BATCH_SIZE = 16

```

```
5
6 train_dir = r'E:\MalikPersonel\D\enseignement\2024-2025\Rihana\Programmes\Prog2\output
7     \train'
8 val_dir = r'E:\MalikPersonel\D\enseignement\2024-2025\Rihana\Programmes\Prog2\output\
9     val'
10 test_dir = r'E:\MalikPersonel\D\enseignement\2024-2025\Rihana\Programmes\Prog2\output\
11     test'
12
13 train_datagen = ImageDataGenerator(
14     rescale=1./255,
15     rotation_range=15,
16     width_shift_range=0.1,
17     height_shift_range=0.1,
18     zoom_range=0.1,
19     shear_range=0.1,
20     brightness_range=[0.8, 1.2],
21     horizontal_flip=True,
22     vertical_flip=True,
23     fill_mode='nearest'
24 )
25
26 test_datagen = ImageDataGenerator(rescale=1./255)
27
28 train_data = train_datagen.flow_from_directory(
29     train_dir,
30     target_size=IMG_SIZE,
31     batch_size=BATCH_SIZE,
32     class_mode='categorical',
33     shuffle=True
34 )
35
36 val_data = train_datagen.flow_from_directory(
37     val_dir,
38     target_size=IMG_SIZE,
39     batch_size=BATCH_SIZE,
40     class_mode='categorical',
41     shuffle=False
42 )
43
44 test_data = test_datagen.flow_from_directory(
45     test_dir,
46     target_size=IMG_SIZE,
47     batch_size=BATCH_SIZE,
48     class_mode='categorical',
49     shuffle=False
50 )
```

II.3.1.1 Explication du code

- **IMG_SIZE** : définit la taille des images à 224×224 pixels, une dimension couramment utilisée avec les réseaux de neurones convolutifs (CNN).
- **BATCH_SIZE** : indique le nombre d'images traitées simultanément à chaque itération pendant l'entraînement.
- **ImageDataGenerator** : génère dynamiquement des variantes d'images pendant l'entraînement afin d'augmenter artificiellement le jeu de données :
 - **rescale=1./255** : normalise les pixels en les ramenant à une échelle de 0 à 1.
 - **rotation_range, width_shift_range, height_shift_range, zoom_range, shear_range** : appliquent des transformations géométriques aléatoires (rotation, translation, zoom, cisaillement).
 - **brightness_range** : ajuste la luminosité de manière aléatoire.
 - **horizontal_flip, vertical_flip** : retournent aléatoirement les images horizontalement et verticalement.
 - **fill_mode='nearest'** : comble les zones vides après transformation en dupliquant les pixels voisins.
- **flow_from_directory** : charge les images depuis les répertoires et applique les transformations définies.
- **class_mode='categorical'** : spécifie que les étiquettes sont codées sous forme de vecteurs one-hot, adaptés aux problèmes de classification multi-classes.

Ce processus permet d'augmenter la diversité des données d'entraînement, ce qui améliore la capacité du modèle à généraliser et réduit le risque de surapprentissage, en particulier dans le cas d'un jeu de données limité.

II.3.1.2 Répartition des données

Afin d'assurer un apprentissage efficace du modèle et une évaluation rigoureuse de ses performances, le jeu de données a été divisé en trois sous-ensembles distincts :

- **Ensemble d'entraînement (70%)** des données ont été allouées à l'ensemble d'entraînement (**training set**). Cet ensemble est utilisé pour ajuster les paramètres internes du modèle à partir des exemples qu'il voit.
- **Ensemble de validation (20%)** des données ont été réservées pour l'ensemble de validation (**validation set**), qui permet d'évaluer la performance du modèle pendant l'apprentissage et de détecter les phénomènes de surapprentissage (overfitting).
- Enfin, **Ensemble de test (10%)** des données ont été conservées pour l'ensemble de test (**test set**), utilisé uniquement à la fin de l'entraînement pour mesurer la capacité de généralisation du modèle sur des données qu'il n'a jamais vues.

Le tableau II.3 fournit des nombres spécifiques d'images d'entraînement, de validation et de test, ainsi que le nombre de classes pour chaque ensemble de données.

ENSEMBLE DE DONNÉES	Nombre de classes	Augmentation des données	Nombre d'images		
			Training	Validation	Test
Blood Cells Image dataset	8	Non	11,959	3,415	1,718
		Oui	95,672	27,320	1,718
Lymphoblastic Leukemia (ALL)	4	Non	2,277	649	330
		Oui	18,216	5,192	330
Blood Cells Cancer (ALL) dataset	4	Non	2,268	647	327
		Oui	18,144	5,176	327

TABLE II.3 – Un nombre total d'images d'entraînement, de validation et de test sont utilisées pour les ensembles de données Blood Cells Image dataset, Lymphoblastic Leukemia (ALL) et Blood Cells Cancer (ALL) dataset, avec et sans augmentation de données..

II.4 Conclusion

Le chapitre présent a été consacré à la collecte et à la gestion des données, qui constituent la première étape du développement d'une application mobile de classification du cancer du sang au moyen d'algorithmes d'apprentissage profond sur des images médicales. Nous avons rapporté des données provenant de différentes sources, à savoir les jeux de données Leucémie lymphoblastique (ALL), Blood Cells Image Dataset et Blood Cells Cancer (ALL). Chacun des ensembles de données a fait l'objet d'une description basée sur ses caractéristiques, sa structure, et son utilité dans le cadre du projet.

Le traitement des données s'est révélé comme l'une des étapes majeures de ce travail, car garantir la qualité et la fiabilité des résultats est une nécessité. Nous avons fait appel à des techniques de prétraitement et de traitement telles que l'augmentation de données.

III

Développement des Modèles de Classification du cancer du sang

III.1 Introduction

Dans ce chapitre, nous présenterons en détail le processus de développement des modèles de classification du cancer du sang à partir d'images médicales, en exploitant les techniques avancées de l'apprentissage profond. Face à la complexité des images microscopiques du sang et à la nécessité d'un diagnostic rapide et fiable, les modèles de Deep Learning, en particulier les réseaux de neurones convolutifs (CNN), se sont révélés extrêmement performants pour extraire automatiquement les caractéristiques discriminantes des images.

Nous commencerons par décrire l'environnement de développement utilisé, incluant les outils et bibliothèques indispensables tels que Python, TensorFlow et Keras. Ensuite, nous détaillerons la conception de notre système de classification, basé à la fois sur des modèles CNN construits sur mesure et sur des architectures pré-entraînées telles que MobileNetV2 et ResNet101, via la technique du transfert d'apprentissage.

Enfin, une évaluation rigoureuse des performances sera réalisée en s'appuyant sur des métriques standards telles que l'accuracy, la précision, le rappel et le F1-score, permettant ainsi de comparer les résultats obtenus et de sélectionner le modèle le plus performant pour cette tâche critique de détection et de classification du cancer du sang.

III.2 Environnement de développement

Cette partie offre un aperçu succinct des langages de programmation et des logiciels employés dans notre travail, notamment Python, TensorFlow, et le paramètre de matériels utilisés :

Matériels/Logiciels	Paramètres
Processeur	Intel 13 ^{ème} génération -13900KF 3.00 GHz.
Capacité Mémoire (RAM)	128 GB
Unité de traitement graphique (GPU)	NVIDIA GeForce RTX 3060
Version CUDA	12.8
Windows	10 Professionnel
Python	3.10
TensorFlow	2.10.1

TABLE III.1 – Spécifications du système

III.2.1 Python

Python est un langage de programmation de haut niveau, polyvalent et largement utilisé. La dernière version de Python, Python 3, est utilisée dans le développement web, les applications d'apprentissage automatique et toutes les technologies logicielles de pointe. Python est employé par pratiquement tous les géants de l'internet, notamment Google, Amazon, Facebook, Instagram, Dropbox, Uber, etc. Aujourd'hui, Python est l'un des langages de programmation de haut niveau les plus utilisés, permettant la programmation dans les paradigmes orienté objet et procédural. Les applications Python sont généralement plus compactes que celles écrites dans d'autres langages de programmation tels que Java. Les programmeurs doivent taper très peu de code, et l'exigence d'indentation du langage garantit que leur code reste toujours compréhensible. [58]

Plusieurs Framework open sources sont disponibles dans la littérature, la grande majorité supporte le langage Python. Voici un exemple :

III.2.2 TensorFlow

TensorFlow est une interface de programmation scalable et multiplateforme pour la mise en œuvre et l'exécution d'algorithmes d'apprentissage automatique.

TensorFlow a été développé par les chercheurs et les ingénieurs de l'équipe Google Brain.

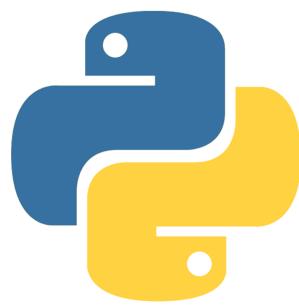


FIGURE III.1 – Python

Initialement conçu pour un usage interne à Google, TensorFlow a été publié en novembre 2015 sous une licence open source.

Afin d'améliorer les performances des modèles d'apprentissage automatique, TensorFlow permet une exécution à la fois sur les CPU et les GPU. Toutefois, ses plus grandes capacités de performance se révèlent lors de l'utilisation des GPU. TensorFlow prend actuellement en charge des interfaces frontend pour plusieurs langages de programmation. L'API (Application Programming Interface) Python de TensorFlow est la plus complète, attirant ainsi de nombreux praticiens de l'apprentissage automatique et de l'apprentissage profond. De plus, TensorFlow possède une API officielle en C++.

TensorFlow dispose d'un vaste écosystème de composants connexes, notamment des bibliothèques comme TensorBoard, ainsi que des API de déploiement et de production.

TensorFlow 1.x utilisait un système complexe de classes Python pour la construction de modèles. En raison de l'énorme popularité de Keras, lorsque TensorFlow 2.0 est sorti, TF a adopté Keras comme API officielle pour TensorFlow.



FIGURE III.2 – TensorFlow

III.2.2.1 Utilité de TensorFlow

TensorFlow est une bibliothèque open-source développée par Google Brain qui permet de concevoir, d'entraîner et de déployer des modèles d'apprentissage automatique et d'apprentissage profond. Dans le cadre de ce projet, TensorFlow est utilisé pour sa capacité à gérer efficacement de grandes quantités de données visuelles et à effectuer des calculs intensifs sur GPU ou TPU, accélérant ainsi le processus d'entraînement des modèles.

TensorFlow est particulièrement adapté au déploiement mobile grâce à des outils comme TensorFlow Lite, qui permet de convertir et d'optimiser les modèles pour une exécution efficace sur les appareils Android. De plus, son intégration avec Keras facilite la construction de réseaux neuronaux via une API simple et modulaire, parfaitement adaptée aux besoins du projet.

L'approche par graphe computationnel, combinée à l'exécution rapide (Eager Execution), rend TensorFlow à la fois performant pour les déploiements en production et flexible pour le prototypage. Il offre également des outils avancés de visualisation (TensorBoard) et de monitoring, très utiles lors de la phase de développement et de validation des modèles de classification d'images médicales.

III.2.3 Keras

Keras est une API d'apprentissage profond de haut niveau qui simplifie grandement la formation et l'exploitation de réseaux de neurones. Elle peut fonctionner sur TensorFlow, Theano ou Microsoft Cognitive Toolkit (anciennement connu sous le nom de CNTK). TensorFlow est livré avec sa propre implémentation de cette API, appelée tf.keras, qui prend en charge certaines fonctionnalités avancées de TensorFlow (par exemple, pour charger efficacement des données).

Bien que Keras reste une bibliothèque distincte de TensorFlow, elle peut désormais être officiellement importée par TensorFlow, ce qui rend son installation inutile.

L'API de Keras est facile à utiliser et permet de construire des modèles en ajoutant simplement des couches les unes sur les autres par de simples appels.



FIGURE III.3 – keras

III.2.3.1 Utilité de Keras

L'un des principaux avantages de Keras est sa simplicité d'utilisation. Elle offre une API intuitive, cohérente et modulaire, ce qui permet aux développeurs, chercheurs et étudiants de construire rapidement des réseaux neuronaux sans se soucier des détails complexes de l'implémentation bas niveau. Cette approche orientée utilisateur facilite l'expérimentation rapide avec les architectures de réseaux neuronaux.

III.3 Notre système de classification du cancer du sang à l'aide de algorithmes d'apprentissage profond

Le système de classification du cancer du sang (Figure III.4) que nous avons développé repose sur l'analyse d'images microscopiques de cellules sanguines à l'aide de l'apprentissage profond, en particulier d'un réseau de neurones convolutif (CNN). Ce système vise à détecter et classer automatiquement les cellules bénignes et malignes, notamment les sous-types de la leucémie lymphoblastique aiguë (ALL), à partir de frottis sanguins numérisés.

La première étape du processus consiste au prétraitement des données. Cette phase est essentielle pour garantir la qualité et la cohérence des données utilisées pour l'entraînement du modèle. Elle comprend plusieurs opérations, telles que le redimensionnement des images à une taille standard et la normalisation des valeurs de pixels pour assurer une homogénéité dans les intensités. Dans certains cas, des techniques d'augmentation des données peuvent également être appliquées afin de simuler une plus grande diversité d'échantillons, ce qui permet d'améliorer la robustesse du modèle.

Une fois les données préparées, elles sont divisées en trois sous-ensembles : entraînement, validation et test. Le modèle CNN est ensuite entraîné sur le jeu d'entraînement durant un total de 1000 époques. Durant cet apprentissage, le réseau apprend à extraire automatiquement des caractéristiques discriminantes des images pour différencier les types cellulaires. Après chaque époque, les performances du modèle sont évaluées sur le jeu de validation à l'aide de deux indicateurs principaux : la fonction de perte (loss) et la précision (accuracy).

Le système est conçu pour sauvegarder automatiquement le modèle dès qu'une nouvelle meilleure précision est atteinte sur le jeu de validation. Cette stratégie permet de conserver la version du modèle qui généralise le mieux aux données non vues, évitant ainsi le surapprentissage.

Enfin, le modèle entraîné est évalué sur le jeu de test, qui contient des images jamais vues auparavant par le réseau. Cette évaluation finale est réalisée à l'aide de plusieurs métriques de performance : la perte (loss), la précision (accuracy), la précision positive (precision), le rappel (recall) et le score F1. Ces indicateurs offrent une évaluation complète de la capacité du modèle à détecter correctement les différentes classes cellulaires, en particulier dans le

contexte critique du diagnostic du cancer du sang.

Ce système, basé sur une architecture CNN et un flux de traitement rigoureux, représente une solution prometteuse pour automatiser l'analyse cytologique et améliorer la détection précoce des leucémies.

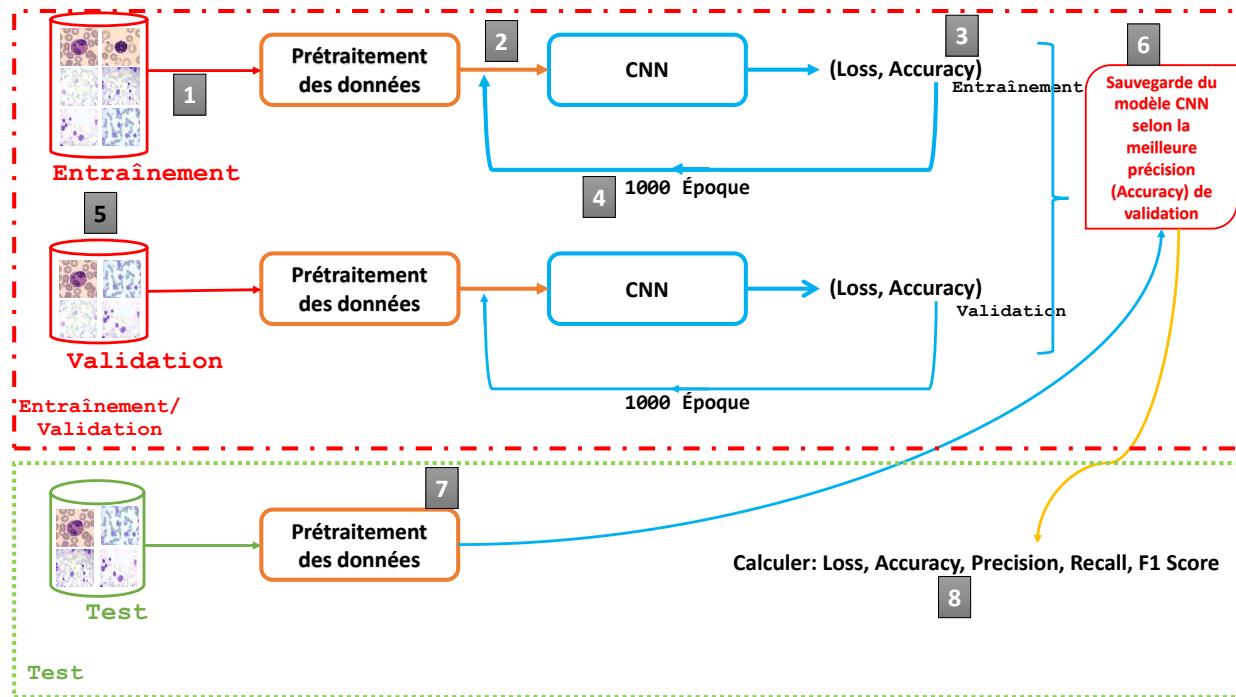


FIGURE III.4 – Notre système de classification du cancer du sang

III.4 Conception des modèles

La conception des modèles est une étape fondamentale dans le développement d'applications d'intelligence artificielle, notamment pour la classification des images médicales. Elle consiste à définir l'architecture du réseau de neurones, à choisir les algorithmes et à ajuster les paramètres afin d'optimiser les performances du modèle. Dans le cadre de la classification du cancer du sang, la conception d'un modèle performant nécessite l'utilisation de techniques avancées, telles que les réseaux de neurones convolutifs (CNN), ainsi que l'exploration d'approches comme l'apprentissage par transfert avec des modèles pré-entraînés. Cette section détaille les différentes étapes et stratégies utilisées pour concevoir un modèle capable de classer efficacement les images médicales et de fournir un diagnostic précis.

III.4.1 Extraction de caractéristiques par CNN

L'extraction de caractéristiques à l'aide des réseaux de neurones convolutifs (CNN) est une méthode puissante pour analyser les images médicales. En particulier, dans la classification du cancer du sang, les CNN permettent d'apprendre automatiquement les motifs pertinents à partir des images brutes, facilitant ainsi le diagnostic. Ce modèle utilise TensorFlow/Keras pour extraire des caractéristiques à partir d'images de 224x224 pixels, en vue de les classer en différentes catégories.

```
1  import tensorflow as tf
2  model = tf.keras.models.Sequential([
3      # Note the input shape is the desired size of the image 200x 200 with 3 bytes
4      # color
5      # The frst convolution
6      tf.keras.layers.Conv2D(8, (3,3), activation='relu', input_shape=(224,224, 3)),
7      tf.keras.layers.BatchNormalization(),
8      tf.keras.layers.MaxPooling2D(2, 2),
9      tf.keras.layers.Conv2D(16, (3,3), activation='relu'),
10     tf.keras.layers.BatchNormalization(),
11     tf.keras.layers.MaxPooling2D(2,2),
12     tf.keras.layers.Conv2D(16, (3,3), activation='relu'),
13     tf.keras.layers.BatchNormalization(),
14     tf.keras.layers.MaxPooling2D(2,2),
15     tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
16     tf.keras.layers.BatchNormalization(),
17     tf.keras.layers.MaxPooling2D(2,2),
18     tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
19     tf.keras.layers.BatchNormalization(),
20     tf.keras.layers.MaxPooling2D(2,2),
21
22     tf.keras.layers.Flatten(),
```

```
22     tf.keras.layers.Dense(128, activation='relu'),  
23  
24     tf.keras.layers.Dense(train_data.num_classes, activation='softmax')  
25 ])  
26 model.summary()
```

- **Création du modèle** : Le modèle est construit en utilisant l'API **Sequential** de Keras, qui permet de définir un réseau de neurones où les couches sont ajoutées de manière séquentielle, une après l'autre. Cela simplifie la construction d'architectures simples de réseaux de neurones.
- **Les couches de convolution** : Le modèle commence par une série de couches **convolutives** (Conv2D). Chaque couche utilise des filtres de taille 3x3 pour extraire les caractéristiques de l'image. Ces filtres détectent des motifs simples, comme les bords, les textures, ou d'autres éléments spécifiques dans l'image. Au fur et à mesure que les couches deviennent plus profondes, le réseau apprend à détecter des motifs de plus en plus complexes.
 - La première couche de convolution utilise 8 filtres pour extraire les caractéristiques de l'image.
 - Les couches suivantes utilisent des nombres croissants de filtres (16, puis 32), ce qui permet au modèle d'extraire des informations de plus en plus complexes à mesure que les données passent à travers les couches.
- **Normalisation et Pooling** : Après chaque couche de convolution, le modèle applique une **normalisation** avec la couche **BatchNormalization**, ce qui permet de stabiliser et d'accélérer l'apprentissage en réduisant les fluctuations internes du réseau. Ensuite, la couche **MaxPooling2D** est utilisée pour réduire la taille des images extraites. Cette opération de **pooling** conserve les informations les plus importantes tout en réduisant la dimensionnalité, ce qui améliore les performances et réduit le coût computationnel.
- **Flattening** : Après les couches convolutives, les caractéristiques extraites sont transformées en un vecteur unidimensionnel à l'aide de la couche **Flatten**. Cela permet de préparer les données pour les couches entièrement connectées (dense), qui traiteront ces informations de manière plus abstraite pour effectuer la classification finale.
- **Les couches entièrement connectées (Dense)** : Le modèle inclut une **couche dense** avec 128 neurones, activée par **ReLU**, qui agit comme un classificateur pour apprendre à partir des caractéristiques extraites. Cette couche est responsable de la combinaison des informations extraites par les couches précédentes et de leur transformation en une représentation plus abstraite qui permet une classification plus précise. La couche finale est une autre couche dense, avec un nombre de neurones égal au

nombre de classes à prédire, et utilise la fonction d'activation **Softmax**. Cette fonction calcule les probabilités que l'image appartienne à chaque classe. Le modèle classe l'image en attribuant à chaque classe une probabilité, et la classe avec la probabilité la plus élevée est choisie comme la sortie finale.

- **Résumé du modèle** : La méthode `summary()` permet d'afficher un résumé détaillé de l'architecture du modèle, y compris le nombre de paramètres dans chaque couche, la forme des données à chaque étape, et le nombre total de paramètres qui seront entraînés.

III.4.2 Utilisation de modèles pré-entraînés (MobileNetV2 et ResNet101) pour le transfert d'apprentissage

Les modèles pré-entraînés, comme MobileNetV2 et ResNet101, sont des réseaux neuro-naux profonds qui ont été formés sur de vastes ensembles de données, tels que ImageNet. Cela leur permet d'apprendre des caractéristiques de haut niveau des images, comme les formes, les textures et les objets. Ces modèles peuvent être utilisés dans des applications de transfert d'apprentissage, où un modèle préexistant est utilisé pour des tâches similaires sans avoir besoin de repartir de zéro.

III.4.2.1 *MobileNetV2*

MobileNetV2 est un modèle léger et efficace, conçu pour des applications mobiles et embarquées. Il est basé sur des architectures de réseaux convolutifs profonds (CNN) et utilise une méthode appelée Depthwise Separable Convolutions, qui permet de réduire le nombre de paramètres du modèle tout en conservant des performances élevées.

Caractéristiques de MobileNetV2 :

- **Optimisé pour les appareils mobiles** : Il est spécialement conçu pour être léger en termes de calcul et de mémoire, ce qui le rend idéal pour des appareils mobiles où les ressources sont limitées.
- **Depthwise Separable Convolutions** : Une technique qui consiste à séparer la convolution standard en deux étapes. D'abord, une convolution de profondeur (chaque filtre agit indépendamment sur chaque canal de l'image), puis une convolution pointwise (une convolution classique 1x1 pour combiner les résultats). Cette approche réduit le nombre de paramètres et le coût computationnel.
- **Verres résiduels (Residual Bottleneck)** : MobileNetV2 utilise des bottlenecks résiduels, un type de couche de réseau de neurones où la sortie de la couche est ajoutée à l'entrée. Cela aide à améliorer la performance du modèle en facilitant l'apprentissage

des caractéristiques plus complexes.

- **Application** : MobileNetV2 est couramment utilisé dans des systèmes où les ressources sont limitées, comme les applications sur smartphones, les dispositifs IoT, et autres appareils à faible consommation d'énergie.

Utilisation en transfert d'apprentissage :

Dans un scénario de transfert d'apprentissage, MobileNetV2 peut être utilisé comme un modèle pré-entraîné sur ImageNet pour des tâches de classification d'images. Les couches initiales, qui extraient les caractéristiques de base (comme les bords, les textures), peuvent être gelées (non modifiables) et seules les couches de sortie peuvent être ajustées pour effectuer une tâche spécifique, telle que la classification d'une image médicale.

III.4.2.2 ResNet101

ResNet101 fait partie de la famille ResNet (Residual Networks), un modèle très populaire dans le domaine de la vision par ordinateur. Il a été conçu pour résoudre les problèmes liés aux réseaux très profonds, tels que la disparition du gradient, et permet à un modèle d'avoir des centaines de couches tout en conservant une performance optimale.

Caractéristiques de ResNet101 :

- **Réseaux résiduels** : ResNet101 utilise des blocs résiduels qui ajoutent une connexion directe entre les couches. Ces connexions permettent à l'information de se propager plus facilement dans les réseaux très profonds, réduisant ainsi la perte d'information et le problème de disparition du gradient.
- **Profondeur** : Le ResNet101 possède 101 couches profondes, ce qui lui permet d'apprendre des caractéristiques très abstraites et complexes des images. La profondeur accrue permet au modèle d'apprendre des représentations plus riches, mais cela peut aussi rendre l'entraînement plus difficile, d'où l'importance des blocs résiduels.
- **Réduction de la dégradation** : ResNet résout le problème de la dégradation qui affecte les réseaux très profonds, où l'ajout de nouvelles couches peut dégrader la performance du modèle au lieu de l'améliorer.
- **Application** : ResNet101 est utilisé pour des tâches complexes de classification d'images, de détection d'objets, et même pour des applications plus avancées comme la segmentation sémantique d'images.

Utilisation en transfert d'apprentissage :

ResNet101 est particulièrement utile pour des applications nécessitant une compréhension plus détaillée des caractéristiques d'images complexes. En transfert d'apprentissage, il est couramment utilisé pour des tâches de classification d'images dans des domaines comme la médecine, où des caractéristiques subtiles doivent être extraites des images. Comme Mobi-

leNetV2, ResNet101 peut être pré-entraîné sur ImageNet, et ses couches profondes peuvent être utilisées pour extraire des caractéristiques générales d'images, avant de fine-tuner le modèle sur un ensemble de données spécifique.

III.5 Paramètres du modèle

Les paramètres du modèle sont essentiels pour configurer l'architecture du réseau de neurones, son entraînement et son évaluation. Voici la description de chaque paramètre clé dans notre travail :

1. IMG_SIZE : tuple (224, 224)

Description : Taille des images d'entrée, redimensionnées à 224 x 224 pixels.

Usage : Normalisation des images pour équilibrer les ressources de calcul et les détails visuels.

2. BATCH_SIZE : int (16)

Description : Nombre d'exemples traités simultanément.

Usage : Influence la mémoire et la stabilité de l'entraînement. Plus petit pour mieux contrôler la mémoire, plus grand pour accélérer la convergence.

3. epochs : int (1000)

Description : Nombre d'itérations sur l'ensemble des données.

Usage : Détermine combien de fois le modèle apprend à partir des données. Un nombre trop élevé risque le surapprentissage.

4. loss : string ('categorical_crossentropy')

Description : Fonction de perte pour évaluer la différence entre les prédictions et les vraies étiquettes.

Usage : Utilisée pour les problèmes de classification multi-classes.

5. optimizer : Adam(lr=0.0005)

Description : Optimiseur adaptatif pour ajuster les poids du modèle.

Usage : Le taux d'apprentissage (0.0005) influence la vitesse d'ajustement des poids.

6. metrics : list

Description : Métriques utilisées pour évaluer la performance du modèle pendant l'entraînement, la validation et test.

Accuracy : Précision globale.

Precision : Mesure de la précision.

Recall : Sensibilité du modèle.

F1 Score : Moyenne harmonique entre précision et rappel, utilisé en cas de classes déséquilibrées.

7. base_model (MobileNetV2 / ResNet101)

Modèle MobileNetV2 :

base_model : Le modèle MobileNetV2 pré-entraîné qui extrait des caractéristiques de l'image.

Dense(2048, activation='relu') : Une couche dense cachée avec 2048 neurones activée par ReLU pour combiner les caractéristiques extraites et permettre une représentation plus complexe.

Dropout(0.5) : Utilisé pour éviter le surapprentissage en désactivant aléatoirement 50 % des neurones pendant l'entraînement.

Dense(train_data.num_classes, activation='softmax') : La couche finale avec le nombre de neurones correspondant au nombre de classes de sortie, et une activation Softmax pour donner une probabilité pour chaque classe.

Modèle ResNet101 :

base_model : Le modèle ResNet101 pré-entraîné, qui est plus profond et utilise des blocs résiduels pour mieux apprendre des caractéristiques complexes.

Dense(2048, activation='relu') : Une couche cachée de 2048 neurones avec ReLU pour l'abstraction des caractéristiques.

Dropout(0.5) : Empêche le surapprentissage en désactivant aléatoirement 50 % des neurones pendant l'entraînement.

Dense(train_data.num_classes, activation='softmax') : La couche finale pour effectuer la classification, avec un neurone pour chaque classe et une activation Softmax pour produire des probabilités.

III.6 Évaluation des performances

L'évaluation des performances permet de mesurer l'efficacité des modèles de classification. Pour cela, plusieurs métriques sont utilisées, telles que la perte (loss), l'accuracy, la précision, le rappel, le F1-score, ainsi que la matrice de confusion. Ces indicateurs permettent de comparer les modèles et de sélectionner celui offrant les meilleurs résultats.

III.6.1 Fonction de perte (Loss Function)

La "loss" (ou fonction de perte) est une mesure de l'erreur entre les prédictions du modèle et les valeurs réelles (les étiquettes). La fonction de perte quantifie à quel point le modèle se trompe dans ses prédictions. L'objectif de l'entraînement d'un modèle est de minimiser cette fonction de perte.

III.6.2 Taux de reconnaissance (Accuracy)

L'accuracy est couramment utilisée pour évaluer les performances d'un modèle, car elle mesure le nombre de prédictions correctes par rapport au nombre total de prédictions. Cependant, dans certains cas, en particulier lorsque les données sont déséquilibrées, la précision peut être trompeuse. Par exemple, si un jeu de données contient une classe minoritaire, l'accuracy peut être élevée, même si le modèle ne prédit correctement que quelques exemples de cette classe [59].

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Total des échantillons}} \quad (\text{III.1})$$

III.6.3 Précision (Precision)

La précision est définie comme la proportion d'exemples vrais positifs parmi l'ensemble total des instances positives récupérées (true positive (TP) + false positive (FP)). Elle varie entre 0 et 1, et peut être calculée selon l'équation suivante :

$$\text{Précision} = \frac{TP}{TP + FP} = \frac{TP}{\text{Total des prédictions positives}} \quad (\text{III.2})$$

III.6.4 Rappel (Recall)

Également appelé sensibilité ou taux de vrais positifs (total true positive, TPR), le rappel est défini comme la proportion d'exemples TP parmi le total des vrais positifs (TP + faux négatifs (FN)). Sa valeur est toujours comprise entre 0 et 1. On peut le calculer à l'aide de l'équation suivante [59] :

$$\text{Rappel} = \frac{TP}{TP + FN} = \frac{TP}{\text{Taux de Vrais Positifs Total Actuel}} \quad (\text{III.3})$$

III.6.5 F1-score

Le score F1 est une mesure combinant le rappel et la précision de manière équilibrée, ce qui en fait une métrique populaire dans de nombreux algorithmes d'apprentissage automatique. Son calcul est défini dans l'équation suivante [59] :

$$\text{Score-F1} = \frac{2 \times (\text{Précision} \times \text{Rappel})}{\text{Précision} + \text{Rappel}} \quad (\text{III.4})$$

III.6.6 Matrice de Confusion

La matrice de confusion est un tableau permettant d'évaluer les performances d'un modèle de classification en comparant les prédictions du modèle avec les valeurs réelles (les

étiquettes). Elle contient les éléments suivants :

- **Vrai positif (TP - True Positive)** : Le modèle prédit correctement la classe positive.
- **Faux positif (FP - False Positive)** : Le modèle prédit la classe positive, alors que l'exemple appartient à la classe négative.
- **Vrai négatif (TN - True Negative)** : Le modèle prédit correctement la classe négative.
- **Faux négatif (FN - False Negative)** : Le modèle prédit la classe négative, alors que l'exemple appartient à la classe positive.

La matrice de confusion est généralement présentée sous cette forme :

	Prédit positif (P)	Prédit négatif (N)
Réel positif (P)	<i>TP</i>	<i>FN</i>
Réel négatif (N)	<i>FP</i>	<i>TN</i>

III.7 Évaluation des performances des modèles

Les performances des différents modèles de classification ont été évaluées sur deux bases de données représentatives des cancers du sang : Blood Cells Image Tableau III.2, et Lymphoblastic Leukemia ALL Tableau III.3. Cinq architectures ont été comparées : un CNN personnalisé, MobileNetV2 (entraînement complet), MobileNetV2 avec transfert d'apprentissage, ResNet101 (entraînement complet) et ResNet101 avec transfert d'apprentissage. Les résultats ont été étudiés selon des critères standards : précision, rappel, F1-score, temps d'entraînement, et poids mémoire.

III.7.1 Résultats sur la base de données 'Blood Cells Image'

Le tableau III.2 présente les performances des modèles sur la base de données 'Blood Cells Image'. Voici les principales observations :

- **MobileNetV2** : Ce modèle offre une bonne performance en termes de précision, rappel et F1-score, avec des valeurs proches de 0.99 pour l'ensemble d'entraînement et de validation. La perte est relativement faible, ce qui montre que le modèle apprend efficacement. De plus, le temps d'entraînement est raisonnable (29.5 heures), ce qui rend ce modèle adapté aux applications mobiles.
- **TL MobileNetV2** : L'usage du transfert d'apprentissage améliore légèrement les performances du modèle en termes de précision et rappel, mais augmente la perte en validation. Cela pourrait être dû à une mauvaise adaptation au jeu de données spécifique. Le modèle a cependant un temps d'entraînement réduit par rapport au modèle de base, ce qui est un avantage pour des applications pratiques.

- **ResNet101** : Ce modèle montre de très bonnes performances, avec des résultats excellents pour le rappel et la précision (près de 1). Cependant, le temps d'entraînement est plus long (45.85 heures), ce qui peut être un inconvénient si le temps de calcul est limité. Ce modèle est plus adapté pour des tâches complexes, mais nécessite plus de ressources.
- **TL ResNet101** : Comme pour TL MobileNetV2, l'utilisation du transfert d'apprentissage améliore les performances dans certains cas, mais avec un compromis sur la perte. Cela montre que la qualité du modèle pré-entraîné peut influencer l'efficacité du transfert.
- **CNN personnalisée** : Bien que ce modèle ait un nombre de paramètres relativement faible et un temps d'entraînement plus court, ses performances en termes de précision et rappel sont légèrement inférieures à celles de ResNet101 et MobileNetV2, ce qui pourrait suggérer qu'il est moins performant sur des tâches complexes comme la classification du cancer du sang.

Modèle profond	Métriques	Entraînement	Validation	Test	Par. d'ap.	Esp. sto.(.h5) Mo	Esp. sto.(.tflite) Mo
MobileNetV2	Loss	0.0213	0.0778	0.116	2,235,208	27.20	2.45
	Accuracy	0.9936	0.9836	0.9802			
	Precision	0.9937	0.9839	0.9802			
	Recall	0.9936	0.9836	0.9802			
	F1 Score	0.9934	0.9830	0.9792			
	Time	29.5 h	14.8 s	1.5 s			
TL MobileNetV2	Loss	0.017	0.2026	0.1846	2,639,880	1.96	2.65
	Accuracy	0.9943	0.9537	0.9587			
	Precision	0.9946	0.9537	0.9603			
	Recall	0.9939	0.9534	0.9587			
	F1 Score	0.9937	0.9498	0.9553			
	Time	18.35 h	15.3 s	1.4 s			
ResNet101	Loss	0.0103	0.0883	0.0721	42,516,552	500.36	42.30
	Accuracy	0.9978	0.9874	0.9884			
	Precision	0.9979	0.9877	0.9884			
	Recall	0.9978	0.9874	0.9884			
	F1 Score	0.9971	0.9867	0.9875			
	Time	45.85 h	29.5 s	5.1 s			
TL ResNet101	Loss	0.0639	0.2777	0.2037	4,212,744	49.39	46.82
	Accuracy	0.9764	0.9435	0.9563			
	Precision	0.9783	0.9467	0.9575			
	Recall	0.9750	0.9417	0.9563			
	F1 Score	0.9766	0.9382	0.9568			
	Time	18.93 h	15 s	5.2 s			
CNN personnalisée	Loss	0.0081	0.0607	0.0891	121,368	1.53	0.131
	Accuracy	0.9968	0.9830	0.9796			
	Precision	0.9968	0.9833	0.9796			
	Recall	0.9968	0.9830	0.9790			
	F1 Score	0.9972	0.9827	0.9798			
	Time	18.88 h	14.8 s	1.5 s			

TABLE III.2 – Mesures de performance des différents modèles de CNN sur la base de données 'Blood Cells Image' pour la classification du cancer du sang. **Par. d'ap.** : Paramètres d'apprentissage, **Esp. Sto.** :Espace de stockage.

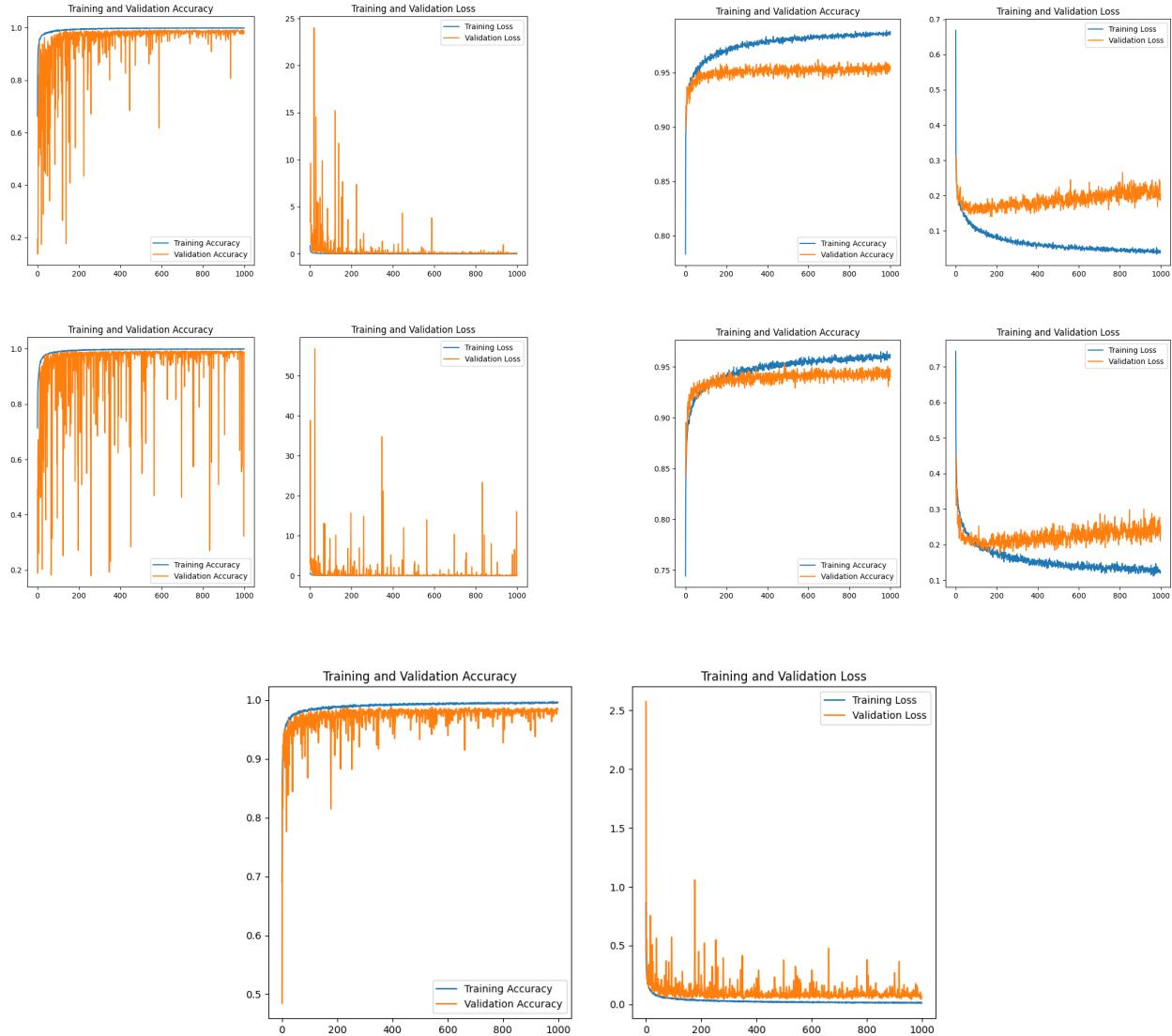


FIGURE III.5 – (a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée

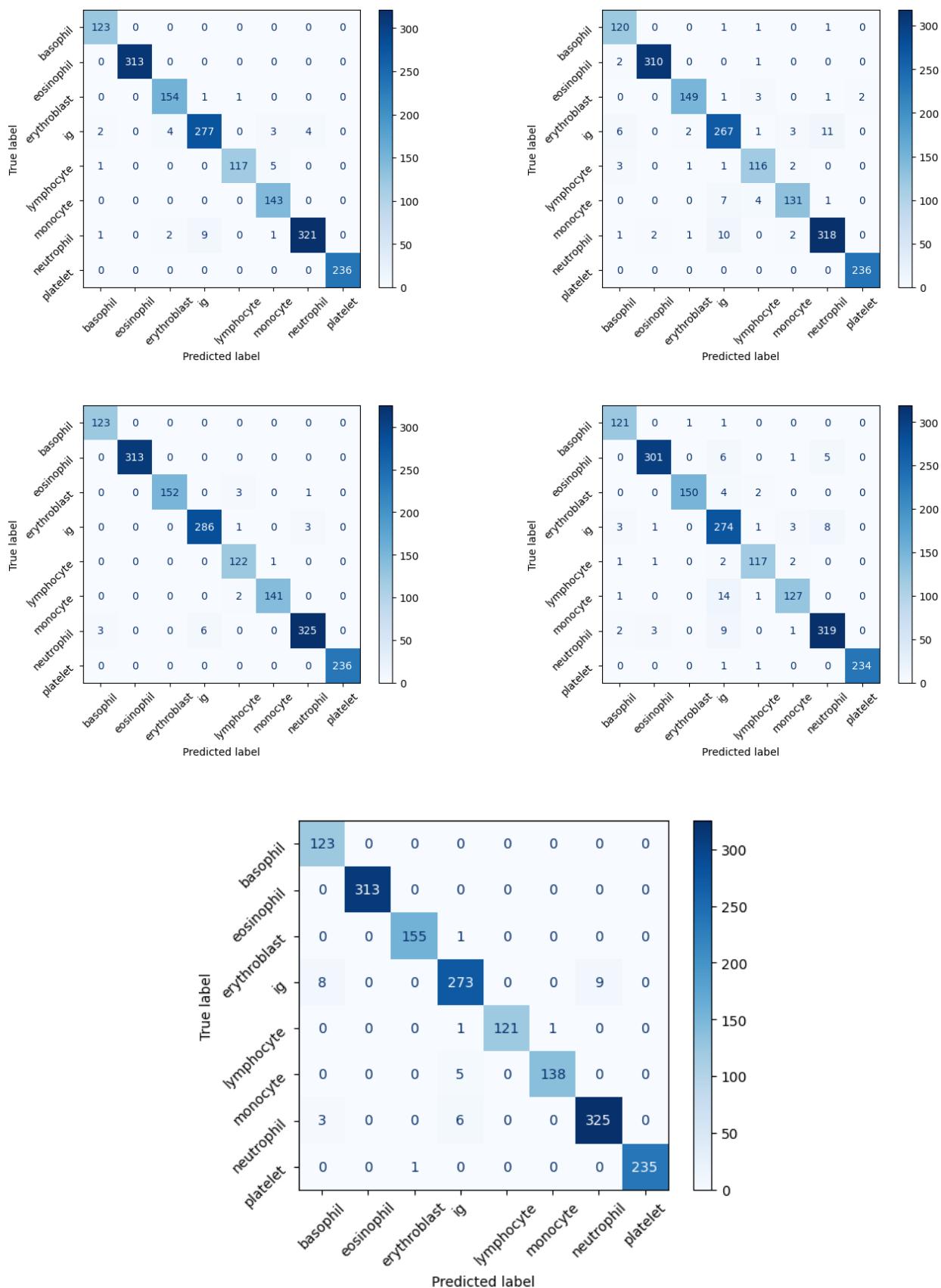


FIGURE III.6 – (a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée

III.7.2 Résultats sur la base de données 'Lymphoblastic Leukemia ALL'

Le tableau III.3 présente les performances des modèles sur la base de données 'Lymphoblastic Leukemia ALL'. Voici les principales observations :

- **MobileNetV2** : Le modèle atteint de bons résultats avec une précision et un rappel proches de 0.99. Cependant, il montre une légère augmentation de la perte sur l'ensemble de validation et de test par rapport aux autres modèles, ce qui pourrait indiquer que ce modèle est moins adapté aux données spécifiques de cette base de données.
- **TL MobileNetV2** : Ce modèle pré-entraîné sur ImageNet montre une amélioration des performances en termes de précision et de rappel par rapport au modèle de base, et la perte en validation est plus faible. Cela montre l'avantage de l'apprentissage par transfert, surtout lorsque les données spécifiques sont similaires aux données sur lesquelles le modèle a été pré-entraîné.
- **ResNet101** : Comme dans l'analyse sur 'Blood Cells Image', ce modèle montre d'excellentes performances, avec des valeurs de précision et rappel très élevées. La perte est faible et le temps d'entraînement relativement court par rapport à la base de données précédente, ce qui en fait un modèle adapté pour des applications complexes dans des environnements avec des ressources plus puissantes.
- **TL ResNet101** : L'utilisation du transfert d'apprentissage a amélioré la précision et la rappel, avec une perte qui reste raisonnable, mais moins performante que celle de ResNet101 sans transfert d'apprentissage. Il semble que le modèle pré-entraîné ait une meilleure adaptation aux données de la base 'Lymphoblastic Leukemia ALL'.
- **CNN personnalisée** : Ce modèle montre une performance inférieure à celle des autres modèles en termes de précision et de rappel, ce qui peut être dû à un nombre de paramètres insuffisant pour capturer les caractéristiques complexes des images de cette base de données. Il a cependant un temps d'entraînement relativement court, ce qui peut être un avantage dans des cas où les ressources sont limitées.

Modèle profond	Métriques	Entraînement	Validation	Test	Par. d'ap.	Esp. sto.(.h5) Mo	Esp. sto.(.tflite) Mo
MobileNetV2	Loss	0.0082	0.0185	0.0527	2,631,684	30.68	5.07
	Accuracy	0.9974	0.9908	0.9848			
	Precision	0.9974	0.9908	0.9848			
	Recall	0.9974	0.9908	0.9848			
	F1 Score	0.9969	0.9888	0.9825			
	Time	29.5 h	14.8 s	1.5 s			
TL MobileNetV2	Loss	0.0028	0.0062	0.0486	2,230,084	27.14	2.45
	Accuracy	0.9978	0.9954	0.9879			
	Precision	0.9978	0.9954	0.9879			
	Recall	0.9978	0.9954	0.9879			
	F1 Score	0.9973	0.9944	0.9856			
	Time	3.45 h	3 s	0.4 s			
resnet101	Loss	0.0049	0.044	0.0654	42,508,356	200.27	42.29
	Accuracy	0.9982	0.9908	0.9939			
	Precision	0.9982	0.9908	0.9939			
	Recall	0.9982	0.9908	0.9939			
	F1 Score	0.998	0.9888	0.9927			
	Time	8.46 h	2.7 s	1.0 s			
TL ResNet101	Loss :	0.0368	0.1239	0.0318	4,204,548	49.3	46.81
	Accuracy	0.9934	0.9753	0.9879			
	Precision	0.9934	0.9753	0.9909			
	Recall	0.9934	0.9753	0.9879			
	F1 Score	0.9931	0.9725	0.9892			
	Time	4.43 H	2.8 s	1.1 s			
CNN personnalisée	Loss :	0.0814	0.0822	0.148	120,852	1.53	0.132
	Accuracy	0.9776	0.9753	0.9667			
	Precision	0.9785	0.9769	0.9667			
	Recall	0.9772	0.9753	0.9667			
	F1 Score	0.9752	0.9738	0.9614			
	Time	3.46 H	2.8 S	0.4 S			

TABLE III.3 – Mesures de performance des différents modèles de CNN sur la base de données 'Lymphoblastic Leukemia ALL' pour la classification du cancer du sang. **Par. d'ap.** : Paramètres d'apprentissage, **Esp. Sto.** :Espace de stockage.

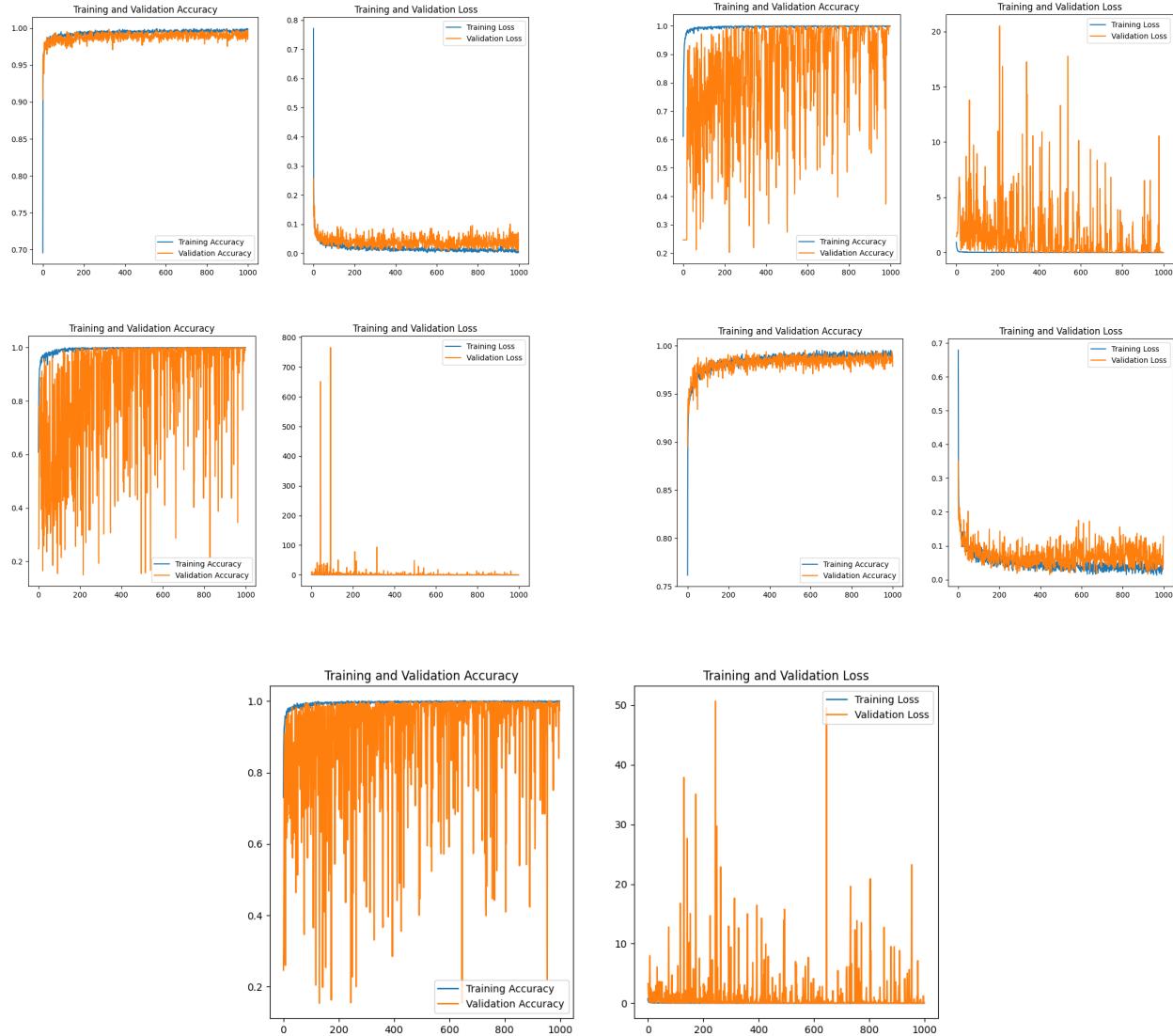


FIGURE III.7 – (a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée

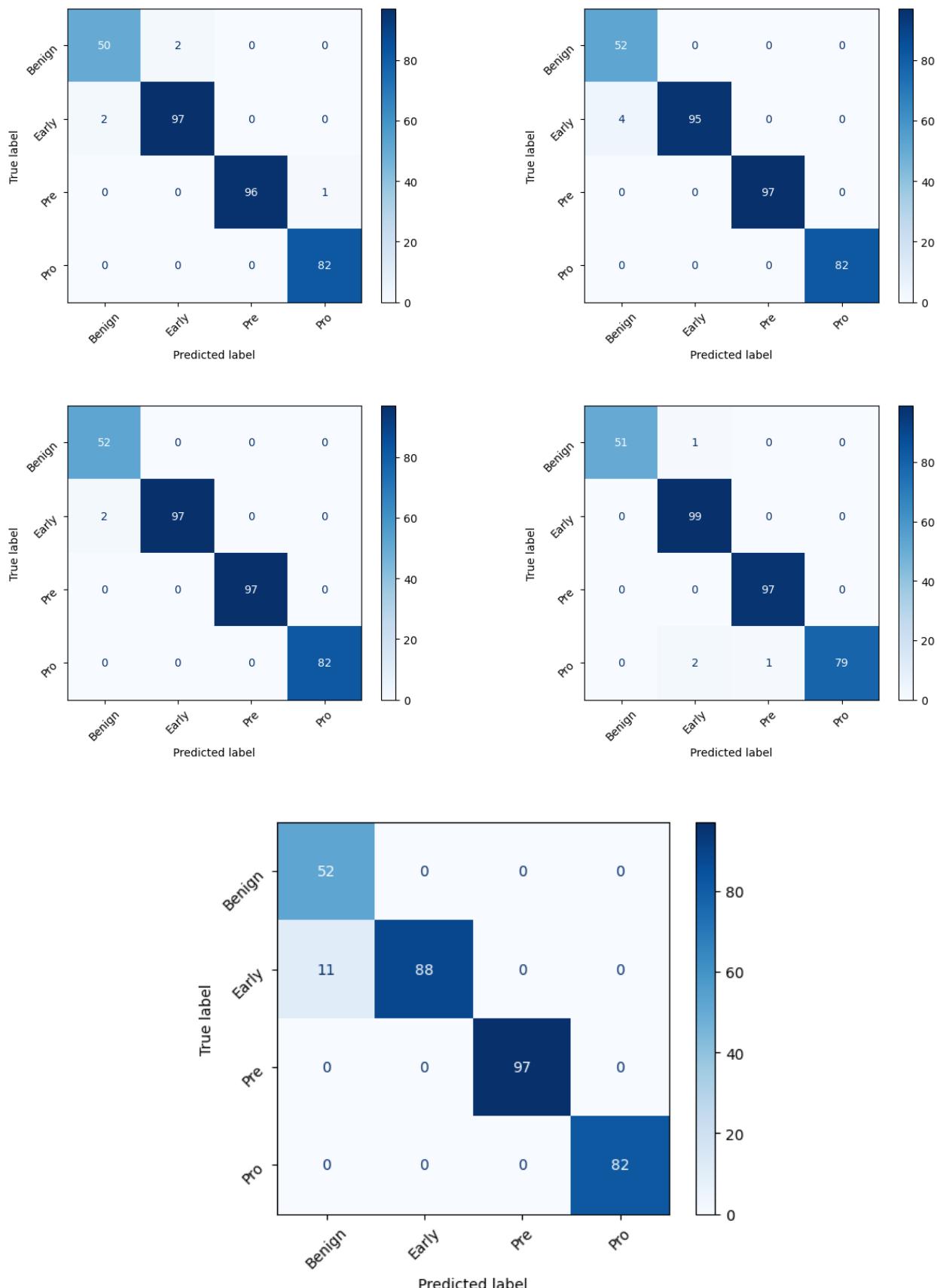


FIGURE III.8 – (a) MobileNetV2, (b) TL MobileNetV2, (c) resnet101, (d) TL ResNet101, (e) CNN personnalisée

Les résultats des expériences menées avec différents modèles de réseaux de neurones (CNN, MobileNetV2, ResNet101, etc.) sur les bases de données 'Blood Cells Image' et 'Lymphoblastic Leukemia ALL' ont été présentés dans les tableaux précédents. L'analyse des résultats permet de mieux comprendre les performances de chaque modèle et de choisir celui qui offre les meilleures performances pour la classification du cancer du sang.

III.7.3 Comparaison générale des résultats

En comparant les résultats obtenus sur les deux bases de données, on observe que les modèles **MobileNetV2** et **ResNet101** (en particulier avec l'apprentissage par transfert) offrent de très bonnes performances en termes de précision et de rappel. Cependant, ResNet101 semble être plus adapté aux tâches complexes et aux jeux de données plus variés, bien qu'il soit plus coûteux en termes de temps d'entraînement.

Le modèle **CNN personnalisé** présente des performances inférieures mais offre des avantages en termes de vitesse d'entraînement, ce qui peut être pertinent dans des scénarios où les ressources computationnelles sont limitées.

L'utilisation de modèles pré-entraînés avec l'apprentissage par transfert (TL MobileNetV2 et TL ResNet101) a montré des résultats améliorés, ce qui confirme que l'apprentissage par transfert est une stratégie efficace, notamment pour des applications dans des domaines spécialisés comme la classification d'images médicales.

III.8 Conclusion

Les résultats obtenus montrent que les modèles d'apprentissage profond sont bien adaptés à la classification du cancer du sang à partir d'images. Les architectures pré-entraînées permettent un entraînement plus rapide, mais leur succès dépend fortement de la compatibilité entre les données source et cible. Le modèle ResNet101 reste le plus performant, mais le CNN personnalisé, grâce à son efficacité et sa légèreté, se révèle particulièrement prometteur pour une implémentation mobile. L'application développée pourra donc s'appuyer sur un modèle équilibré comme MobileNetV2 ou un CNN compact pour garantir une performance optimale tout en respectant les contraintes matérielles.

IV

Développement de l'Application Mobile et Tests

IV.1 Introduction

Dans ce chapitre, nous présentons les différentes étapes de développement de l'application mobile BloodDetect AI, destinée à assister les professionnels de santé dans le diagnostic rapide du cancer du sang à partir d'images sanguines. Nous décrivons tout d'abord la conception de l'interface utilisateur, en mettant l'accent sur l'ergonomie, l'accessibilité et la simplicité de navigation à travers les différentes fonctionnalités proposées, telles que l'enregistrement, le diagnostic, la consultation, l'historique et la gestion du profil. Ensuite, nous détaillons le processus d'intégration du modèle de deep learning dans l'environnement Android, en exposant les méthodes de conversion et d'optimisation adoptées pour garantir une exécution efficace sur des dispositifs mobiles à ressources limitées. Enfin, nous présentons les tests utilisateurs réalisés en conditions réelles pour évaluer la fiabilité du système et sa pertinence clinique.

IV.2 Diagrammes de formation de modèles

Le **Langage de Modélisation Unifié (UML)** est un outil standardisé largement utilisé en génie logiciel pour représenter visuellement les différents aspects d'un système logiciel. Il fournit un ensemble de symboles graphiques permettant de décrire la **structure**, le **comportement** et les **interactions** du système. Les diagrammes UML jouent un rôle essentiel en tant que *langage commun* facilitant la communication entre développeurs, concepteurs, parties prenantes et autres membres du projet, tout au long du cycle de vie du développement logiciel.

Dans cette section, nous présentons notre projet à l'aide de différents types de diagrammes UML, notamment :

- un **diagramme de cas d'utilisation** pour décrire les interactions entre l'utilisateur et le système ;
- un **diagramme de séquence** pour illustrer la dynamique des échanges ;
- un **diagramme de classes** pour modéliser la structure statique du système.

IV.2.1 Diagramme de cas d'utilisation

Le diagramme ci-dessous illustre les différents cas d'utilisation de l'application **BloodDetect AI**, développée pour l'analyse et la classification du cancer du sang à partir d'images médicales. L'acteur principal ici est le **médecin**, qui interagit avec l'application à travers plusieurs fonctionnalités essentielles.

- **S'inscrire et Se connecter** à la plateforme ;
- **Analyser une image sanguine**, incluant la capture ou le téléchargement de l'image ainsi que la saisie du nom et de l'âge du patient ;
- **Afficher et enregistrer le résultat du diagnostic** ;
- **Consulter l'historique et les consultations précédentes** ;
- **Modifier son profil, changer de mot de passe et se déconnecter**.

Les relations de type « *include* » expriment les dépendances fonctionnelles entre les cas d'utilisation. Par exemple, l'analyse sanguine inclut obligatoirement la fourniture d'une image ainsi que les informations du patient.

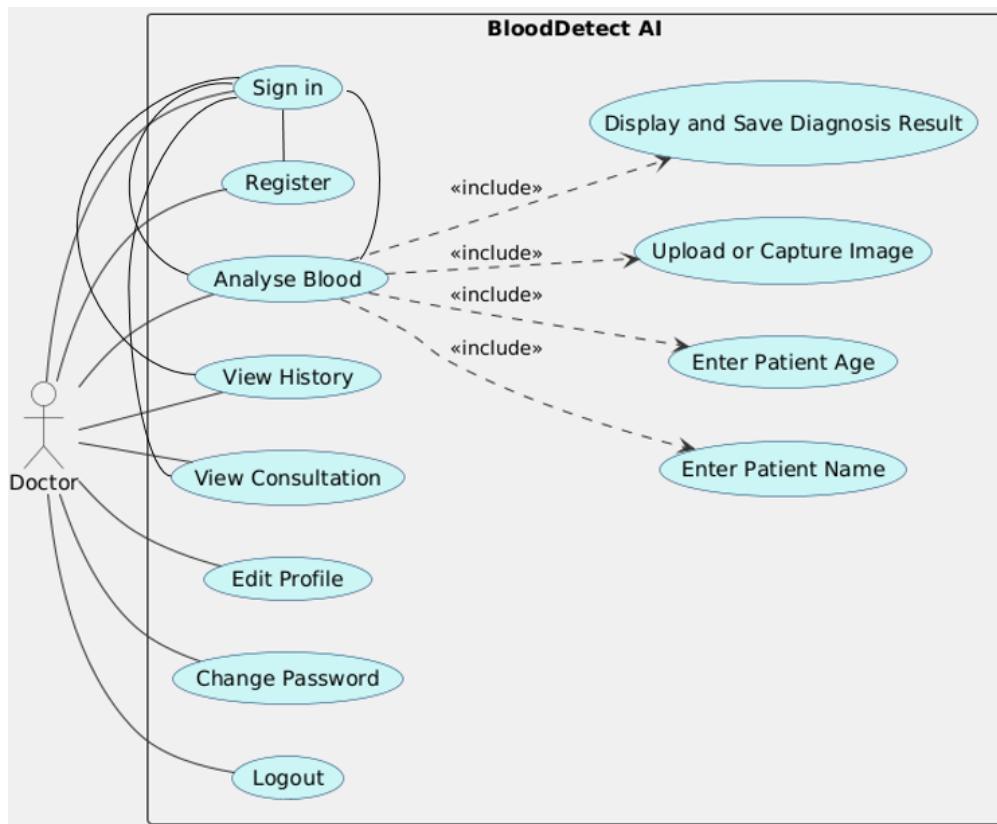


FIGURE IV.1 – Diagramme de cas d'utilisation de l'application BloodDetect AI

IV.2.2 Diagramme de Séquence

IV.2.2.1 Système d'Authentification et d'Enregistrement des Utilisateurs

Ce diagramme illustre le flux des processus de connexion (Login) et de création de compte (Inscription) dans le système, mettant en évidence les interactions entre l'utilisateur, le médecin, le système et la base de données.

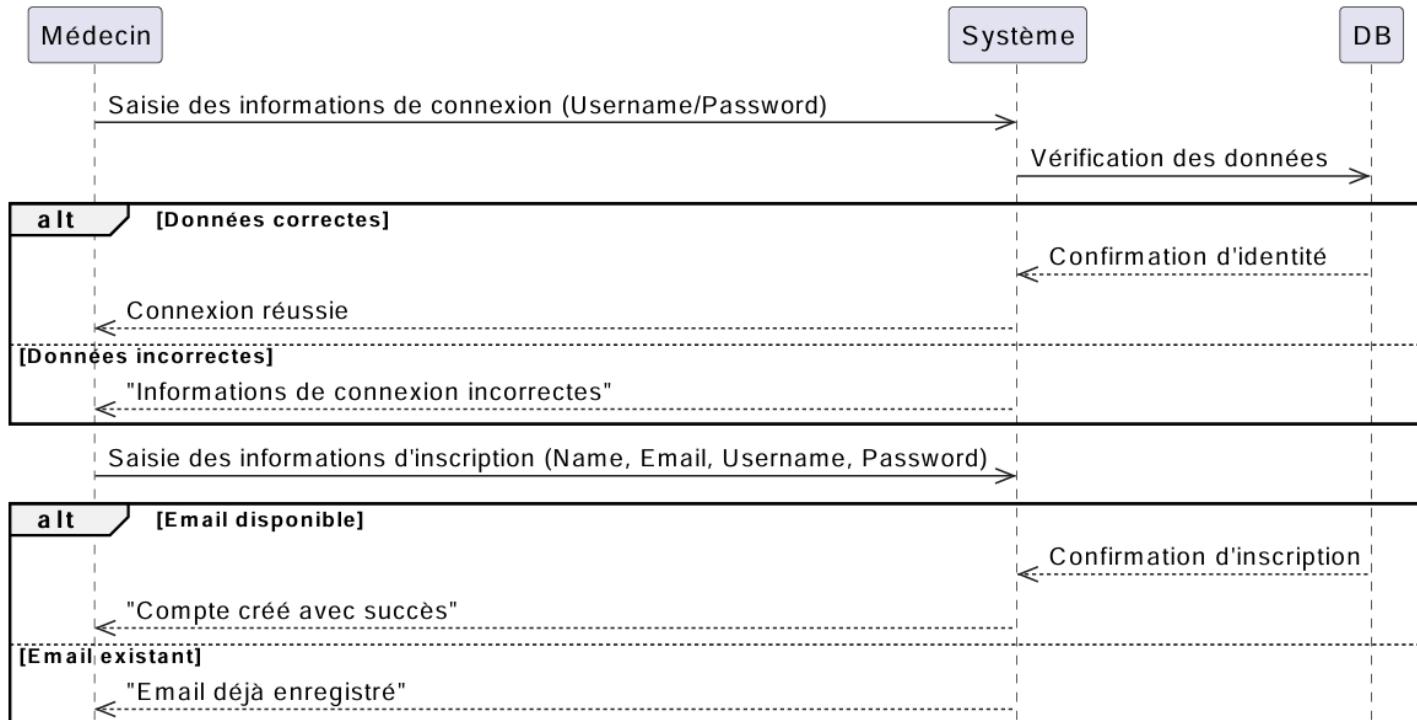


FIGURE IV.2 – Diagramme de séquence — Système d'authentification et d'enregistrement des utilisateurs

IV.2.2.2 Gestion du Profil Médecin

Ce diagramme montre comment un médecin modifie son profil dans l'application. Il présente les étapes clés : demande de modification, vérification par le système, mise à jour sécurisée des données (avec chiffrement du mot de passe), et retour d'information immédiat au médecin. Le processus garantit à la fois la sécurité et la simplicité d'utilisation.

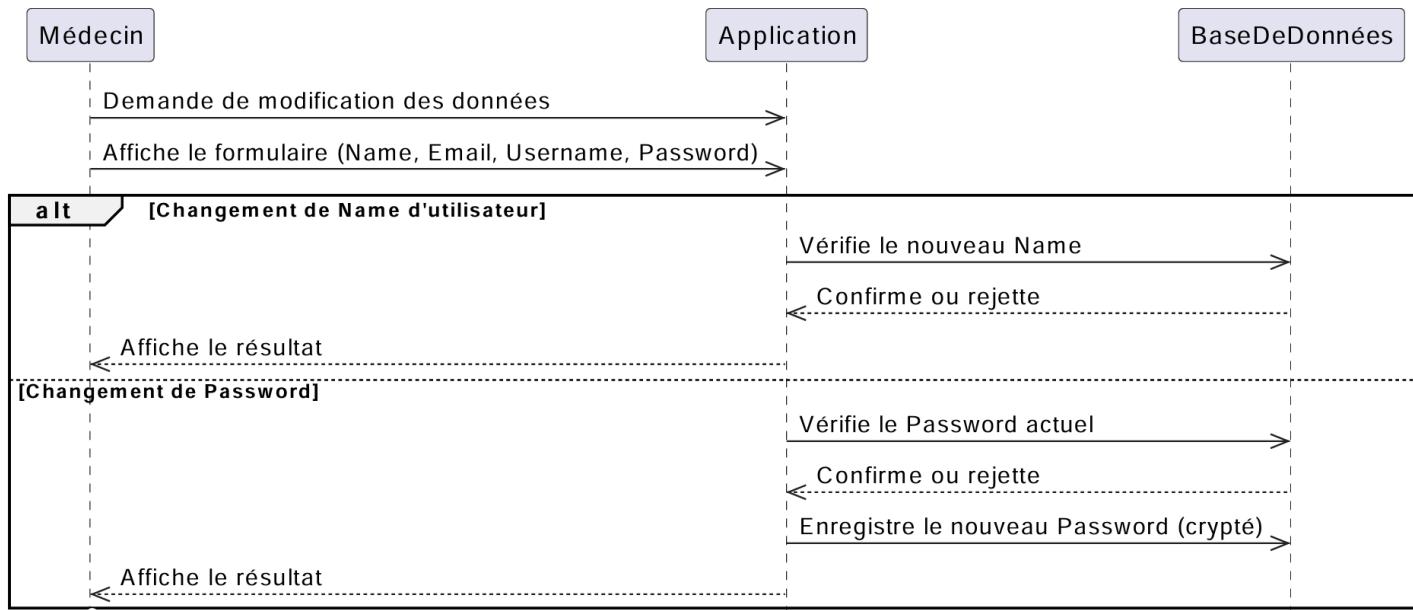


FIGURE IV.3 – Diagramme de séquence — Gestion du profil médecin

IV.2.2.3 Processus de Diagnostic et de Sauvegarde — BloodDetect AI

Ce diagramme montre le parcours complet d'une analyse sanguine dans BloodDetect AI : le médecin saisit les données du patient et envoie l'image, le système effectue une analyse à l'aide de l'intelligence artificielle pour produire un diagnostic détaillé, enregistre automatiquement les résultats dans le dossier médical et confirme l'opération. Un processus médical sécurisé et automatisé.

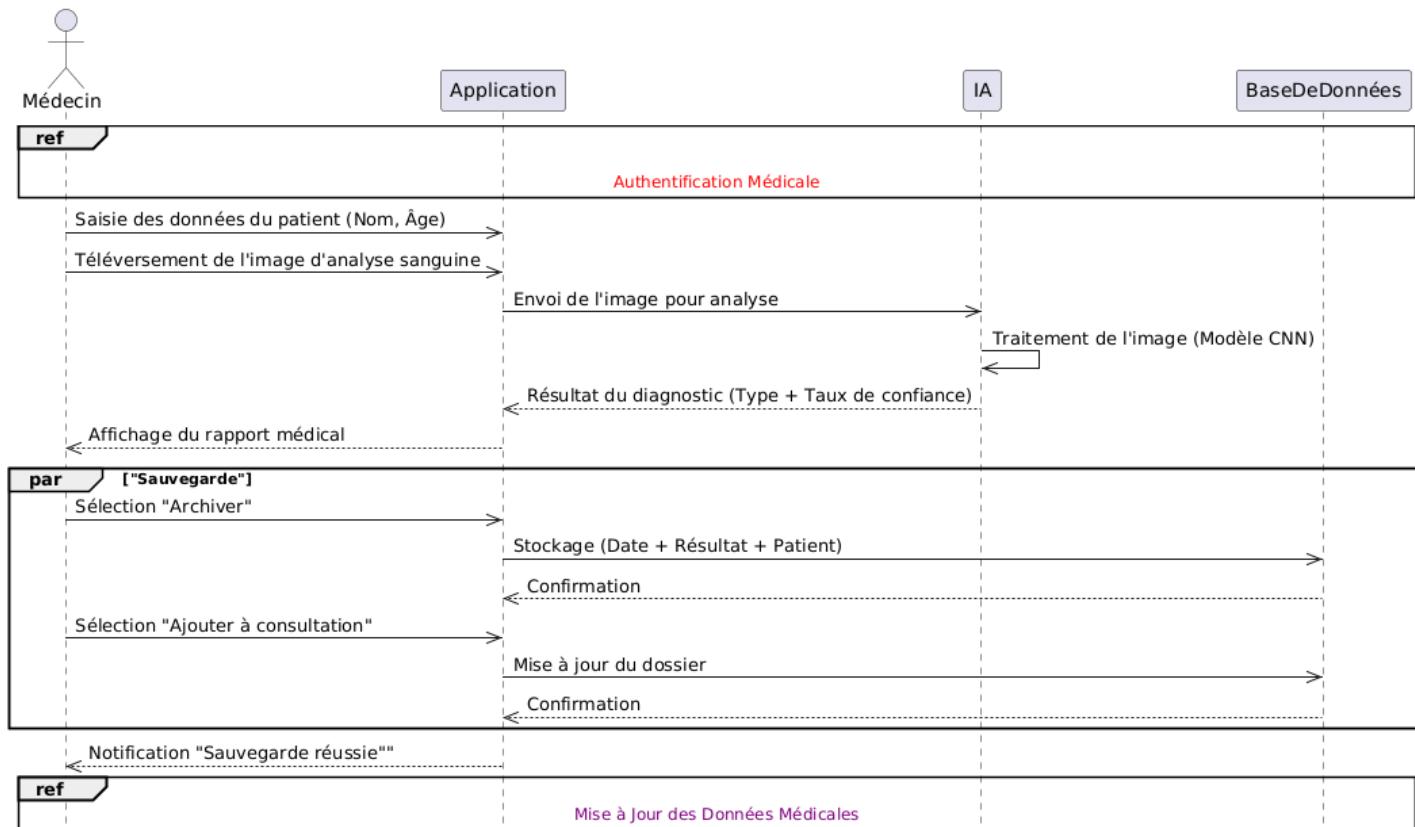


FIGURE IV.4 – Diagramme de séquence — Processus de diagnostic et de sauvegarde dans BloodDetect AI

IV.2.3 Diagramme de Classe

Un médecin effectue une analyse sanguine pour un patient. Cette analyse produit un diagnostic, qui est enregistré dans l'historique des diagnostics. Ensuite, une consultation est créée et résumée via une vue dédiée. Le système prend en charge l'authentification, la mise à jour du profil, ainsi que d'autres fonctionnalités liées à la gestion des utilisateurs.

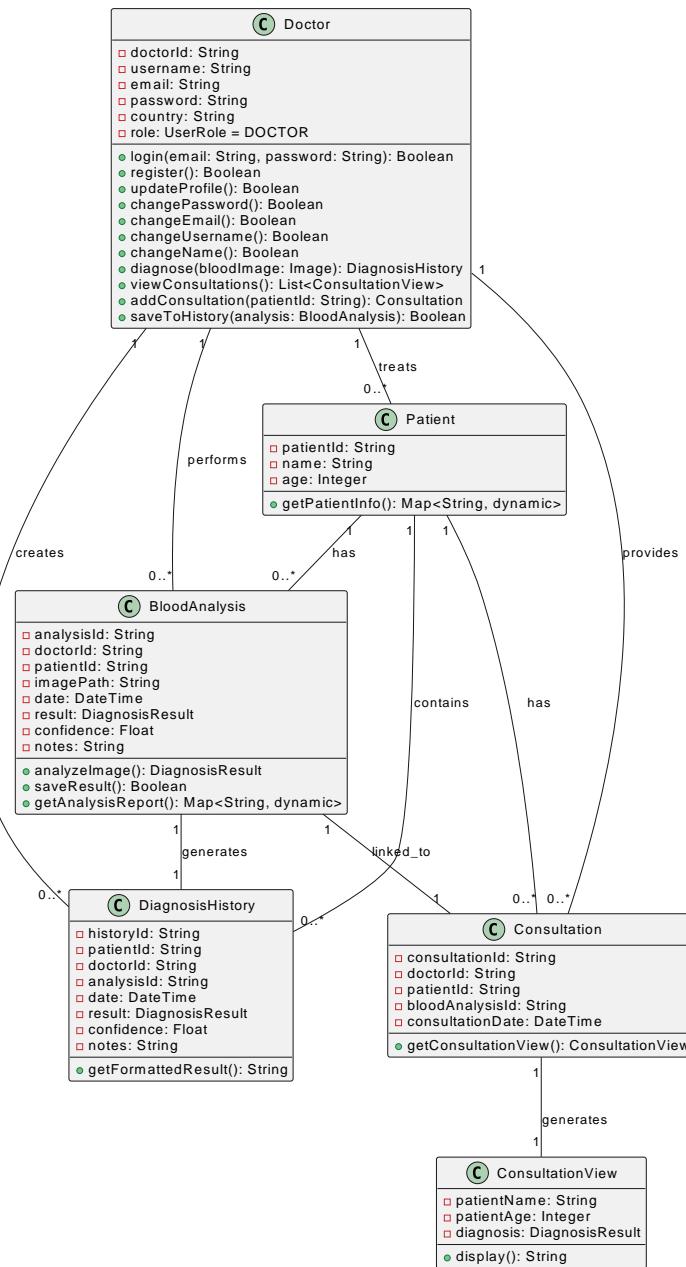


FIGURE IV.5 – Diagramme de classe du système

IV.3 Technologies pour le Développement Mobile

IV.3.1 Android Studio

Android Studio est un environnement de développement intégré (IDE) officiel de Google pour la création d'applications mobiles Android. Il propose un éditeur de mise en page visuel, permettant aux développeurs de concevoir des interfaces complexes et de les prévisualiser sur différentes tailles d'écran et appareils.

Ce logiciel offre plusieurs fonctionnalités essentielles :

- Un éditeur de code source et de ressources.
- Des outils de compilation basés sur **Gradle**, un système de construction performant permettant d'automatiser la génération de projets.
- Des outils de test, de débogage et d'optimisation.

Android Studio permet ainsi aux développeurs :

- de créer des interfaces utilisateurs avancées et adaptatives,
- de réduire la taille des applications générées,
- d'utiliser des émulateurs rapides pour tester les applications,
- de bénéficier de statistiques en temps réel,
- de configurer des constructions personnalisées, etc.

IV.3.2 Utilité d'Android Studio

Android Studio joue un rôle essentiel dans le cycle de vie du développement des applications Android. Il centralise tous les outils nécessaires au développement, aux tests, et à la distribution des applications mobiles.

Grâce à son éditeur de code intelligent, à sa gestion efficace des ressources, et à l'intégration d'outils de débogage et d'analyse, Android Studio facilite la conception d'applications performantes, adaptées à divers formats d'écrans.

Il permet également :

- de simuler différents environnements Android via des émulateurs,
- de surveiller les performances des applications en temps réel,
- d'optimiser le code et les ressources,
- d'améliorer la qualité des applications livrées sur le Google Play Store.



FIGURE IV.6 – Environnement Android Studio

IV.4 Conception de l'interface utilisateur

IV.4.1 BloodDetect AI – Connexion ou Inscription

Écran d'accueil de l'application *BloodDetect AI*, affichant le logo et le nom de l'application. La page principale propose deux boutons : **Login** pour se connecter et **Sign Up** pour créer un nouveau compte. Elle permet à l'utilisateur de commencer à utiliser l'application.

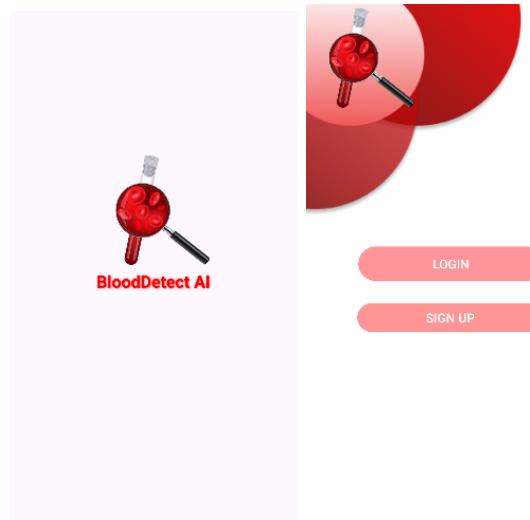


FIGURE IV.7 – BloodDetect AI – Connexion ou Inscription

IV.4.2 Login et Sign Up

La page d'inscription permet aux médecins de créer un compte en saisissant leur nom, e-mail, nom d'utilisateur et mot de passe. La page de connexion leur permet ensuite d'accéder à leur espace personnel en utilisant leur nom d'utilisateur et mot de passe.

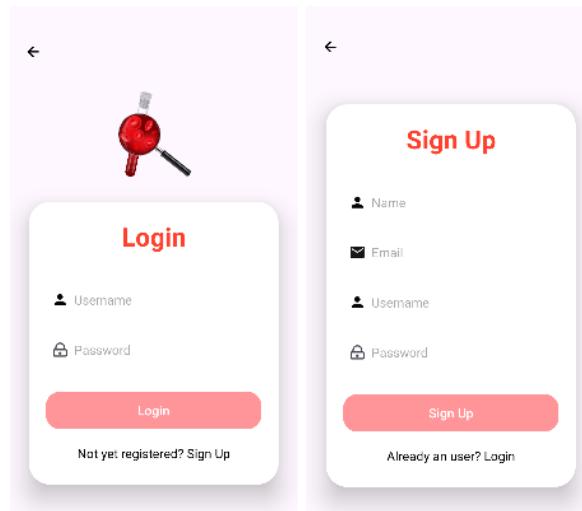


FIGURE IV.8 – Pages de connexion (Login) et d'inscription (Sign Up)

IV.4.3 Écran d'accueil de l'application Doctar

Cet écran s'affiche après la connexion du médecin. Il affiche le nom d'utilisateur et propose quatre boutons : ****Diagnosis**** (faire un diagnostic), ****Consultation**** (consulter un expert), ****History**** (voir l'historique médical), et ****Settings**** (paramètres). Un bouton ****Back**** permet de revenir à l'écran précédent.



FIGURE IV.9 – Écran d'accueil de l'application Doctar – Diagnostic rapide du cancer du sang

IV.4.4 Page de diagnostic sanguin

Cette page permet au médecin d'enregistrer un patient, de charger une image sanguine et d'obtenir un diagnostic rapide et précis grâce à l'intelligence artificielle. Après analyse, elle affiche les résultats pertinents et recommande une consultation médicale en cas d'anomalie.

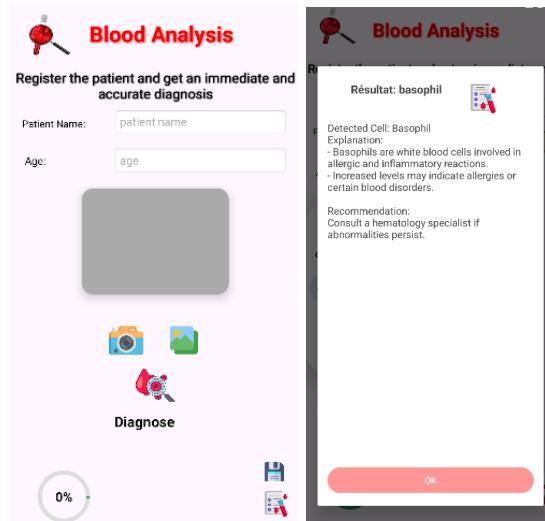


FIGURE IV.10 – Page de diagnostic sanguin : enregistrement du patient et affichage des résultats détaillés

IV.4.5 Historique des analyses sanguines (History)

Cette page affiche de manière détaillée l'historique d'une analyse sanguine spécifique. Elle présente clairement le résultat du diagnostic, le taux de fiabilité, ainsi que les informations du patient : nom, âge, date et heure de l'analyse.

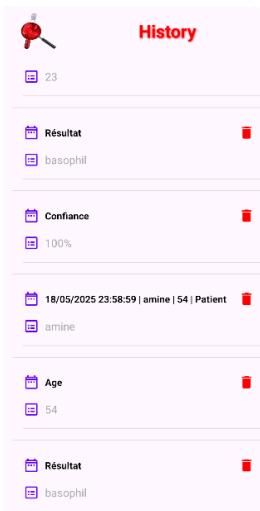


FIGURE IV.11 – Historique des analyses sanguines

IV.4.6 Gestion des consultations

La page de gestion des consultations présente la liste des consultations avec des informations résumées. Elle permet de naviguer vers les détails de chaque consultation. Ces derniers incluent définitions, complications possibles, traitements recommandés et conseils médicaux.

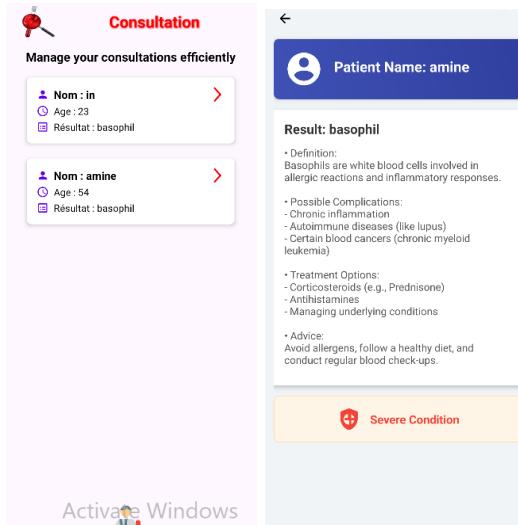


FIGURE IV.12 – Gestion des consultations

IV.4.7 Paramètres de compte (Settings) et modification du profil

La page **Settings** permet à l'utilisateur d'afficher ses informations de profil et d'accéder aux options de modification du profil ou de déconnexion. La page **Edit Profile** permet à l'utilisateur de mettre à jour ses informations personnelles (nom, e-mail, nom d'utilisateur et mot de passe), puis d'enregistrer les modifications.

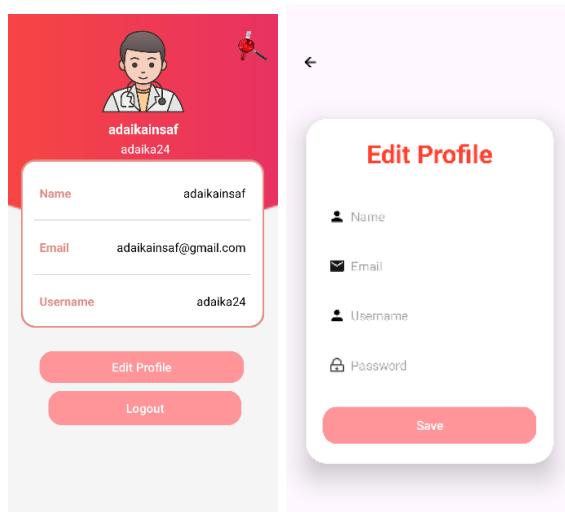


FIGURE IV.13 – Paramètres de compte et modification du profil

IV.5 Intégration du modèle

L'intégration du modèle de deep learning dans une application mobile constitue une étape cruciale pour exploiter les résultats de l'apprentissage automatique en conditions réelles. Après l'entraînement sur des données médicales, il est nécessaire d'adapter le modèle aux contraintes matérielles et logicielles des appareils Android. Cette intégration repose sur deux étapes essentielles : la conversion du modèle dans un format optimisé pour les dispositifs mobiles et l'optimisation des performances pour garantir une exécution fluide, rapide et économique en ressources. Cette section détaille les méthodes et outils employés.

IV.5.1 Conversion du modèle

Après l'entraînement des modèles de deep learning (CNN personnalisé, MobileNetV2, ResNet101) avec TensorFlow, ceux-ci doivent être convertis vers un format compatible avec Android. Pour cela, nous avons utilisé TensorFlow Lite (TFLite), permettant une exécution efficace sur mobiles et systèmes embarqués.

La conversion s'effectue avec **TensorFlow Lite Converter**, qui transforme un modèle (.h5 ou SavedModel) en fichier .tflite.

Deux méthodes sont possibles :

- Via la ligne de commande (CLI)
- En Python, via l'API TFLite

Exemple en Python :

```

1 import tensorflow as tf
2 model.save('my_model.h5')
3 print("Model saved as my_model.h5")

```

```

4 converter = tf.lite.TFLiteConverter.from_keras_model(model)
5 converter.optimizations = [tf.lite.Optimize.DEFAULT]
6
7 tflite_model = converter.convert()
8 with open('my_model.tflite', 'wb') as f:
9     f.write(tflite_model)
10 print("Model exported to TFLite successfully!")

```

Cette étape est essentielle pour intégrer le modèle dans l'application Android à l'aide de l'API TensorFlow Lite Interpreter.

IV.5.2 Optimisation des performances

Une fois le modèle converti en format TFLite, il doit être optimisé pour une exécution rapide et fluide sur des smartphones limités en ressources (RAM, processeur, GPU, etc.). Plusieurs techniques ont été utilisées :

Quantification : réduction de la précision des données Elle consiste à remplacer les poids et activations en float32 par des int8 ou float16, ce qui :

- Réduit la taille du modèle
- Accélère l'inférence
- Diminue la consommation mémoire et énergétique

Exemple de quantification dynamique :

```

1 converter.optimizations = [tf.lite.Optimize.DEFAULT]
2 tflite_quant_model = converter.convert()

```

IV.6 Tests de l'application

Une fois le développement et l'intégration du modèle terminés, des tests approfondis sont nécessaires pour évaluer la performance, la fiabilité et l'ergonomie de l'application dans un contexte réel. Ces tests simulent l'usage en milieu clinique et recueillent l'avis de professionnels de santé.

IV.6.1 Tests utilisateurs

Les tests utilisateurs ont été réalisés en collaboration avec des médecins biologistes et des techniciens de laboratoire. Cette étape permet d'évaluer :

- La précision des prédictions
- L'expérience utilisateur
- La facilité d'utilisation
- L'adéquation aux besoins du personnel médical

IV.7 Conclusion

Ce chapitre a mis en lumière le processus complet de transformation d'un modèle d'intelligence artificielle en une application mobile fonctionnelle, intuitive et adaptée au domaine médical. De la conception d'une interface utilisateur conviviale à l'intégration et l'optimisation du modèle de deep learning, en passant par la validation par des professionnels de santé, chaque étape a été conçue pour assurer la robustesse, la performance et la facilité d'utilisation de BloodDetect AI. Les tests réalisés confirment la pertinence de cette solution numérique dans le cadre d'un usage médical, posant ainsi les bases d'un outil intelligent et portable pour le diagnostic précoce du cancer du sang. Les résultats obtenus ouvrent la voie à de futures améliorations, notamment en termes de précision, de personnalisation et d'intégration avec d'autres systèmes de santé.

Conclusion Générale

Dans un paysage médical en constante évolution, où la demande en solutions de diagnostic rapides et fiables ne cesse de croître, l'intégration de l'intelligence artificielle (IA) dans les processus cliniques s'impose comme un levier stratégique majeur. Ce projet s'est inscrit pleinement dans cette dynamique, en développant une application mobile dédiée à la classification automatisée du cancer du sang à partir d'images microscopiques, combinant ainsi innovation technologique et utilité clinique.

En s'appuyant sur la puissance des algorithmes d'apprentissage profond, plusieurs architectures de réseaux de neurones convolutifs (CNN) ont été conçues et évaluées, notamment un modèle personnalisé, MobileNetV2 et ResNet101. Afin de garantir une exécution fluide et efficace sur les appareils Android, les modèles entraînés ont été convertis au format TensorFlow Lite, en recourant à des techniques avancées d'optimisation telles que la quantification et l'utilisation de délégués matériels.

L'évaluation approfondie des performances — en termes de précision des prédictions, de rapidité de traitement et d'ergonomie — ainsi que les retours positifs des professionnels de santé ayant testé l'application en contexte clinique, ont confirmé sa fiabilité et son potentiel en tant qu'outil d'aide au diagnostic. Ce projet met ainsi en évidence l'importance de l'interaction entre la technologie et l'expertise médicale, contribuant à améliorer la qualité des diagnostics précoces et à réduire les erreurs humaines.

Enfin, ce travail ouvre des perspectives de développement prometteuses, notamment l'intégration d'autres types de cancers hématologiques, l'enrichissement du jeu de données, ainsi que l'ajout de fonctionnalités avancées telles que la segmentation automatique des cellules et l'explicabilité des décisions du modèle via des méthodes d'interprétabilité comme Grad-CAM. Ces évolutions futures permettront de renforcer davantage l'impact clinique de l'application et d'accompagner les professionnels de santé dans leur pratique quotidienne.

Bibliographie

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets : Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv :1704.04861*, 2017.
- [3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetv2 : Inverted residuals and linear bottlenecks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [4] E. W. Jamoom, N. Yang, E. Hing, and S. Schappert, “State of adoption and use of electronic health records in the united states,” National Center for Health Statistics Data Brief No. 236, 2016, available at : <https://www.cdc.gov/nchs/products/databriefs/db236.htm>.
- [5] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, “Artificial intelligence in healthcare : past, present and future,” *Stroke and vascular neurology*, vol. 2, no. 4, pp. 230–243, 2017.
- [6] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, “A guide to deep learning in healthcare,” *Nature medicine*, vol. 25, no. 1, pp. 24–29, 2019.
- [7] E. J. Topol, “High-performance medicine : the convergence of human and artificial intelligence,” *Nature medicine*, vol. 25, no. 1, pp. 44–56, 2019.
- [8] R. B. Hegde, K. Prasad, H. Hebbar, B. M. K. Singh, and I. Sandhya, “Automated decision support system for detection of leukemia from peripheral blood smear images,” *Journal of Digital Imaging*, vol. 33, pp. 361–374, 2019.

- [9] S. M. Namayandeh, Z. Khazaei, M. L. Najafi, E. Goodarzi, and A. Moslem, “Global leukemia in children 0–14 statistics 2018, incidence and mortality and human development index (hdi) : Globocan sources and methods,” *Asian Pacific Journal of Cancer Prevention*, vol. 21, no. 5, pp. 1487–1494, 2020.
- [10] Unknown, “Childhood leukemias,” <https://www.nejm.org/doi/full/10.1056/NEJM199506153322407>, 2020.
- [11] F. Xing and L. Yang, “Robust nucleus/cell detection and segmentation in digital pathology and microscopy images : A comprehensive review,” *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 234–263, 2016.
- [12] J. Wen, Y. Xu, Z. Li, Z. Ma, and Y. Xu, “Inter-class sparsity based discriminative least square regression,” *Neural Networks*, vol. 102, pp. 36–47, 2018.
- [13] Z. Obermeyer and E. J. Emanuel, “Predicting the future—big data, machine learning, and clinical medicine,” *New England Journal of Medicine*, vol. 375, no. 13, pp. 1216–1219, 2016.
- [14] M. Ghaderzadeh, F. Rebecca, and A. Standring, “Comparing performance of different neural networks for early detection of cancer from benign hyperplasia of prostate,” *Applied Medical Informatics*, vol. 33, pp. 45–54, 2013.
- [15] J. Zhao, M. Zhang, Z. Zhou, J. Chu, and F. Cao, “Automatic detection and classification of leukocytes using convolutional neural networks,” *Medical Biological Engineering Computing*, vol. 55, no. 8, pp. 1287–1301, 2017.
- [16] T. W. Reader, R. Flin, K. Mearns, and B. H. Cuthbertson, “Developing a team performance framework for the intensive care unit,” *Critical Care Medicine*, vol. 37, no. 5, pp. 1787–1793, 2009.
- [17] N. Bibi, M. Sikandar, I. U. Din, A. Almogren, and S. Ali, “Iomt-based automated detection and classification of leukemia using deep learning,” *J. Healthcare Eng.*, vol. 2020, pp. 1–12, 2020.
- [18] R. G. Bagasjvara, I. Candraewi, S. Hartati, and A. Harjoko, “Automated detection and classification techniques of acute leukemia using image processing : a review,” in *2nd International Conference on Science and Technology Computer (ICST)*, Yogyakarta, Indonesia, 2016.
- [19] R. B. Hegde, K. Prasad, H. Hebbar, B. M. K. Sing, and I. Sandhya, “Automated decision support system for detection of leukemia from peripheral blood smear images,” *Digital Imaging*, vol. 33, pp. 361–374, 2019.

- [20] L. Lin, W. Wang, and B. Chen, “Leukocyte recognition with convolutional neural network,” *Algorithms Comput. Technol.*, vol. 13, pp. 1–8, 2018.
- [21] W. Yu, J. Chang, C. Yang, L. Zhang, H. Shen, Y. Xia, and J. Sha, “Automatic classification of leukocytes using deep neural network,” in *Proceedings of the 2017 IEEE 12th International Conference on ASIC (ASIC-CON)*, Guiyang, China, 2017.
- [22] M. A. Alsalem, A. A. Zaidan, B. B. Zaidan, M. Hashim, H. T. Madhloom, N. D. Azeez, and S. Alsyyisuf, “A review of the automated detection and classification of acute leukaemia : coherent taxonomy, datasets, validation and performance measurements, motivation, open challenges and recommendations,” *Comput. Methods Programs Biomed.*, vol. 158, pp. 93–112, 2018.
- [23] K. K. Anilkumar, V. J. Manoj, and T. M. Sagi, “Colour based image segmentation for automated detection of leukaemia : a comparison between cielab and cmyk colour spaces,” in *International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, Kottayam, India, 2018.
- [24] J. Laosai and K. Chamnongthai, “Classification of acute leukemia using medical knowledge-based morphology and cd marker,” *Biomed. Signal Process. Control*, vol. 44, pp. 127–137, 2018.
- [25] T. Pansombut, S. Wikaisuksakul, K. Khongkraphan, and A. Phon-on, “Convolutional neural networks for recognition of lymphoblast cell images,” *Comput. Intell. Neurosci.*, 2019.
- [26] S. N. M. Safuan, M. R. M. Tomari, W. N. W. Zakaria, B. M. K. Sing, M. N. H. Mohd, and N. S. Suriani, “Computer aided system for lymphoblast classification to detect acute lymphoblastic leukemia,” *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 14, pp. 597–607, 2019.
- [27] J. M. Bennett, D. Catovsky, M. T. Daniel, G. Flandrin, D. A. G. Galton, H. R. Gralnick, and C. Sultan, “Proposals for the classification of the acute leukaemias,” *Brit. J. Hematol.*, vol. 33, no. 4, pp. 451–458, 1976.
- [28] A. Rehman, N. Abbas, T. Saba, S. I. Ur-Rahman, Z. Mehmood, and H. Kolivand, “Classification of acute lymphoblastic leukemia using deep learning,” *Microsc. Res. Tech.*, vol. 81, no. 11, pp. 1310–1317, 2018.
- [29] S. Shafique and S. Tehsin, “Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks,” *Technol. Cancer Res. Treatment*, vol. 17, pp. 1–7, 2018.

- [30] T. Terwilliger and M. Abdul-Hay, “Acute lymphoblastic leukemia : a comprehensive review and 2017 update,” *Blood Cancer J.*, vol. 7, p. e577, 2017.
- [31] W. Wiharto, E. Suryani, and Y. R. Putra, “Classification of blast cell type on aml based on image morphology of white blood cells,” *Telecomm. Comput. Electron. Control (TELKOMNIKA)*, vol. 17, pp. 645–652, 2019.
- [32] B. J. Bain, *Blood Cells : A Practical Guide*, 6th ed. Wiley-Blackwell, 2021. [Online]. Available : <https://www.wiley.com/en-us/Blood+Cells%3A+A+Practical+Guide%2C+6th+Edition-p-9781119630573>
- [33] A. V. Hoffbrand and P. A. H. Moss, *Essential Haematology*, 7th ed. Wiley-Blackwell, 2019. [Online]. Available : <https://www.wiley.com/en-us/Essential+Haematology%2C+7th+Edition-p-9781119398060>
- [34] D. M. Harmening, *Clinical Hematology Atlas*, 6th ed. Elsevier, 2022. [Online]. Available : <https://www.elsevier.com/books/clinical-hematology-atlas/harmening/978-0-323-71399-5>
- [35] B. J. Bain, *Bone Marrow Pathology*, 5th ed. Wiley-Blackwell, 2015. [Online]. Available : <https://www.wiley.com/en-us/Bone+Marrow+Pathology%2C+5th+Edition-p-9781118802711>
- [36] World Health Organization, *WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues*, 4th ed. International Agency for Research on Cancer (IARC), 2017. [Online]. Available : <https://publications.iarc.fr/Book-And-Report-Series/Who-Classification-Of-Tumours/WHO-Classification-Of-Tumours-Of-Haematopoietic-And-Lymphoid-Tissues-2017>
- [37] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, “Applications of deep learning and reinforcement learning to biological data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2063–2079, Jun. 2018.
- [38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks ?” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
- [39] E. H. Houssein, M. M. Emam, A. A. Ali, and P. N. Suganthan, “Deep and machine learning techniques for medical imaging-based breast cancer : A comprehensive review,” *Expert Syst. Appl.*, vol. 167, Apr. 2021, art. no. 114161.
- [40] P. P. Banik, R. Saha, and K.-D. Kim, “An automatic nucleus segmentation and CNN model based classification method of white blood cell,” *Expert Syst. Appl.*, vol. 149, Jul. 2020, art. no. 113211.

- [41] K. K. Jha and H. S. Dutta, "Mutual information based hybrid model and deep learning for acute lymphocytic leukemia detection in single cell blood smear images," *Comput. Methods Programs Biomed.*, vol. 179, Oct. 2019, art. no. 104987.
- [42] A. I. Shahin, Y. Guo, K. M. Amin, and A. A. Sharawi, "White blood cells identification system based on convolutional deep neural learning networks," *Comput. Methods Programs Biomed.*, vol. 168, pp. 69–80, Jan. 2019.
- [43] M. Claro, L. Vogado, R. Veras, A. Santana, J. Tavares, J. Santos, and V. Machado, "Convolution neural network models for acute leukemia diagnosis," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 63–68.
- [44] L. H. Vogado, R. M. S. Veras, F. H. D. Araujo, R. R. V. Silva, and K. R. T. Aires, "Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 415–422, Jun. 2018.
- [45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe : Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
- [46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv :1409.1556*, 2014.
- [47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [49] F. Chollet, "Xception : Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
- [50] R. B. Hegde, K. Prasad, H. Hebbbar, and B. M. K. Singh, "Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images," *Biocybernetics Biomed. Eng.*, vol. 39, no. 2, pp. 382–392, Apr. 2019.

- [51] M. R. Reena and P. M. Ameer, “Localization and recognition of leukocytes in peripheral blood : A deep learning approach,” *Comput. Biol. Med.*, vol. 126, Nov. 2020, art. no. 104034.
- [52] S. S. que and S. Tehsin, “Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks,” *Technol. Cancer Res. Treatment*, vol. 17, Sep. 2018, art. no. 1533033818802789.
- [53] W. Yu, J. Chang, C. Yang, L. Zhang, H. Shen, Y. Xia, and J. Sha, “Automatic classification of leukocytes using deep neural network,” in *Proc. IEEE 12th Int. Conf. ASIC (ASICON)*, Oct. 2017, pp. 1041–1044.
- [54] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [55] P. K. Mallick, S. K. Mohapatra, G.-S. Chae, and M. N. Mohanty, “Convergent learning-based model for leukemia classification from gene expression,” *Pers. Ubiquitous Comput.*, vol. 2, pp. 1–8, Oct. 2020.
- [56] Disponible sur Kaggle. [Online]. Available : <https://doi.org/10.34740/kaggle/dsv/2175623>
- [57] Vol. 37, pp. 5113–5133. [Online]. Available : <https://doi.org/10.1002/int.22753>
- [58] GeeksforGeeks. Python programming language. GeeksforGeeks. Consulté le 01-11-2023. [Online]. Available : <https://www.geeksforgeeks.org/python-programming-language>
- [59] C. Ferri, J. Hernández-Orallo, and M. Modroiu, “An experimental comparison of performance measures for classification,” *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, 2009.