

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université -Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département Math-Informatique



Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : CYSIA
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Cyber sécurité et intelligence artificielle

Thème

Deep Learning pour l'analyse et la classification d'images

Présenté par :

- 1) Melle Guindo Sadio
- 2) Melle Traoré Alimata

Devant le jury composé de :

Dr Beddad Fatima	MCB	UAT.B.B (Ain Temouchent)	Président
Dr Benaribi Fethi Imad	MAA	UAT.B.B (Ain Temouchent)	Examineur
Dr Benomar Mohamed Lamine	MCA	UAT.B.B (Ain Temouchent)	Encadreur

Année Universitaire 2023/2024



Remerciement



(◇)

Avant toute chose, nous tenons à exprimer notre plus profonde gratitude envers Allah, le Tout-Puissant. C'est grâce à sa volonté et ses bénédictions que nous avons pu mener à bien la réalisation de ce mémoire.

Nos plus sincères remerciements s'adressent à notre cher encadreur, **Mr Benomar Mohamed Lamine** pour ses judicieux conseils, sa patience, ses encouragements, sa disponibilité, son expertise et son savoir-faire, qui nous ont été d'une aide inestimable pour l'aboutissement de ce projet. Que votre engagement remarquable soit récompensé à sa juste valeur.

Un immense merci à nos familles, pour leur amour inconditionnel, leurs bénédictions et leur soutien indéfectible à la fois moral et financier, qui ont été une source inépuisable de réconfort et de soutien malgré la distance.

À mes chers amis(e), vous avez été des piliers solides, présents à chaque étape. Merci pour votre soutien sans faille, vos motivations, et vos encouragements.

Enfin, à toutes ces âmes bienveillantes croisées sur ce chemin, merci pour votre soutien inattendu qui a rendu ce voyage encore plus mémorable. Votre contribution a donné vie à ces pages.

Dédicace

(◇)



À mon cher père Tidiani Demba Guindo

À ma mère Lallé Diakité

À ma grande sœur Mah Eiye

À mon grand frère Mamadou Lamine Guindo

À ma tante Kia Sy

À mes amis(e) Nana, Mao, Doussou, Oumou, Fatim et

Mamadou

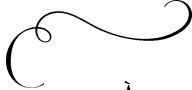
Et à tous les autres membres de ma famille.

Sadio



Dédicace

(◇)



À ma maman Mariam Sissoko et ma tante Fadima Sissoko

À mon très cher oncle Ibrahima Sissoko

À la mémoire de père Sériba

À mes sœurs Adiara, Fatim, Koro, Kadi, Mimi et Oumou

À mes adorables neveux et nièces

À mes amies Assou, Fadi, Mariam et Marthe

À toute ma famille.

Alimata



Résumé

Dans le domaine médical, la classification d'images a connu un grand succès suite aux progrès des technologies d'acquisition d'images et de l'apprentissage profond (Deep Learning), qui est considéré comme l'outil le plus performant en reconnaissance visuelle, automatisant l'extraction des caractéristiques et développant des représentations hiérarchiques des structures et motifs permettant ainsi de renforcer l'efficacité des méthodes de classification. Dans cette optique, nous avons développé un système automatique visant à classifier efficacement les cellules sanguines périphériques en exploitant les réseaux du Deep Learning par l'utilisation des techniques d'apprentissage par transfert pour une meilleure performance et un temps de calcul raisonnable. Trois conceptions ont été proposées sur la base des architectures convolutives VGG16, InceptionV3 et ResNet50, connues pour leurs grandes performances sur le dataset ImageNet. La première réalisation est une combinaison de VGG16 et InceptionV3, la deuxième est une concaténation de InceptionV3 et ResNet50 et enfin la troisième est une association de VGG16 et ResNet50. Les expériences ont été menées sur l'ensemble de données "Peripheral Blood Cell" (PBC) composé de 17 092 images réparties en huit (8) classes distinctes. Les résultats obtenus se sont avérés très satisfaisants, avec une précision maximale de 99%.

Mots-clés: Deep Learning, classification, apprentissage par transfert, Image médicale, cellules sanguines.

Abstract

In the medical field, image classification has achieved great success following advances in imaging technologies and deep learning, which is considered the most effective tool in visual recognition by automating feature extraction and developing hierarchical representations of structures and patterns, thereby enhancing the efficiency of classification methods. In this context, we have developed an automatic system aimed at efficiently classifying peripheral blood cells by leveraging Deep Learning networks through the use of transfer learning techniques for better performance and reasonable computation time. Three designs were proposed based on the convolutional architectures VGG16, InceptionV3, and ResNet50, known for their high performance on the ImageNet dataset. The first implementation is a combination of VGG16 and InceptionV3, the second is a concatenation of InceptionV3 and ResNet50, and the third is an association of VGG16 and ResNet50. Experiments were conducted on the "Peripheral Blood Cell" (PBC) dataset, consisting of 17 092 images divided into eight distinct classes. The results obtained were very satisfactory, with a maximum accuracy of 99%.

Keywords : Deep Learning, classification, transfer learning, medical imaging, blood cells.

ملخص

تصنيف الصور هو مفهوم ليس بالجديد، لكنه شهد في المجال الطبي، حققت حُلْسْفَحْتَن د«امگس نجاحًا كبيرًا عقب التقدم في تقنيات الحصول على الصور والتعلم العميق (Deep Learning) ، الذي يعتبر الأداة الأكثر فعالية في التعرف البصري عن طريق أتمتة استخراج الخصائص وتطوير تمثيلات هرمية للهياكل والأنماط، مما يعزز فعالية طرق التصنيف (classification) . في هذا السياق، قمنا بتطوير نظام تلقائي يهدف إلى تصنيف خلايا الدم المحيطية بكفاءة من خلال الاستفادة من شبكات التعلم العميق (Deep Learning) باستخدام تقنيات إيبيرنتسگ ڤرترنسڤرت لتحقيق أداء أفضل ووقت حساب معقول. تم اقتراح ثلاثة تصاميم بناءً على البنى التلافيفية VGG16 و InceptionV3 و ResNet50 ، المعروفة بأدائها العالي على مجموعة بيانات ImageNet . التنفيذ الأول هو مزيج من VGG16 و InceptionV3 ، والثاني هو دمج بين InceptionV3 و ResNet50 ، والثالث هو اقتران بين VGG16 و ResNet50 . أجريت التجارب على مجموعة بيانات آخلية الدم المحيطية (Peripheral Blood Cell) (PBC) المكونة من ١٧٠٩٢ صورة مقسمة إلى ثمانية فئات متميزة. كانت النتائج المحصلة مرضية للغاية، حيث وصلت الدقة القصوى إلى ٩٩٪.

الكلمات الرئيسية : التعلم العميق، التصنيف، التعلم بالنقل، التصوير الطبي، خلايا الدم.

TABLE DES MATIÈRES

Introduction générale	17
1 Imagerie Médicale	20
1.1 Introduction	20
1.2 Image Numérique	20
1.2.1 Définitions	20
1.2.2 Image noir et blanc	21
1.2.3 Image niveaux de gris	21
1.2.4 Image couleur	21
1.3 Les technologies d'imagerie médicale	23
1.3.1 Définitions	23
1.3.2 Radiographie	23
1.3.3 Scanner	24
1.3.4 Imagerie par Résonance Magnétique (IRM)	24
1.3.5 Échographie	25
1.3.6 Microscopie	25
1.4 Conclusion	27
2 Cellules sanguines	29
2.1 Introduction	29
2.2 Rôle et composition du sang	29
2.3 Hématopoïèse	30
2.4 Préparation des frottis sanguins	31
2.4.1 Réalisation d'un frottis sanguin	31
2.4.2 Coloration du frottis	32
2.4.2.1 Coloration May-Grünwald-Giemsa (MGG)	32
2.4.2.2 Les étapes de la coloration	32
2.5 Les types de cellules sanguines	33
2.5.1 Les globules rouges (Hématies ou Érythrocytes)	33
2.5.2 Les globules blancs (Leucocytes)	33
2.5.3 Les plaquettes (Thrombocytes)	35

2.6	Conclusion	36
3	Classification d'images	38
3.1	Introduction	38
3.2	Intelligence Artificielle (IA)	38
3.2.1	Définitions	38
3.2.2	Apprentissage Automatique (Machine Learning)	39
3.2.2.1	Définition	39
3.2.2.2	Apprentissage supervisé	40
3.2.2.3	Apprentissage non supervisé	41
3.2.2.4	Apprentissage par renforcement	42
3.2.3	Apprentissage Profond (Deep Learning)	42
3.2.3.1	Convolutional Neural Networks (CNN)	43
3.2.3.2	Les types d'architectures CNN	45
3.2.3.3	Autres approches du Deep Learning	47
3.2.3.4	Transfert learning	48
3.2.3.5	Le sur-ajustement (Overfitting)	49
3.3	Classification d'images cytologiques	50
3.3.1	Définitions	50
3.3.2	Types de classifications	50
3.3.2.1	Classification Binaire (Binary Classification)	50
3.3.2.2	Classification Multi-classes	51
3.3.2.3	Classification multi-label	51
3.3.3	État de l'art	51
3.4	Conclusion	59
4	Implémentation	61
4.1	Introduction	61
4.2	Outils utilisés	61
4.2.1	Langage utilisé (Python)	61
4.2.2	Google Colab (Colaboratory)	62
4.2.3	Les bibliothèques	62
4.3	Dataset utilisé	64
4.4	Méthodes proposées	65
4.4.1	Modèle 1 (VGG16 et InceptionV3)	66
4.4.2	Modèle 2 (InceptionV3 et ResNet50)	68
4.4.3	Modèle 3 (VGG16 et ResNet50)	69
4.5	Pré-traitement	70
4.5.1	Chargement du dataset	70
4.5.2	Répartition des données	71
4.5.3	Normalisation des données	71
4.6	Apprentissage des modèles	72
4.7	Résultats et Discussion	74
4.7.1	Prédiction avec l'interface Gradio	80

4.7.2	Difficultés rencontrées	82
4.8	Conclusion	83
	Conclusion générale et perspectives	85

LISTE DES FIGURES

1.1	Image noir et blanc [Souad and Tinhinane, 2017]	21
1.2	Image niveau de gris [Souad and Tinhinane, 2017]	21
1.3	Image couleur [Benali et al., 2022]	22
1.4	Représentation spatiale du modèle RGB [Benomar, 2018]	22
1.5	Représentation spatiale du modèle L*a*b [Benomar, 2018]	22
1.6	Représentation spatiale du modèle HSL [Web 2]	23
1.7	Image radiographique du thorax [Mersaoui and Fouiri, 2018]	24
1.8	Les étapes de réalisation du Scanner [Web 1]	24
1.9	Les étapes de l’imagerie par résonance magnétique [Web 1]	25
1.10	Image d’échographie obstétricale [Mersaoui and Fouiri, 2018]	25
1.11	Image Microscopique	26
1.12	Image histologique d’un tissu [Web 5]	26
1.13	Image microscopique de cytologie [Benomar et al., 2014]	27
2.1	Représentation schématique des étapes de séparation des composants du sang: (1) prélèvement du sang, (2) centrifugation, (3) séparation des composants [Web 6]	30
2.2	Cellule souche fille [Web 8]	30
2.3	Hématopoïèse [Web 8]	31
2.4	Les étapes de réalisation du frottis [Benomar, 2018]	32
2.5	Résultat après coloration MGG [Acevedo et al., 2020]	33
2.6	Globules rouges [Benomar, 2018]	33
2.7	Aspect morphologique des polynucléaires et monocytes : (a) neutrophile, (b) éosinophile, (c) basophile, (d) monocyte [Benomar, 2018]	34
2.8	Aspect morphologique des lymphocytes : (a) lymphocyte, (b) plasmocyte [Benomar, 2018]	35
2.9	Aspect morphologique des plaquettes [Acevedo et al., 2020]	35
3.1	Relation entre IA et ML et DL [Web 12]	38
3.2	Système expert [Web 13]	39
3.3	Les types d’apprentissage automatique [Taileb, 2022]	39
3.4	Régression linéaire [Web 14]	40

3.5	Logistique régression [Web 16]	41
3.6	SVM [Web 17]	41
3.7	Exemple de clustering [Web 18]	42
3.8	Processus du machine learning et du deep learning [Web 20]	43
3.9	Exemple d'un réseau de neurone artificiel [Web 21]	43
3.10	Schéma représentant l'architecture d'un CNN [Web 22]	43
3.11	Schéma représentant la convolution [Web 23]	44
3.12	Schéma d'exemple de Maxpooling [Web 24]	44
3.13	Illustration de la couche entièrement connectée [Web 25]	45
3.14	Graphe de la fonction d'activation ReLU [Web 26]	45
3.15	Architecture d'AlexNet [Web 27]	46
3.16	Schéma standard de l'architecture du réseau VGG16 [Web 28]	46
3.17	Schéma de l'architecture du réseau ResNet50 (B: BatchNormalization, R: Fonction ReLU, S: Softmax) [Web 29]	47
3.18	Exemple de Modèle RNN [Web 32]	47
3.19	Exemple du Modèle Aes [Web 34]	48
3.20	Illustration du tranfert learning [Web 35]	48
3.21	Overfitting et Underfitting [Kumar, 2022]	49
3.22	Exemple de classification Binaire (x1,x2: attributs)[Web 37]	50
3.23	Exemple de classification multi-classes (x1,x2: attributs) [Web 37]	51
3.24	Exemple de classification multi-label (X: individus, Y: étiquettes) [Joshi, 2022]	51
3.25	Image microscopique de cytologie [Benomar et al., 2014]	52
4.1	Logo de python [Web 38]	62
4.2	Interface de Google Colab	62
4.3	Logo de Tensorflow [Web 40]	63
4.4	Logo de Keras [Web 41]	63
4.5	Logo de Numpy [Web 42]	63
4.6	Logo de Scikit-learn [Web 43]	64
4.7	Logo de Matplotlib [Web 46]	64
4.8	Logo de Gradio [Web 48]	64
4.9	Exemple des huit (8) types de cellules normales du sang périphérique [Acevedo et al., 2020]	65
4.10	Étapes d'acquisition du dataset (1: Traitement des échantillons, 2: Coloration au MGG, 3: Acquisition automatique de l'image, 4: Labelisation) [Acevedo et al., 2020]	65
4.11	Importation des modèles	66
4.12	Architecture du modèle InceptionV3 [Web 50]	67
4.13	Architecture du Modèle 1	67
4.14	Architecture du Modèle 2	68
4.15	Architecture du Modèle 3	69
4.16	La fonction read_images()	70
4.17	La distribution des huit (8) classes (0: Neutrophiles, 1: Éosinophiles, 2: Basophiles , 3: Lymphocytes, 4: Monocytes, 5: Granulocytes Immatures(IG), 6: Érythroblastes, 7: Plaquettes)	70

4.18	Division des données en données d'apprentissage, de validation et de test	71
4.19	Normalisation des données	71
4.20	Catégorisation des huit (8) classes des cellules	72
4.21	Techniques d'améliorations appliquées	72
4.22	Exemple d'images après augmentation (zoom, rotation, vertical flip, horizontal flip ...etc.)	73
4.23	Apprentissage du Modèle 1	73
4.24	Apprentissage du Modèle 2	74
4.25	Apprentissage du Modèle 3	74
4.26	Performances de l'apprentissage du Modèle 1 (VGG16 et InceptionV3)	75
4.27	Performances de l'apprentissage du Modèle 2 (InceptionV3 et ResNet50)	75
4.28	Performances de l'apprentissage du Modèle 3 (VGG16 et ResNet50)	75
4.29	Les courbes ROC	76
4.30	Matrice de confusion du Modèle 1	77
4.31	Matrice de confusion du Modèle 2	78
4.32	Matrice de confusion du Modèle 3	78
4.33	Deux images de même type cellulaire (neutrophile)	79
4.34	Rapport de classification	79
4.35	Implémentation de l'interface utilisateur	81
4.36	Page d'accueil (à gauche: les inputs, à Droite: les outputs)	81
4.37	Exemple de prédiction (à gauche : image d'entrée, à droite : résultat de classification	81
4.38	Saturation de la RAM	82
4.39	Preuve d'achat des unités	82

LISTE DES TABLEAUX

3.1	Récapitulatif de certaines études proposées dans la littérature pour la classification des images cytologiques	58
4.1	Matrice de confusion binaire	76
4.2	Tableau de comparaison	80

LISTE DES ABRÉVIATIONS

<i>AeS</i>	Autoencoder
<i>AL</i>	Apprentissage Actif
<i>API</i>	Application Programming Interface
<i>AUC</i>	Area Under the Curve
<i>CAD</i>	Computer Aided Diagnostic
<i>CPU</i>	Central Processing Unit
<i>DL</i>	Deep Learning
<i>ECP</i>	Eosinophil Cationic Protein
<i>GPU</i>	Graphics Processing Unit
<i>HSL</i>	Hue , Saturation, Luminance
<i>IA</i>	Intelligence Artificielle
<i>IgE</i>	Immunoglobuline E
<i>ILSVRC</i>	Large Scale Visual Recognition Challenge
<i>IRM</i>	Imagerie par Résonance Magnétique
<i>KNN</i>	K-Nearest-Neighbor
<i>LR</i>	Linear Regression
<i>MAES</i>	Autoencodeur Masqué
<i>MDP</i>	Major Basic Protein
<i>MGG</i>	May-Grünwald-Giemsa
<i>MIC</i>	Mutual Information Classification

<i>ML</i>	Machine Learning
<i>NK</i>	Natural Killer
<i>PBC</i>	Pheripheral Blood Cell
<i>QDA</i>	Analyse Discriminante Quadratique
<i>ReLU</i>	Rectified Linear Unit
<i>RGB</i>	Red, Green, Blue
<i>RL</i>	Reinforcement Learning
<i>RNN</i>	Reccurent Neural Network
<i>ROC</i>	Receiver Operating Characteristic
<i>RVB</i>	Rouge , Vert, Bleu
<i>SVM</i>	Support Vector Machine
<i>ViT</i>	Vision Transformer
<i>WBC</i>	White Blood Cell
<i>YOLO</i>	You Only Look Once

INTRODUCTION GÉNÉRALE

L'avènement de la technologie informatique et des systèmes d'analyse d'images médicales a permis le développement d'outils et méthodes pour la classification automatique des cellules dans les images de cytologies sanguines. Cependant, les progrès significatifs dans ce domaine ont été réalisés au fil des décennies grâce aux avancées technologiques et à l'intégration de techniques telles que la vision par ordinateur et l'apprentissage automatique, contribuant ainsi au développement des systèmes d'aide à la décision dans les domaines médicaux, industriels, météorologiques et bien d'autres domaines d'application.

Les méthodes d'analyse d'images et de reconnaissance de formes ont été largement utilisées dans le domaine de l'analyse pathologique pour aider les médecins à étudier différents motifs/cellules dans les images microscopiques. On distingue deux types d'examen: l'examen histologique, qui consiste en l'observation de la coupe microscopique d'un tissu et l'examen cytologique, où les prélèvements sont étalés sur une lame puis fixés et colorés afin de reconnaître les différentes cellules présentes. La lecture (screening) et l'analyse de ces lames se font au microscope par le pathologiste dans le but soit de détecter des cellules anormales, soit de compter les cellules selon leurs types afin d'établir un diagnostic final.

Malgré le développement important des outils de diagnostic modernes, l'anatomie pathologique reste le principal moyen par lequel la plupart des diagnostics sont posés; cependant, le comptage manuel des leucocytes présente des problèmes majeurs, tels que le processus qui prend du temps, la nécessité d'un pathologiste expérimenté, la variabilité interprétative, la présence de cellules normales similaires et les défis dans l'identification des cellules anormales rares.

Un CAD (computer aided diagnostic) ou système automatique pour la classification des cellules peut être une solution pour atténuer ces problèmes et fournir des résultats plus précis et plus rapides par rapport à l'analyse manuelle dans l'objectif principal d'aider le médecin dans son travail. Dans notre projet de mémoire de fin d'études, l'objectif est de concevoir un modèle d'apprentissage profond visant à résoudre le problème de classification des cellules sanguines périphériques par catégories, ce qui est une étape importante dans le processus de diagnostic médical des maladies liées au sang. Pour cela, des architectures basées sur les réseaux de neurones convolutifs (CNN) sont réalisées en utilisant des modèles pré-entraînés comme point de départ (VGG16, InceptionV3 et ResNet50),

cela représente l'approche d'apprentissage par transfert pour faire une classification automatique des cellules en différents sous-types (neutrophiles, éosinophiles, basophiles, lymphocytes, monocytes, granulocytes immatures (promyélocytes, myélocytes et métamyélocytes), érythroblastes et plaquettes ou thrombocytes) à partir des images cytologiques. Cette méthode permet de tirer parti des connaissances acquises par un modèle sur une tâche pour améliorer ses performances en l'adaptant à notre tâche spécifique, tout en réduisant le temps, les ressources et les données d'entraînement nécessaires. Des expérimentations sont menées sur un ensemble de données (dataset) contenant 17 092 images des cellules sanguines périphériques, chacune étant associée à un sous-type [Acevedo et al., 2020].

Notre mémoire est structuré en quatre chapitres. Le premier chapitre présente des notions de base sur l'imagerie médicale en général et plus particulièrement sur les images microscopiques de cytologies sanguines. Le deuxième chapitre est dédié aux cellules sanguines, nous exposons ces notions fondamentales ainsi que les caractéristiques et rôles de chaque type de cellule. Dans le troisième chapitre, nous abordons la classification d'images de manière générale, en présentant quelques approches de classification et quelques travaux sur la classification d'images cytologiques. Finalement, le quatrième chapitre est consacré à la partie résultats et expérimentation, où nous détaillons les méthodes proposées, la base de données (dataset) utilisée et nous terminons avec une discussion et une analyse sur les modèles réalisés. Une conclusion générale est donnée à la fin du mémoire, rappelant ainsi le contexte du travail avec des perspectives de recherches futures.

Chapitre 1 : Imagerie Médicale

IMAGERIE MÉDICALE

1.1 Introduction

L'imagerie médicale a fait son apparition avec la découverte des rayons X par Wilhelm Röntgen (physicien allemand) en 1896 [Web 1]. C'est certainement l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Faisant appel à de nombreuses technologies; il est de plus en plus utilisée dans le diagnostic de plusieurs maladies (cancers, épilepsie ...etc.) en complément d'un examen clinique et d'autres investigations comme les examens biologiques ou encore les tests neuropsychologiques. Et est considéré également comme un élément essentiel à la recherche clinique et à la mise au point de nouveaux traitements. L'imagerie recouvre à une grande variété de technologies développées grâce à l'exploitation des grandes découvertes de la physique du 20e siècle [Web 1] :

- Les ondes radio et rayons X;
- La radio activité de certains éléments;
- Les champs magnétiques.

1.2 Image Numérique

1.2.1 Définitions

L'image est une représentation d'un objet, d'une personne, d'une scène, et bien d'autres représentations à travers la peinture, la sculpture, le dessin, la photographie, le film,...etc.

C'est aussi un ensemble d'informations structurées qui, après affichage sur écran, elle devient significative pour l'œil humain [Sara, 2020].

Une image numérique est tout type d'image acquise, traitée et stockée sous une forme codée représentable par des nombres (valeurs numériques). Il existe trois type de couleur dans l'image numérique:

- Image noir et blanc;
- Image niveau de gris;
- Image couleur.

1.2.2 Image noir et blanc

Un pixel prend la valeur 0 ou 1 selon qu'il prend la couleur blanc ou noir (voir Figure 1.1). Pour mémoriser cette information, un bit suffit. La profondeur de l'image est donc de 1 bit par pixel.



Figure 1.1: Image noir et blanc [Souad and Tinhinane, 2017]

1.2.3 Image niveaux de gris

De manière générale, les images niveaux de gris sont des images de profondeur de 8 bits, c'est à dire la valeur de chaque pixel est comprise entre $[0 \dots 255]$, où la valeur 0 représente le noir et 255 le blanc (voir Figure 1.2). Ce type d'image est fréquemment utilisé pour représenter des photos en noir et blanc ou du texte.



Figure 1.2: Image niveau de gris [Souad and Tinhinane, 2017]

1.2.4 Image couleur

Sont basées sur trois couleurs primaires, à savoir le Rouge, le Vert et le Bleu (RVB ou RGB en anglais)(voir Figure 1.3 et Figure 1.4). Chaque couleur utilise 8 bits, donc chaque pixel nécessite $3 * 8 = 24$ bits permettant de représenter plusieurs couleurs ($256 \times 256 \times 256$ couleurs). Dans le domaine de l'imagerie, c'est la plus utilisée parmi les types de représentation d'image, même si elle présente quelques inconvénients à savoir:

- L'existence d'un grand nombre de systèmes RVB [Vandenbroucke, 2000];
- Les coordonnées et les composantes trichromatiques peuvent prendre des valeurs négatives [Benomar, 2018];
- L'existence de couleurs non reproductible par addition des trois spectres [Benomar, 2018];
- Deux stimuli de couleur peuvent avoir la même chrominance avec des composantes trichromatiques (R,G,B) différentes, à cause des composantes trichromatiques qui sont liées à la luminance [Benomar, 2018].

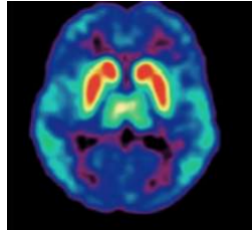


Figure 1.3: Image couleur [Benali et al., 2022]

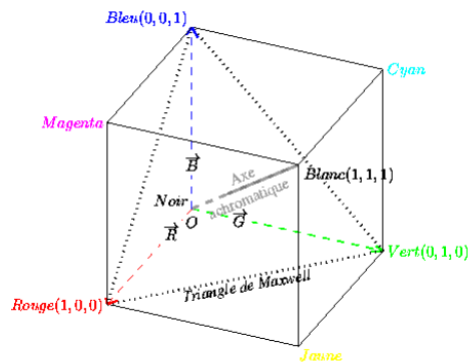
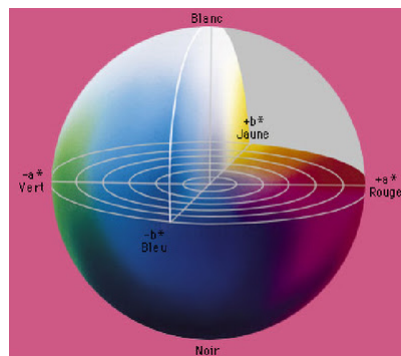


Figure 1.4: Représentation spatiale du modèle RGB [Benomar, 2018]

IL existe d'autres espaces couleurs comme par exemple:

- Espace couleur L^*a^*b

Également appelé CIE 1976 lab où L représente la luminosité perceptive avec une valeur de 0 (noir) à 100 (blanc), a^* et b^* les quatre couleurs (rouge, vert, bleu et jaune). Il est basé sur le modèle des couleurs opposées. Dans l'espace, sa représentation est sous forme de sphère légèrement aplatie (voir Figure 1.5). Le lab peut être utilisé pour calculer la distance euclidienne entre deux points, ce qui lui permet d'évaluer la différence calorimétrique entre deux stimuli de couleurs.

Figure 1.5: Représentation spatiale du modèle L^*a^*b [Benomar, 2018]

- Espace couleur HSL

Le modèle HSL (Hue, Saturation, Luminance) ou en français TSL, s'appuyant sur les travaux du peintre Albert H.Munsell (qui créa l'Atlas de Munsell), est un modèle de représentation dit "naturel", c'est-à-dire proche de la perception physiologique de la couleur par l'oeil humain [Web 2]. L'un des principaux avantages de la couleur HSL par rapport à la couleur RGB est que les

couleurs complémentaires sont situées les unes en face des autres, ce qui rend l'ensemble du système très intuitif (voir Figure 1.6) .

Le HSL permet de décomposer la couleur selon certaines critères:

- **La teinte (H)** représente la perception de la couleur;
- **La saturation (S)** correspond au niveau de coloration, il décrit la pureté de la couleur;
- **La luminance (L)** indique la quantité de lumière de la couleur (clair ou sombre).

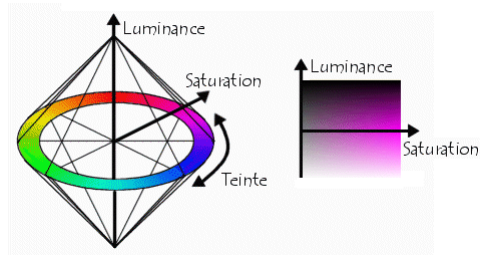


Figure 1.6: Représentation spatiale du modèle HSL [Web 2]

1.3 Les technologies d'imagerie médicale

1.3.1 Définitions

L'imagerie médicale est une spécialité médicale qui consiste à produire des images du corps humain vivant et à les interpréter à des fins de diagnostic, thérapeutique (imagerie interventionnelle) ou surveillance de l'évolution des pathologies [Web 3].

Il existe de nombreux types d'imagerie médicale et de nouveaux moyens sont développés au fur et à mesure que la technologie progresse. Parmi les méthodes d'imagerie médicale les plus couramment utilisées en médecine, on peut citer d'une part les méthodes tomographiques basées soit sur les rayons X, soit sur la résonance magnétique (IRM), les méthodes échographiques et enfin les méthodes optiques utilisant les rayons.

1.3.2 Radiographie

La radiographie repose sur l'utilisation des rayons X, qui ont la propriété de traverser les tissus de manière plus ou moins importante selon leur densité (voir Figure 1.7). Ainsi, une source émettrice de rayon X est placée devant le corps à radiographier et un détecteur est placé à l'arrière du corps. Les photons émis vont traverser le corps en étant plus ou moins absorbés par les tissus rencontrés sur leur chemin. Cela permet de différencier les os des muscles sur le cliché final [Web 1].



Figure 1.7: Image radiographique du thorax [Mersaoui and Fouiri, 2018]

1.3.3 Scanner

Un dispositif médical également connu sous le nom tomographie par ordinateur ou TDM (voir Figure 1.8), est une méthode d'exploration qui utilise les rayons X à faible dose pour générer des images d'une région spécifique du corps. Ce dispositif réalise des images tridimensionnelles (3D) en effectuant une rotation simultanée de la source émettrice de rayons X et du détecteur autour du corps. Les projections intermédiaires en deux dimensions (2D) sont ensuite traitées par un ordinateur pour produire des images tridimensionnelles (3D).

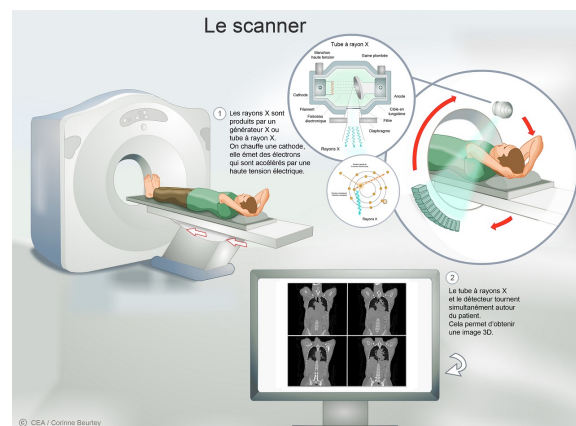


Figure 1.8: Les étapes de réalisation du Scanner [Web 1]

1.3.4 Imagerie par Résonance Magnétique (IRM)

L'IRM est un examen qui permet de visualiser la structure anatomique du corps et de fournir des informations sur des lésions qui ne sont pas visibles sur les radiographies standards, l'échographie ou le scanner (voir Figure 1.9). Son principe est basé sur les propriétés magnétiques des nombreux atomes d'hydrogène, représentant plus de 80% du corps humain. Il utilise un puissant champ magnétique pour analyser la réaction des différents tissus. Les données recueillies sont ensuite traitées par des logiciels afin d'obtenir des images en 2D ou 3D de l'intérieur du corps avec une grande précision. Le développement de l'IRM a permis d'ouvrir des perspectives de compréhension et de recherche sur les organes les plus complexes (le cerveau, la moelle épinière, les muscles ...etc).

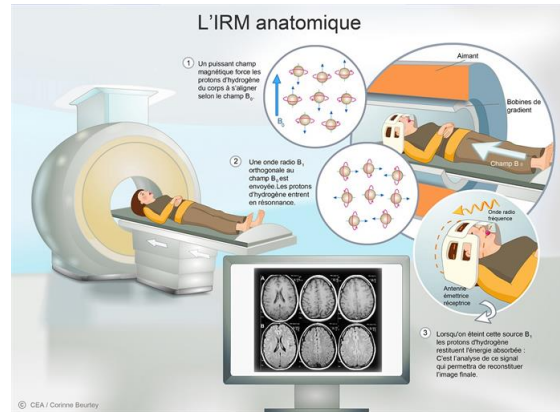


Figure 1.9: Les étapes de l'imagerie par résonance magnétique [Web 1]

1.3.5 Échographie

C'est un examen qui permet de visualiser les organes et les tissus internes du corps. Il repose sur l'utilisation des ondes ultrasonores qui sont produites grâce à un transducteur (sonde émettrice et réceptrice des ondes). Une sonde envoie un faisceau d'ultrasons dans la partie du corps à examiner. Les tissus sont traversés par ces ondes qui génèrent un écho dont l'intensité varie selon la densité des tissus (plus un tissu est dense, plus l'écho est important). Un traitement informatique de ces échos permet de faire une visualisation des organes examinés. Il existe différentes modalités d'échographie en fonction des tissus étudiés, telles que l'échographie cardiaque (ou échocardiographie), l'échographie mammaire, l'échographie obstétricale, ...etc. Par exemple, pendant la grossesse, l'échographie obstétricale permet d'observer le fœtus (voir Figure 1.10), de distinguer les organes et le squelette, dans le but de surveiller le bon déroulement de la grossesse et de détecter d'éventuelles malformations.



Figure 1.10: Image d'échographie obstétricale [Mersaoui and Fouiri, 2018]

1.3.6 Microscopie

Il s'agit des techniques permettant d'obtenir et d'observer les différentes structures biologiques invisibles à l'œil nu, soit en raison de leur taille, soit de leurs couleurs. Une onde est envoyée sur la préparation ou émise par la préparation; cette onde est captée par un objectif qui la concentre et passe par un oculaire qui crée une image observable. cette image est observée soit à l'œil nu, soit photographiée, soit enregistrée par caméra et stockée sur ordinateur pour traitement [Semmame, 2020]. L'appareil utilisé pour rendre cette observation possible est le microscope (voir Figure 1.11). Il existe plusieurs techniques qui utilisent cet appareil pour visualiser les composantes cellulaires. Parmi ces techniques, on peut citer: l'histologie et la cytologie.

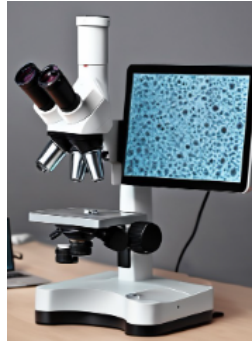


Figure 1.11: Image Microscopique

1. Histologie

C'est une branche de l'anatomie qui s'intéresse à la structure microscopique des tissus et cellules composant un organisme, afin de mieux comprendre et analyser leur fonctionnement [Web 4] (voir Figure 1.12). En médecine, elle permet d'apporter plus de précision au dépistage et au diagnostic de certaines pathologies, mais elle contribue également à plusieurs travaux de recherche. Les techniques histologiques utilisent des microscopes pour étudier des coupes de tissus, qui sont le fruit de procédures, qui requièrent plusieurs étapes successives à savoir: prélèvement, fixation, inclusion, coupe, coloration et montage sur des lames.

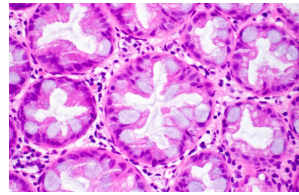


Figure 1.12: Image histologique d'un tissu [Web 5]

2. Cytologie

Elle consiste en l'étude des cellules quant à leur structure, leur composition, leur rôle et leurs propriétés chimiques [Web 5]. La cytologie a pour but de détecter les changements pathologiques au niveau cellulaire en étudiant des cellules individuelles et des structures cellulaires. En médecine, l'imagerie microscopique permet l'examen des cellules prélevées à partir de plusieurs tissus et fluides du corps. Un examen courant est le frottis cytologique où des échantillons sont prélevés, étalés sur une lame de verre, colorés puis examinés au microscope (voir Figure 1.13).

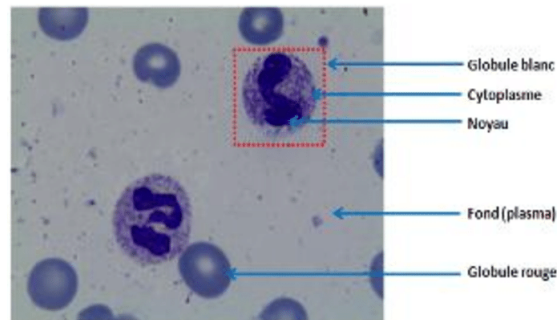


Figure 1.13: Image microscopique de cytologie [Benomar et al., 2014]

1.4 Conclusion

L'imagerie médicale a révolutionné le monde de la médecine, notamment à travers l'avènement de la technologie qui a permis la mise en place des différents moyens d'examen tels que le scanner, la radiographie, l'IRM, l'échographie, la microscopie ...etc. Contribuant ainsi à la précision des diagnostics, à la prévention de plusieurs maladies, à la compréhension de nombreuses pathologies ...etc. Dans le chapitre suivant, nous nous intéressons aux notions fondamentales des cellules sanguines ainsi qu'aux caractéristiques et aux rôles de chaque type de cellule dans le corps humain.

Chapitre 2 : Cellules sanguines

CELLULES SANGUINES

2.1 Introduction

Le sang est composé d'un ensemble de cellules mélangées avec le plasma. L'étude de la morphologie des cellules sanguines a beaucoup progressé au cours des dernières années, grâce au développement d'analyseurs d'hématologie ¹ automatisés, qui permettent d'identifier des cellules sanguines normales. Malgré cela, une analyse microscopique est toujours nécessaire pour une bonne reconnaissance sur frottis sanguin des cellules pathologiques. Cette analyse intègre l'étape de préparation et de coloration des frottis médullaires et sanguins, suivie de l'acquisition d'images microscopiques.

2.2 Rôle et composition du sang

Le sang est un élément extrêmement complexe de par sa composition [Web 6]; il représente 7% ou 8% du poids corporel d'un individu normale (soit un volume de 5 à 7 litres). Après centrifugation ², il se divise en différents éléments (voir Figure 2.1):

- Le plasma représente 55% du sang;
- Les globules rouges constituent 45%;
- Les globules blancs et les plaquettes forment tous les deux 1% du sang.

Ces cellules (globules rouges, globules blancs et plaquettes) sont en suspension dans le plasma et circulent de façon unilatérale dans le corps grâce aux contractions cardiaques.

¹Hématologie: étude des maladies liées au sang

²La centrifugation est un procédé de séparation des composés d'un mélange en fonction de leur différence de densité en les soumettant à une force centrifuge.



Figure 2.1: Représentation schématique des étapes de séparation des composants du sang: (1) prélèvement du sang, (2) centrifugation, (3) séparation des composants [Web 6]

Le rôle principal du sang est d'assurer la circulation et la diffusion des éléments indispensables à la survie de l'être humain dans tout le corps. Le sang circule dans les artères et les veines pour délivrer l'oxygène et les matières organiques vers les cellules de tout l'organisme et ensuite récupérer leurs déchets (dioxyde de carbone ou composants azotés). Il est également responsable du transport des cellules immunitaires (globules blancs), qui permettent au corps de lutter contre les maladies.

2.3 Hématopoïèse

C'est l'ensemble des phénomènes qui concourent à la fabrication et au remplacement continu et régulé des cellules sanguines [Web 7].

Les types de cellules sanguines descendent tous de la même cellule souche (l'hématopoïèse). Une caractéristique des cellules souches est qu'elles se divisent de manière asymétrique. Ça veut dire qu'une cellule fille reste une cellule souche du même type, tandis que l'autre cellule fille acquiert une nouvelle identité (voir Figure 2.2) [Web 8].

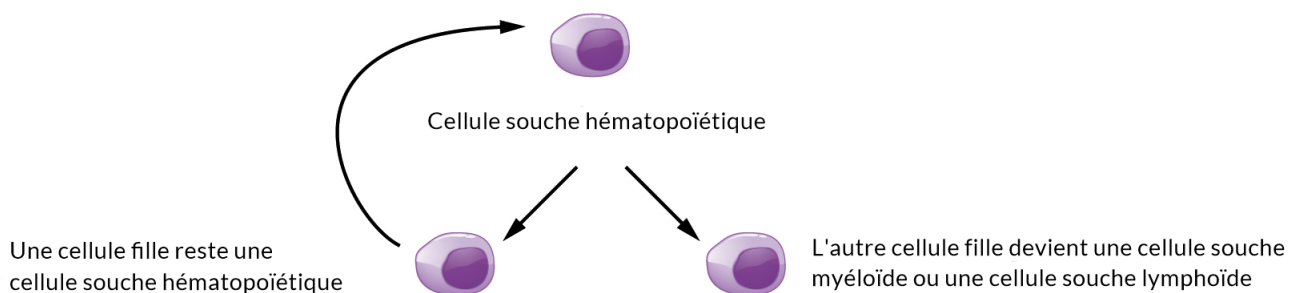


Figure 2.2: Cellule souche fille [Web 8]

Les cellules souches myéloïdes et les cellules souches lymphoïdes se divisent également de manière asymétrique. Leurs cellules filles non-souches deviennent les cellules matures du sang. Les cellules souches myéloïdes donnent naissance aux globules rouges, aux plaquettes et à certains types de globules blancs, alors que les cellules souches lymphoïdes donnent naissance aux types de globules blancs classés comme lymphocytes [Web 8] (voir Figure 2.3).

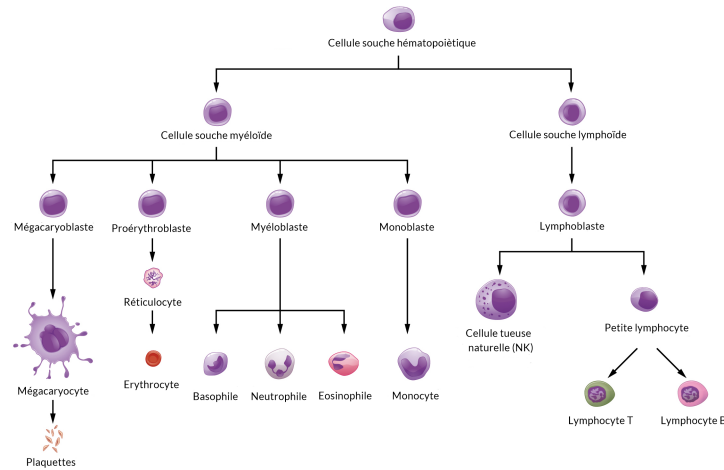


Figure 2.3: Hématopoïèse [Web 8]

L'hématopoïétique, la myéloïde et les cellules souches lymphoïdes se divisent pendant toute la vie d'une personne, générant de nouvelles cellules sanguines pour remplacer les cellules anciennes et usées [Web 8].

2.4 Préparation des frottis sanguins

Pour observer la structure des cellules sanguines au microscope, les étapes suivantes sont nécessaires : la préparation des frottis, la coloration par la méthode May-Grünwald-Giemsa (voir 2.4.2.1) ensuite la lecture et l'acquisition des images au microscope.

2.4.1 Réalisation d'un frottis sanguin

- Immédiatement après le prélèvement sanguin, déposer une goutte de sang à l'extrémité de la lame porte-objet [Web 9];
- Placer devant la goutte une deuxième lame et former un angle de 30 à 45° avec la première lame [Web 9] (voir Figure 2.4);
- Reculer cette deuxième lame inclinée, jusqu'au contact de la goutte de sang pour la faire s'étendre, par capillarité, sur toute la largeur de la lame inclinée [Web 9] (voir Figure 2.4);
- Déplacer cette deuxième lame, d'un mouvement rapide vers l'avant, en glissant sur la première lame [Web 9] (voir Figure 2.4);
- Le frottis obtenu [Web 9] :
 - Ne doit être ni trop épais, ni trop mince;
 - Ne doit atteindre ni les bords, ni les extrémités de la lame;
 - Ne doit pas présenter de trous, ni de stries;
 - Doit être aussi régulier que possible.

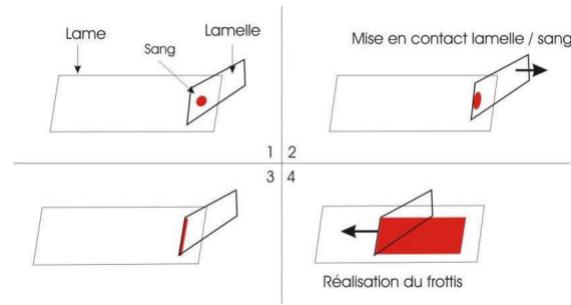


Figure 2.4: Les étapes de réalisation du frottis [Benomar, 2018]

2.4.2 Coloration du frottis

2.4.2.1 Coloration May-Grünwald-Giemsa (MGG)

La coloration de May-Grünwald-Giemsa (MGG) également appelée coloration de Pappenheim, est une technique de coloration utilisée en hématologie cellulaire afin de différencier les différentes cellules sanguines. Elle figure parmi les colorations polychromes dites de Romanowsky, dont le principe est un mélange d'éosine Y et d'azur B de méthylène. Et permet la coloration des différents éléments cellulaires (noyau, cytoplasme et granulations), permettant ainsi une identification précise. Elle associe deux colorants : May-Grünwald et le Giemsa [Web 10].

Le May-Grünwald colore les éléments acidophiles ainsi que les granulations neutrophiles des leucocytes et le Giemsa colore le cytoplasme des monocytes, des lymphocytes et la chromatine des noyaux [Web 10].

2.4.2.2 Les étapes de la coloration

1. Fixation : les échantillons sont fixés sur une lame de verre à l'aide de l'alcool méthylique;
2. Coloration au May-Grünwald : la première étape de coloration utilise le May-Grünwald, qui consiste à immerger la lame dans le colorant May-Grünwald pendant 2 minutes. Ce qui induit une coloration orthochromatique (couleur rose ou orange pour les éléments acidophiles due à l'éosine et couleur bleue ou violette pour les éléments basophiles et neutrophiles due au bleu de méthylène) [Benomar, 2018];
3. Coloration au Giemsa : Ensuite, la lame est placée dans le colorant Giemsa pendant 20 à 30 minutes. Ce qui induit une coloration métachromatique (couleur rouge pour les éléments azurophiles) [Benomar, 2018];
4. Séchage : enfin, attendre le séchage complet de la lame à l'air avant observation au microscope.

Après cette étape de coloration, le cytoplasme des lymphocytes et des monocytes ainsi que les nucléoles apparaissent en bleu (voir Figure 2.5). Les granulations primaires en rouge-violet, les granulations des éosinophiles en rouge-orangé (voir Figure 2.5) [Benomar, 2018].

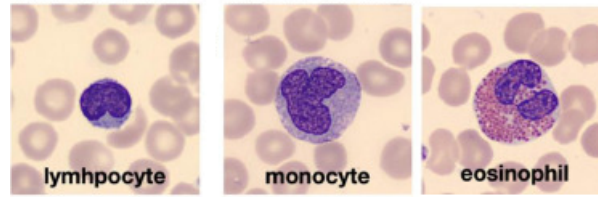


Figure 2.5: Résultat après coloration MGG [Acevedo et al., 2020]

2.5 Les types de cellules sanguines

2.5.1 Les globules rouges (Hématies ou Érythrocytes)

Sont des cellules anucléées dont le constituant essentiel est une hémoprotéine de liaison de l'oxygène : l'hémoglobine (environ 14,5 g / 100 ml). Le rôle principal de ces cellules est d'assurer le transport de l'oxygène et du gaz carbonique entre les alvéoles pulmonaires et les tissus [Kohler, 2010]. Ils ont un diamètre d'environ $7\mu\text{m}$, un volume moyen de $90\mu\text{m}^3$, et ont une durée de vie moyenne d'environ 120 jours, habituellement ils ont une coloration pourpre (voir Figure 2.6). Une fois arrivés à la fin de leur vie, ils sont retenus dans la rate où ils sont phagocytés par les macrophages [Tagliasacchi and Carboni, 1997].



Figure 2.6: Globules rouges [Benomar, 2018]

2.5.2 Les globules blancs (Leucocytes)

Sont des cellules du système immunitaire chargées de défendre l'organisme contre des infections virales ou parasitaires. Elles contiennent un noyau plurilobé. Les lobes sont reliés les uns aux autres par des ponts fins de chromatine. Dans le cytoplasme, il existe deux types de granulations : des granulations non spécifiques primaires, riches en hydrolases et en peroxydases, communes à l'ensemble des polynucléaires et des granulations secondaires spécifiques à chaque groupe ayant des propriétés tinctoriales différentes. Dans la cellule mature, les granulations non spécifiques diminuent [Kohler, 2010]. Les leucocytes se divisent en deux catégories: les polynucléaires (ou granulocytes) et les hyalins (ou agranulocytes).

1. **Les leucocytes polynucléaires:** se divisent en trois types de cellules: les neutrophiles, les basophiles et les éosinophiles.

(a) Polynucléaires neutrophiles:

Ce sont les polynucléaires les plus nombreux - 40 à 75 % de l'ensemble des globules blancs. Leur durée de vie est de l'ordre de 24 heures [Kohler, 2010]. Ils ont environ $12\mu\text{m}$ de diamètre, le noyau a une chromatine condensée segmentée en 2 à 5 lobes, réunis par un

filament fin de chromatine³ mais la majorité a 3 lobes, le fond du cytoplasme est coloré en beige-rosé avec de très nombreuses granulations neutrophiles d'aspect marron-beige (voir Figure 2.7). Ils jouent un rôle de phagocytose,⁴ leur permettant de détruire les micro-organismes pathogènes (bactéries, virus, parasites etc..) en les ingérant et en les digérant.

(b) **polynucléaires basophiles:**

Ils sont les moins nombreux des polynucléaires; ils représentent moins de 1% des globules blancs. Leur durée de vie est de 3 à 4 jours, avec un diamètre de 10 à 14 μ m, colorés en violet par le MGG. Ils ont un noyau peu ou pas segmenté, avec une forme bilobée ou irrégulière (voir Figure 2.7).

Ils permettent de défendre l'organisme contre certaines parasites et manifestations allergiques grâce à leur récepteur aux IgE et leur contenu en histamin.

(c) **Polynucléaires éosinophiles**

Ce sont des cellules assez rares dans le sang, représentant 1 à 3% des globules blancs. Elles ont une durée de vie de 8 à 10 jours et ont un diamètre de 10 à 14 μ m, le noyau est généralement bilobé, avec une chromatine condensée, leur cytoplasme contient plusieurs granulations spécifiques de grande taille et apparaît en orangé après coloration au MGG (voir Figure 2.7).

Elles ont une capacité phagocytaire et bactéricide moins importante que les neutrophiles. Elles permettent également la destruction de parasites par l'utilisation de protéines de haut poids moléculaire telles que l'Eosinophil Cationic protein (ECP) et la Major Basic Protein (MBP)

2. Les leucocytes hyalins (agranulocytes)

(a) **Les monocytes:**

Issus de la moelle, ils sont les plus grands des cellules sanguines: leur noyau est encoché mais non polylobé, leur cytoplasme "ciel d'orage" contient de très fines granulations. Après 2 ou 3 jours dans le sang, ils gagnent les tissus et deviennent des macrophages à fonction phagocytaire moins spécifiquement bactérienne que celle des polynucléaires neutrophiles [Barbe, 2001]. Ils ont un diamètre de 12 à 20 μ m (voir Figure 2.7).

Ils ont une fonction phagocytaire et jouent un rôle important dans le premier stade d'une réaction immunitaire. Ils permettent également de nettoyer les tissus des débris cellulaires et des microbes qui s'y trouvent à la suite d'une infection.

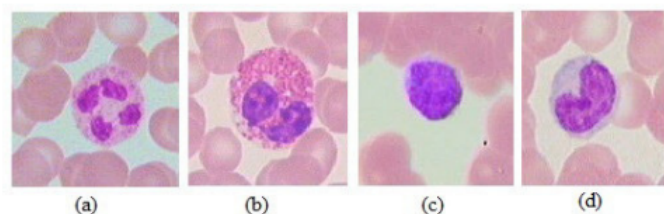


Figure 2.7: Aspect morphologique des polynucléaires et monocytes : (a) neutrophile, (b) éosinophile, (c) basophile, (d) monocyte [Benomar, 2018]

³La chromatine constitue une masse filiforme de matériel génétique contenue dans le noyau des cellules

⁴mécanisme permettant aux cellules de digérer des particules et des micro-organismes

(b) Les lymphocytes

Ce sont des Cellules de petite taille, on en distingue trois sous-types (lymphocytes B, lymphocytes T et les lymphocytes NK (Natural Killer)). Elles sont relativement nombreuses et représentent entre 20 à 40% des leucocytes circulant dans le sang [Web 11]. Elles ont une durée de vie variable, avec un noyau de couleur violette, rond légèrement dentelé et un cytoplasme transparent et réduit. Il en existe de deux tailles et de deux morphologies différentes, les petits et les grands lymphocytes, mais cette différenciation n'a pas d'intérêt pratique car les activités biologiques diversifiées des lymphocytes ne sont pas associées à leur aspect morphologique. Certains lymphocytes sont capables de se transformer en plasmocytes. Ces derniers ont des caractéristiques spécifiques par rapport aux autres cellules. Le plasmocyte est une cellule oblongue, arrondie à une extrémité et pointue à l'autre, avec un diamètre de 15 à 20 μm . Il a un cytoplasme abondant et très bleu (voir Figure 2.8).

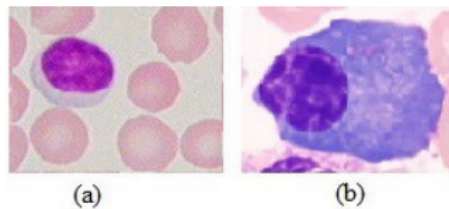


Figure 2.8: Aspect morphologique des lymphocytes : (a) lymphocyte, (b) plasmocyte [Benomar, 2018]

Les lymphocytes jouent un rôle important dans le système immunitaire en protégeant celui-ci contre les menaces des micro-organismes pathogènes (les virus, les bactéries, les champignons ...etc). Ils produisent également des anticorps et les disposent sur leurs membranes.

2.5.3 Les plaquettes (Thrombocytes)

Ce sont des cellules sans noyau (anucléées), avec 2 ou 3 μm de diamètre, arrondies et contenant des fines granulations rosâtres. Elles ont une durée de vie de 8 à 12 jours, avec une densité 200 000 à 300 000 /mm³. Les cellules de la moelle qui donnent naissance aux plaquettes sont géantes et possèdent de multiples noyaux, d'où leur nom de mégacaryocytes. Elles produisent des plaquettes par fragmentation de leur cytoplasme [Barbe, 2001] (voir Figure 2.9).

Les plaquettes ont des fonctions multiples, dont les principales sont en rapport avec différentes étapes de l'hémostase. Elles participent à l'hémostase primaire en libérant des facteurs favorisant la coagulation du sang pour arrêter le saignement des plaies.

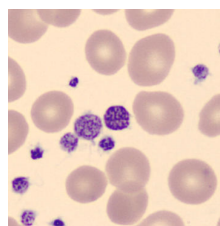


Figure 2.9: Aspect morphologique des plaquettes [Acevedo et al., 2020]

2.6 Conclusion

Les cellules sanguines jouent un rôle important dans la défense du système immunitaire contre les maladies. L'équilibre entre les trois principaux types (globules rouges, globules blancs et plaquettes) assure la vitalité et la résilience du corps. D'où, la compréhension et la maintenance de cet équilibre sont cruciales pour un meilleur système immunitaire. Ainsi, dans le prochain chapitre, nous abordons les concepts généraux de la classification d'images notamment le machine learning, le deep learning, quelques approches de classification et enfin nous parlons de certains travaux liés à la classification d'images cytologiques.

Chapitre 3 : Classification d'images

CLASSIFICATION D'IMAGES

3.1 Introduction

La classification d'images est un domaine ancré au cœur de la vision par ordinateur et de l'intelligence artificielle, permettant de rendre les machines capables d'appréhender et d'interpréter le contenu visuel des images. Elle a beaucoup progressé grâce à la disponibilité croissante de grandes quantités de données et à l'augmentation de la puissance de calcul. Cependant, la croissance significative de ces données a rendu les méthodes traditionnelles fastidieuses, posant un défi à l'analyse humaine en raison des contraintes physiques telles que la concentration, la fatigue et le temps nécessaire. Cela a permis le développement d'approches plus sophistiquées basées sur le Deep Learning (apprentissage profond). Elle est utilisée dans plusieurs domaines, notamment dans la médecine (pour les diagnostics et les recherches), l'industrie (pour l'identification des défauts sur les lignes de production, l'analyse de l'usure des équipements), dans les domaines urbains (pour la reconnaissance des passages piétons, des panneaux de signalisation) et bien d'autres.

3.2 Intelligence Artificielle (IA)

3.2.1 Définitions

«L'automatisation d'activités que nous associons à la pensée humaine, des activités telles que la prise de décision, la résolution de problèmes, l'apprentissage...» [Bellman, 1978]. Elle a pour but de rendre la machine indistinguable de l'être humain et englobe l'apprentissage automatique et l'apprentissage profond (voir Figure 3.1).

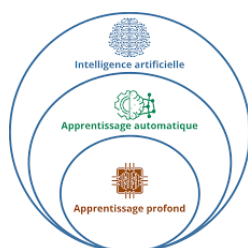


Figure 3.1: Relation entre IA et ML et DL [Web 12]

L'intelligence artificielle a débuté avec la construction des systèmes experts. Ces systèmes experts sont des systèmes très complexes dont la construction nécessitait des milliers d'ingénieurs, qui apprenaient à l'ordinateur comment accomplir des tâches par la programmation rigoureuse de nombreuses règles. Ils sont généralement constitués d'une base de connaissances (contenant des objets, des attributs et des conditions) et d'un moteur d'inférence capable d'utiliser des faits et des règles pour produire de nouveaux faits¹, jusqu'à parvenir à la réponse à la question posée (voir Figure 3.2). Exemple: Une machine qui établit un diagnostic médical à partir d'une liste de symptômes.

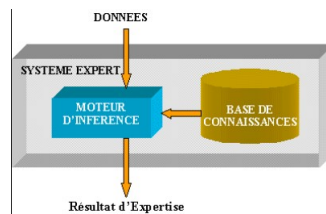


Figure 3.2: Système expert [Web 13]

Malgré la facilité de mise en œuvre, ces systèmes experts se révèlent particulièrement difficiles à appliquer pour des tâches complexes. Par exemple, l'identification d'un chien ou d'une chaise dans un ensemble d'images qui présente une variabilité infinie d'apparences. Résoudre un tel problème à l'aide d'un ensemble de règles prédéfinies est quasiment impossible, d'où l'apparition de l'apprentissage machine, permettant aux systèmes informatiques d'apprendre à partir des données plutôt que de dépendre uniquement d'une base de connaissances explicite.

3.2.2 Apprentissage Automatique (Machine Learning)

3.2.2.1 Définition

Le Machine Learning (ML) est une forme d'intelligence artificielle (IA) axée sur la création de systèmes capables d'apprendre ou d'améliorer leurs performances en fonction des données qu'ils traitent [Mahloul and Mecieb, 2023].

Il implique l'utilisation d'algorithmes qui permettent aux systèmes informatiques d'améliorer leurs performances à partir de l'expérience acquise avec des données. Il englobe des techniques telles que: l'apprendre supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement(voir Figure 3.3).

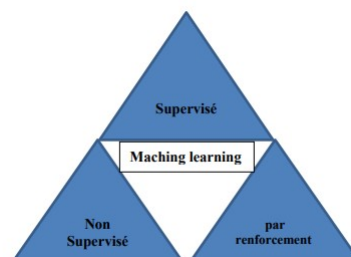


Figure 3.3: Les types d'apprentissage automatique [Taileb, 2022]

¹"faits" fait généralement référence à des déclarations ou propositions considérées comme étant vraies dans un contexte spécifique

3.2.2.2 Apprentissage supervisé

C'est une technique d'apprentissage automatique dans laquelle les modèles sont formés à partir d'un ensemble de données étiquetées où chaque entrée est associée à une sortie connue. Le but principal est que le modèle apprenne la relation entre les entrées et les sorties, de sorte qu'après l'entraînement, il puisse généraliser ses connaissances afin de prédire les sorties correctes pour d'autres nouvelles données. Par exemple, prédire la valeur d'une maison à partir d'entrées telles que le nombre de pièces, l'année de construction, la surface du terrain, l'emplacement ...etc.

Le machine learning supervisé se divise en deux types :

1. Classification: la variable de sortie est une catégorie (exemples d'algorithmes: Machine à Vecteur de Support (SVM), Réseaux de neurones, KNN-le plus proche voisin...etc);
2. Régression: la variable de sortie est une valeur spécifique (régression linéaire, régression logistique...etc).

Régression linéaire

En anglais linear regression (LR), tente de modéliser la relation entre deux variables en ajustant une équation linéaire aux données observées (voir Figure 3.4). Une variable est considérée comme une variable explicative et l'autre est considérée comme une variable dépendante [Yan and Su, 2009]. L'objectif est de créer une équation linéaire qui peut prédire la valeur de la variable dépendante en fonction des valeurs des variables explicatives. L'équation générale d'une régression linéaire simple est: $y = ax + b$, où x est la variable explicative, y la variable dépendante, a la pente de la ligne et b l'ordonnée à l'origine.

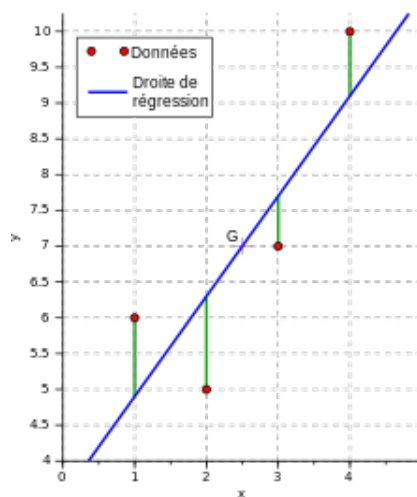


Figure 3.4: Régression linéaire [Web 14]

Régression Logistique

Elle propose de tester un modèle de régression dont la variable dépendante est dichotomique (codée 0-1) et dont les variables indépendantes peuvent être continues ou catégorielles [Web 15] (voir Figure 3.5). Contrairement à la régression linéaire, la sortie de la régression logistique est transformée à l'aide d'une fonction logistique pour prédire des valeurs comprises entre 0 et 1. L'équation logistique pour une régression logistique simple est donné par:

$$P = \frac{1}{1+e^{-(b_0+b_1*x)}} \quad (1)$$

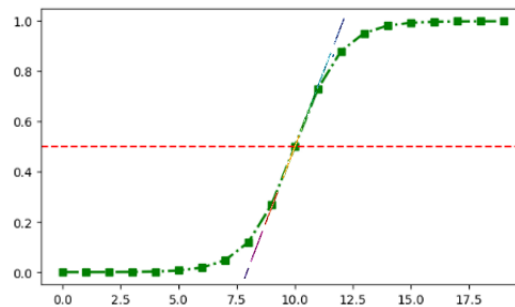


Figure 3.5: Logistique régression [Web 16]

Machine à Vecteur de Support (SVM)

Développés en 1990 [Web 17], les SVM sont des techniques d'apprentissage automatique qui permettent de résoudre des problèmes de classification, de régression ou de détection d'anomalies. Ils sont connus grâce à leurs solides garanties théoriques, leur grande flexibilité ainsi que leur simplicité d'utilisation. Leur principe est de séparer les données en classes à l'aide d'une frontière aussi simple que possible, de telle sorte que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Cette distance est aussi appelée « marge », d'où l'autre nom attribué aux SVM : le séparateur à vaste marge. Le concept de frontière implique que les données soient linéairement séparables (voir Figure 3.6).

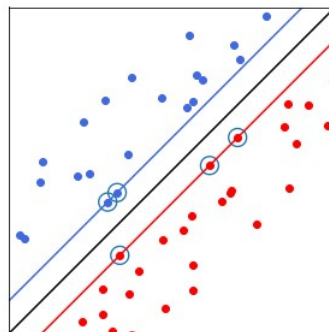


Figure 3.6: SVM [Web 17]

Dans cet espace à deux dimensions, la « frontière » est la droite noire, les « vecteurs de support » sont les points entourés (les plus proches de la frontière) et la « marge » est la distance entre la frontière et les droites bleues et rouges [Web 17].

3.2.2.3 Apprentissage non supervisé

C'est une approche de l'apprentissage automatique où le modèle est exposé à des données non étiquetées et doit identifier une structure, des relations et des schémas au sein de ces données, comme des clusters ou des groupes. Les algorithmes d'apprentissage non supervisé parcourent l'ensemble des données sans aucune indication et essaient de découvrir le partitionnement des données selon leurs paramètres disponibles pour les regrouper en classes homogènes. On peut distinguer deux principales catégories d'apprentissage non supervisé:

1. les algorithmes de clustering: regroupent les données en fonction de similarité (voir Figure 3.7), où les éléments au sein d'un même groupe sont plus similaires les uns aux autres qu'à ceux des autres groupes;
2. Les algorithmes de réduction dimensionnelle: sont utilisés pour extraire ou choisir des caractéristiques dans la base de données. Ces techniques visent à réduire le nombre de variables ou de dimensions dans un ensemble tout en préservant les informations pertinentes.

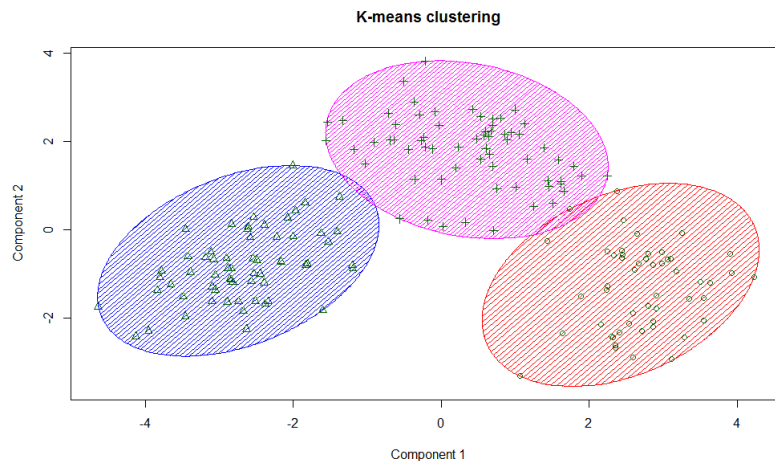


Figure 3.7: Exemple de clustering [Web 18]

3.2.2.4 Apprentissage par renforcement

En anglais Reinforcement Learning (RL), fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre à partir d'expériences successives, ce qu'il convient de faire de façon à trouver la meilleure solution [Web 19]. Dans cette approche, le système interagit avec son environnement afin d'adapter ses paramètres en fonction des réactions reçues de l'environnement, qui fournit ensuite un retour d'information sur les décisions prises.

3.2.3 Apprentissage Profond (Deep Learning)

En ML, l'extraction des caractéristiques et la classification sont faites de manière séparée. Les caractéristiques sont alors transmises à travers les données pour entraîner un modèle qui sera capable de faire des prédictions (voir Figure 3.8).

Avec un nombre important de données, cette extraction de caractéristiques est extrêmement fastidieuse, prend beaucoup de temps et nécessite la connaissance d'un expert. D'où l'intervention de l'apprentissage profond.

Le DL consiste à apprendre automatiquement des caractéristiques (structures) à partir de vastes ensembles de données labellisées (étiquetées). C'est-à-dire la machine est alimentée avec des données brutes et la découverte des attributs est faite de manière automatique (voir Figure 3.8).

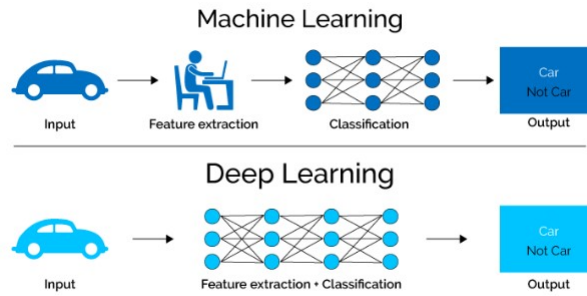


Figure 3.8: Processus du machine learning et du deep learning [Web 20]

Avec le Deep learning, nous parlons d'algorithmes capables de mimer les actions du cerveau humain grâce à des réseaux de neurones artificiels. Les réseaux sont composés de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente [Web 21] (voir Figure 3.9).

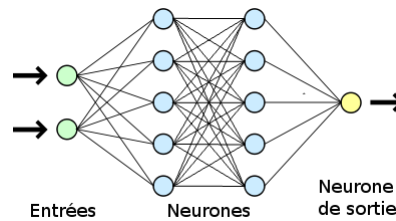


Figure 3.9: Exemple d'un réseau de neurone artificiel [Web 21]

Il existe plusieurs modèles de Deep Learning, parmi lesquels on peut citer:

3.2.3.1 Convolutional Neural Networks (CNN)

Est un modèle de réseau de neurones profond spécifiquement conçu pour traiter les données telles que des images, de l'audio ...etc. C'est un réseau composé de deux parties, la première partie est la partie convolution et la deuxième est la partie classification (voir Figure 3.10). La première partie applique généralement trois types d'opération afin d'extraire les informations pertinentes. Ces opérations sont la convolution, le pooling et la fonction d'activation; la couche de convolution est la couche principale et la plus importante de ce type de réseau, d'où son appellation : réseau de neurones convolutif.

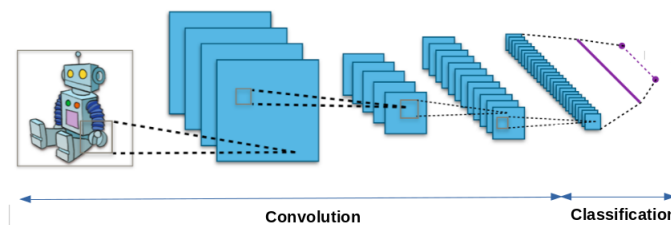


Figure 3.10: Schéma représentant l'architecture d'un CNN [Web 22]

1. Couche de convolution

Est l'élément de base d'un CNN. Son but est de repérer la présence d'un ensemble de caractéristiques (features) dans les images reçues en entrée. Pour cela, on réalise un filtrage par convolution:

on définit une taille de fenêtre qui va se balader à travers toute l'image (voir Figure 3.11). Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image, puis elle va se décaler d'un certain nombre de cases (c'est ce que l'on appelle le pas) vers la droite, et lorsqu'elle arrivera au bout de l'image, elle se décalera d'un pas vers le bas ainsi de suite jusqu'à ce que le filtre est parcourue la totalité de l'image [Web 22].

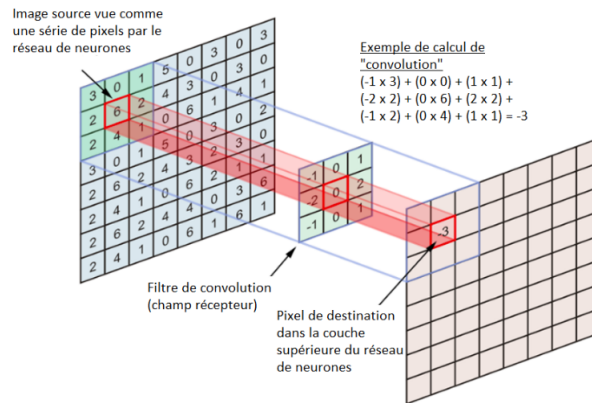


Figure 3.11: Schéma représentant la convolution [Web 23]

2. Couche pooling

Cette couche est généralement placée entre deux couches de convolution. Elle consiste à examiner toutes les valeurs de la fenêtre et à effectuer un calcul statistique simple. Il existe plusieurs techniques telles que le Maxpooling, l'AveragePooling...etc. Le Maxpooling est le plus utilisé; il consiste à prendre la plus grande valeur de la fenêtre et à écarter les autres (voir Figure 3.12). Il est utile lorsqu'on travaille avec de grandes images car il permet de réduire la taille des images tout en préservant leurs caractéristiques pertinentes.



Figure 3.12: Schéma d'exemple de Maxpooling [Web 24]

3. Couches entièrement connectées

Elles représentent la partie de classification où chaque neurone est connecté à tous les neurones de la couche précédente et de la couche suivante, comme le montre la Figure 3.13, d'où le terme fully-connected. Après avoir reçu un vecteur en entrée, la couche fully connected applique successivement une combinaison linéaire, puis une fonction d'activation pour produire la sortie du neurone. Cette sortie devient ensuite l'entrée pour les neurones de la couche suivante.

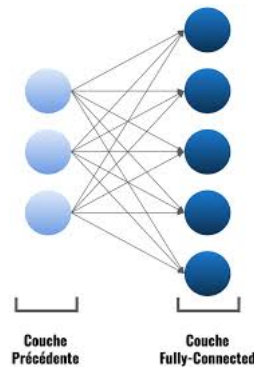


Figure 3.13: Illustration de la couche entièrement connectée [Web 25]

4. Fonction d'activation

Cela permet d'améliorer l'efficacité du traitement en intercalant une couche qui va effectuer une opération mathématique (fonction d'activation) entre les couches de traitement. Il existe plusieurs fonctions d'activation telles que la fonction ReLU, sigmoïde, tanh , softmax ...etc. Mais la fonction ReLU est la plus utilisée dans la pratique; elle est définie comme suit: $\mathbf{F}(\mathbf{x}) = \mathbf{max}(\mathbf{0}, \mathbf{x})$. Cette fonction retourne les valeurs positives. La Figure 3.14 illustre le graphe de la fonction d'activation ReLU.

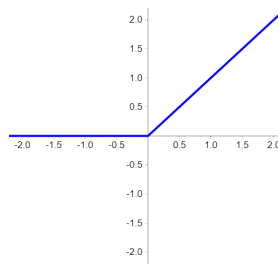


Figure 3.14: Graphe de la fonction d'activation ReLU [Web 26]

3.2.3.2 Les types d'architectures CNN

Il existe différents types d'architectures CNN, qui ont été développées progressivement durant les dernières années, permettant ainsi la réalisation des progrès considérables. Leurs performances sont mesurées grâce au taux d'erreur dans des compétitions telles que le challenge international ImageNet de l'ILSVRC (Large Scale Visual Recognition Challenge). Parmi elles, nous avons : AlexNet, VGG, ResNet ...etc.

- AlexNet [Krizhevsky et al., 2017] : est l'architecture CNN la plus célèbre. Elle a remporté le challenge ImageNet de 2012. Deux techniques de régularisation ont été utilisées par les auteurs afin de réduire le sur-apprentissage, qui sont le dropout (qui consiste à désactiver un certain nombre de neurones aléatoirement à un niveau de l'apprentissage) et le data augmentation (qui consiste à générer des données d'entraînement supplémentaires à partir des exemples existants). Elle est constituée de cinq couches de convolution, trois couches de pooling et trois couches entièrement connectées (voir Figure 3.15).

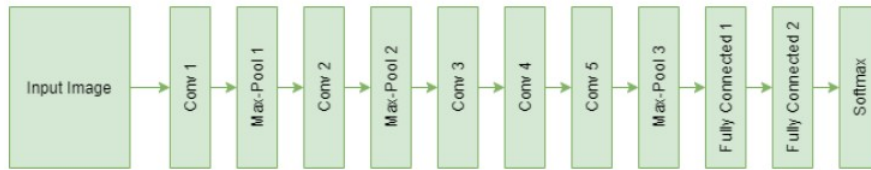


Figure 3.15: Architecture d'AlexNet [Web 27]

- VGG [Simonyan and Zisserman, 2014]: développé dans le laboratoire de recherche Visual Geometry Group (VGG) de l'université d'Oxford, le VGG a été classé comme 2e dans le concours ImageNet en 2014. Il existe deux types: le VGG16 (le modèle contient 16 couches à poids c'est à dire les couches convolutives et les couches entièrement connectées) et VGG19 (le modèle contient 19 couches à poids). Ces couches incluent aussi des couches de pooling (voir Figure 3.16). Toutes les couches de convolution utilisent des filtres de petite taille (3x3), contrairement aux autres réseaux qui utilisent des filtres de tailles différentes pour chaque couche. L'utilisation répétée de ces filtres permet d'apprendre des représentations complexes. Le VGG utilise principalement le Maxpooling.

Après les couches de convolution et de pooling, le VGG a plusieurs couches entièrement connectées pour la classification; elles sont généralement suivies par une couche softmax pour la classification multi-classes. La fonction ReLU est utilisée comme fonction d'activation pour introduire la non-linéarité dans le modèle.

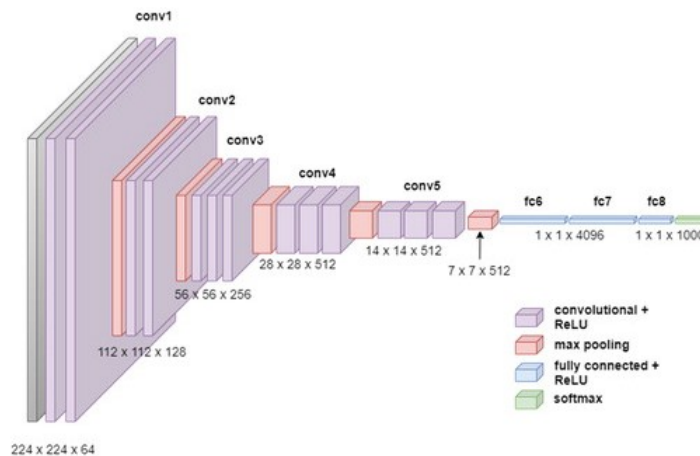


Figure 3.16: Schéma standard de l'architecture du réseau VGG16 [Web 28]

- ResNet [He et al., 2016]: également connu sous le nom de « Deep Residual Networks », est une architecture CNN conçue pour résoudre le problème de classification d'images complexes. Elle a été présentée dans l'article « Deep Residual Learning for Image Recognition » par Kaiming He, Shaoqing Ren, Jian Sun et Xiangyu Zhang en 2015. L'approche consiste à utiliser des blocs résiduels pour faciliter la formation de réseau profond, ce qui permet aux informations de contourner certaines couches plutôt que de traverser chaque couche de manière séquentielle. Cela est réalisé en introduisant des connexions de sauts (Skip connections) ou des connexions résiduelles qui ajoutent l'entrée originale aux sorties des couches intermédiaires. Les blocs de base de ResNet sont les blocs de **conv** et **identity** (voir Figure 3.17). Si l'on

utilise de nombreuses Skip connections, le réseau peut commencer à progresser même si plusieurs couches n'ont pas encore commencé à apprendre.

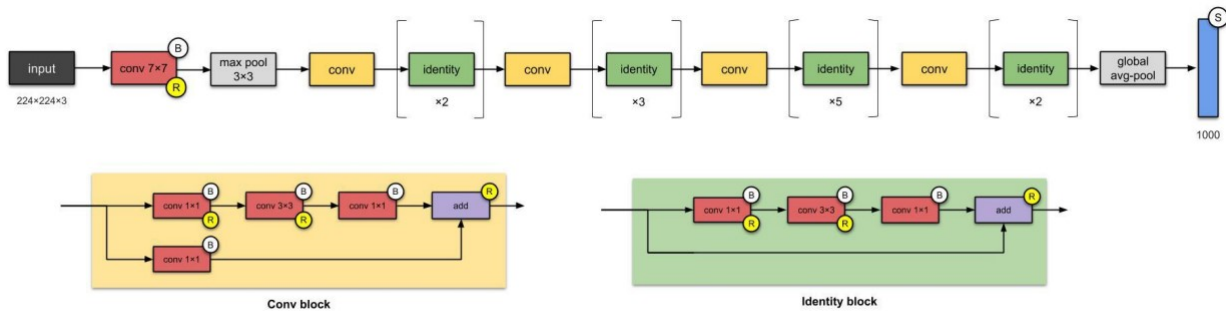


Figure 3.17: Schéma de l'architecture du réseau ResNet50 (B: BatchNormalization, R: Fonction ReLU, S: Softmax) [Web 29]

Le ResNet a atteint des performances exceptionnelles dans diverses tâches de vision par ordinateur, y compris la reconnaissance d'images. L'utilisation de connexions résiduelles permet de résoudre efficacement le problème de la dégradation du signal, facilitant ainsi l'entraînement des réseaux extrêmement profonds.

3.2.3.3 Autres approches du Deep Learning

- Les réseaux de neurones récurrents(RNN): ou Recurrent Neural Networks (RNN) sont des modèles d'apprentissage automatique puissants qui permettent d'analyser des séquences de données, telles que du texte, de la parole ou des séries temporelles. Ces réseaux permettent aux machines de « se souvenir » des informations passées et de les utiliser pour prendre des décisions en temps réel [Web 30]. Les RNN utilisent les sorties précédentes comme entrées supplémentaires; ils sont parfaitement adaptés au traitement de données séquentielles [Web 31] (voir Figure 3.18).

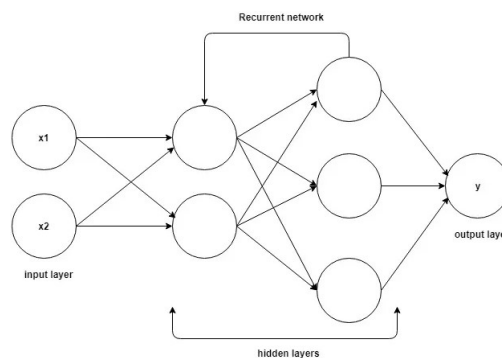


Figure 3.18: Exemple de Modèle RNN [Web 32]

Les réseaux de neurones ont la capacité de prendre en compte des informations contextuelles suite à la récurrence du traitement de la même information. Ils sont utilisés pour résoudre des problèmes liés aux données textuelles, aux données vocales, à la prédiction de classification et à la prédiction de régression.

- Autoencoders (Aes): ce sont des réseaux de neurones un peu particuliers qui possèdent exactement le même nombre de neurones sur leur couche d'entrée et leur couche de sortie. Le but pour

un autoencodeur est d'avoir une sortie la plus proche de l'entrée [Web 33]. Ils sont généralement utilisés pour effectuer des tâches de compression de données, d'apprentissage de représentation et de détection de motifs; c'est un modèle d'apprentissage non supervisé. Les autoencoders produisent en sortie une reconstruction de l'entrée. Un autoencodeur est constitué de deux sous-réseaux : un encodeur et un décodeur (voir Figure 3.19). Lors de l'apprentissage, l'encodeur apprend un ensemble de caractéristiques à partir des données d'entrée, qu'on appelle la représentation latente. Dans le même temps, le décodeur est entraîné à reconstruire les données à partir de ces caractéristiques. L'autoencodeur peut alors être appliqué afin de prédire des entrées qui n'ont pas été précédemment observées. Les autoencoders sont très faciles à généraliser et peuvent être utilisés sur différents types de données, notamment des images, des séries temporelles et du texte [Web 34].

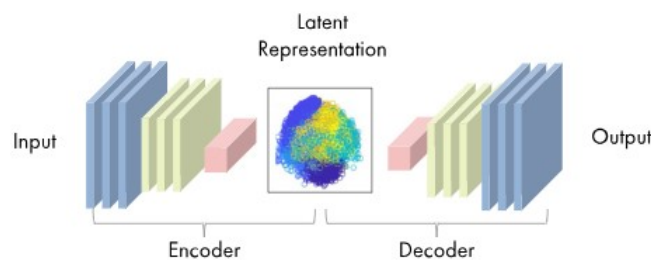


Figure 3.19: Exemple du Modèle Aes [Web 34]

3.2.3.4 Transfert learning

L'apprentissage par transfert désigne l'ensemble des méthodes qui permettent de transférer les connaissances acquises en résolvant des problèmes spécifiques pour résoudre un autre problème [Web 35].

Il repose sur la réutilisation d'un modèle pré-entraîné sur une grande quantité de données pour résoudre une tâche différente. Plutôt que de créer un nouveau modèle à partir de zéro et de l'entraîner sur un ensemble de données de taille limitée, on peut tirer parti des connaissances déjà acquises par le modèle pré-entraîné en l'adaptant à notre tâche spécifique (voir Figure 3.20).

En utilisant des modèles pré-entraînés comme point de départ, l'apprentissage par transfert permet le développement rapide de modèles performants et la résolution efficace de problèmes complexes dans différents domaines en général et en traitement d'images en particulier.

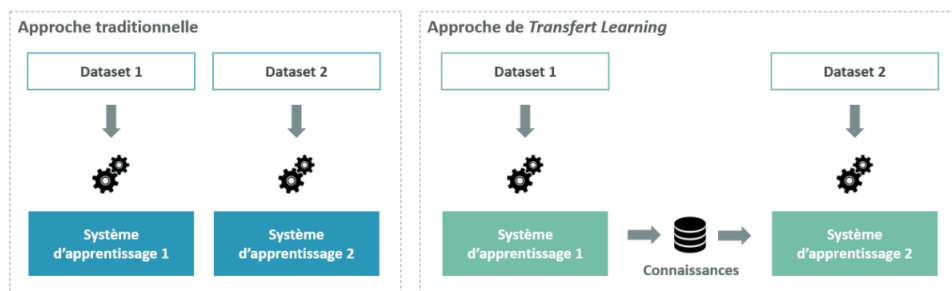


Figure 3.20: Illustration du transfert learning [Web 35]

Dans ce cadre, diverses approches se démarquent selon ce qu'on cherche à transférer, du moment où il faut effectuer le transfert et de la façon de le faire. Feature extraction et fine-tuning sont des

techniques utilisées pour exploiter les connaissances d'un modèle pré-entraîné.

1. Feature extraction : utilise les représentations du réseau pré-entraîné pour extraire des caractéristiques significatives pour une nouvelle tâche. Un nouveau classificateur est ajouté et entraîné à partir de zéro, en réutilisant les caractéristiques apprises. Les paramètres du modèle pré-entraîné restent figés pendant l'entraînement.
2. fine-tuning : lors du fine-tuning, les paramètres du modèle pré-entraîné sont affinés sur le jeu de données de la tâche cible, ainsi que les couches supplémentaires. Les couches supérieures (spécifiques à la tâche) sont généralement ré-entraînées, tandis que les couches inférieures (plus générales) peuvent être fixées ou ajustées légèrement.

Rendre les poids de l'ancien modèle entraînaibles peut aider à spécialiser le nouveau modèle pour le nouveau jeu de données disponible. Cependant, si l'ancien modèle est volumineux, comme le modèle VGG16, le nombre élevé de poids entraînera souvent un sur-ajustement (Overfitting) si le nouvel ensemble de données est trop petit.

3.2.3.5 Le sur-ajustement (Overfitting)

L'overfitting se produit lorsque le modèle devient trop spécialisé sur les données d'entraînement, apprenant non seulement les motifs sous-jacents mais aussi les détails et le bruit spécifiques de ces données, ce qui diminue sa capacité à généraliser à de nouvelles données (voir Figure 3.21). Il se produit également lorsqu'un modèle est formé que sur un groupe en particulier, il peut en résulter des prédictions contradictoires dans les groupes sous-représentés.

Le contraire est le sous-ajustement (underfitting) se produit lorsqu'il y a encore place à l'amélioration des données d'apprentissage.

Lorsque l'erreur de prédiction sur l'ensemble de données d'entraînement et de validation est élevée et que la différence entre les deux est très minime, le modèle est dit sous-ajusté [Kumar, 2022].

Exemple : si le modèle n'est pas assez puissant, s'il est trop régularisé ou s'il n'a tout simplement pas été entraîné assez longtemps. Cela signifie que le réseau n'a pas appris les structures pertinentes dans les données d'apprentissage (training).

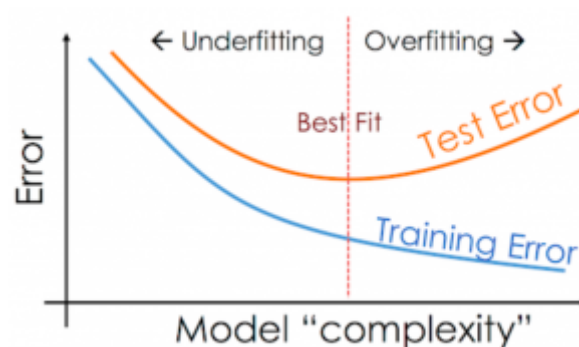


Figure 3.21: Overfitting et Underfitting [Kumar, 2022]

Pour éviter le sur-ajustement, la meilleure solution consiste à utiliser des données d'entraînement plus complètes.

Lorsqu'il existe un petit nombre d'exemples d'apprentissage, le modèle apprend parfois des bruits ou des détails indésirables des exemples d'apprentissage ce qui provoque une dégradation des performances du modèle sur les nouveaux exemples (test et validation). Il existe plusieurs techniques pour lutter contre ce phénomène dans le processus d'apprentissage telles que:

- Augmentation des données (Data Augmentation);
- Utiliser Dropout dans le modèle (désactiver aléatoirement des neurones pendant l'entraînement);
- Ajuster les hyperparamètres (nombre de couche, type de fonction d'activation, optimiseur[SGD, Adam, RMSprop...], nombre d'époque, batch-size, ...);
- KFold Cross Validation;
- Arrêt précoce (EarlyStopping);
- ...etc.

3.3 Classification d'images cytologiques

3.3.1 Définitions

De manière générale, « classification des images consiste à attribuer une étiquette à une image à partir d'un ensemble prédéfini de classes. Concrètement, cela consiste à analyser une image en entrée et à renvoyer une étiquette qui classe cette image. L'étiquette provient toujours d'un ensemble prédéfini de classes possibles [Gafour et al., 2020] ».

La classification d'images cytologiques est un domaine de l'analyse d'images médicales qui a pour objectif d'identifier et de classer les cellules observées au microscope à partir d'images cytologiques. Ces cellules peuvent être issues de divers échantillons biologiques, tels que des frottis de sang, des frottis cervicaux, des échantillons de moelle osseuse ...etc.

Il existe différents types de classification, parmi eux on peut citer:

3.3.2 Types de classifications

3.3.2.1 Classification Binaire (Binary Classification)

Aussi appelée classification binomiale, est une transformation de données qui vise à répartir les membres d'un ensemble dans deux groupes disjoints selon que l'élément possède ou non une propriété/fonctionnalité donnée. Par exemple, un modèle d'apprentissage automatique qui classe les courriers électroniques en tant que « indésirable » ou « légitime » est une classification binaire [Web 36] (voir Figure 3.22) .

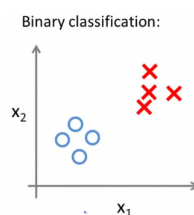


Figure 3.22: Exemple de classification Binaire (x_1, x_2 : attributs)[Web 37]

3.3.2.2 Classification Multi-classes

Est une tâche de classification comportant plus de deux classes et suppose qu'un objet ne peut recevoir qu'une seule classification [Nabi, 2022] (voir Figure 3.23). La classification des visages, des espèces végétales et la reconnaissance de caractères optiques sont des exemples de la classification multi-classes.

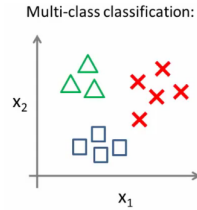


Figure 3.23: Exemple de classification multi-classes (x_1, x_2 : attributs) [Web 37]

3.3.2.3 Classification multi-label

C'est une amélioration de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, permettant ainsi à chaque individu d'appartenir simultanément à une ou plusieurs classes (voir Figure 3.24). Ce type de classification est requis par un grand nombre d'applications actuelles, par exemple, un film appartient souvent à plusieurs genres à la fois.

x	y
x_1	$[t_2, t_5]$
x_2	$[t_1, t_2, t_3, t_4]$
x_3	$[t_3]$
x_3	$[t_2, t_4]$
x_3	$[t_1, t_3, t_4]$

Figure 3.24: Exemple de classification multi-label (X : individus, Y : étiquettes) [Joshi, 2022]

3.3.3 État de l'art

Cette section est dédiée à la présentation de quelques travaux dans le domaine de la classification d'images de cytologie sanguine. Nous avons étudié deux approches, la première basée sur les algorithmes de machine learning (apprentissage automatique) et la deuxième sur les modèles de deep learning (apprentissage profond).

Concernant la première approche, dans la littérature il existe plusieurs travaux et différentes études sur la classification des images cytologiques. Nous avons sélectionné certains papiers pour mener notre étude non exhaustive sur cette approche d'apprentissage automatique.

C'est dans ce cadre que [Benomar et al., 2014] ont proposé une technique de segmentation des images microscopiques dans le but d'extraire les composantes des cellules sanguines (noyau, cyto-

plasme, globule rouge et plasma) (voir Figure 3.25). Cette étape de segmentation fait partie de la procédure d'identification des cellules. La base de données utilisée dans cette étude a été construite à partir d'images réelles (images couleur RVB, 24 bits sous format Bitmap) acquises au sein du service d'hémodiagnostique (CHU de Tlemcen) avec une coloration MGG. Dans un premier temps, un prétraitement est réalisé pour extraire le fond des images (plasma) afin de réduire le temps d'exécution et le bruit. Par la suite, les images couleur ont été converties en images niveaux de gris pour extraire les indices d'Haralick.

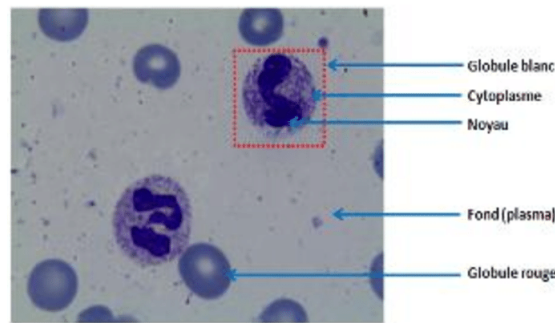


Figure 3.25: Image microscopique de cytologie [Benomar et al., 2014]

Ces attributs, d'Haralick extraits des matrices de cooccurrence, sont utilisés afin de caractériser les textures présentes dans les images niveaux de gris. Ensuite le SVM a été appliqué pour une segmentation par classification pixellaire supervisée.

La méthode proposée a atteint, pour la reconnaissance du noyau une accuracy de 94.11% et une précision de 84.44%; pour le cytoplasme 44.09% et 54.13%; et enfin pour les globules rouges 89.49% et 70.32%.

Pour apporter plus de précision, [Benazzouz et al., 2017] ont présenté une nouvelle méthode robuste et précise pour la segmentation, la caractérisation et la classification automatisées des leucocytes en utilisant une combinaison de méthodes.

Le système s'effectue en deux phases, la première consiste à extraire le noyau du canal vert de l'image grâce à la méthode de seuillage d'Otsu et la seconde opère à travers l'algorithme de croissance de région sur le noyau afin de délimiter le cytoplasme. Après cela, une classification basée sur le SVM est effectuée afin de classer les types de globules blancs et ainsi permettre le comptage de certaines cellules, à savoir les plasmocytes. Une accuracy de 75.25% est atteint à la fin de l'étude.

Une autre approche basée sur la couleur, la texture et la morphologie des globules blancs a été réalisée par [Benomar et al., 2018], afin de préconiser un système automatisé de numération différentielle des globules blancs dans les images de sang périphérique et de moelle osseuse en utilisant la segmentation par caractéristiques multiples et la classification à l'aide de l'algorithme du forêt aléatoire. La méthode proposée se divise en trois étapes (le prétraitement, la segmentation et la classification). Dans la phase de prétraitement, un étirement de décorrélation est utilisé afin d'extraire les caractéristiques chromatiques des cellules, suivi du seuillage d'Otsu et d'une transformation des couleurs pour supprimer l'arrière-plan et la majorité des globules rouges présents sur le plasma. La segmentation a été réalisée à l'aide de deux techniques qui sont : la ligne de partage des eaux (Watershed) contrôlée par un marqueur basée sur la transformation des couleurs et les cartes de distances pour séparer les globules

blancs et un nettoyage de l'image pour différencier ces cellules.

La méthode proposée a été testée sur un ensemble de 87 images couleurs contenant 155 globules blancs obtenues à partir de frottis de moelle osseuse et de sang périphérique colorés par la méthode de coloration MGG. Le modèle a obtenu une accuracy de 95.86% et une reconnaissance robuste des classes majoritaires (lymphocytes, monocytes, neutrophiles et basophiles).

Dans les années récentes, l'utilisation des réseaux de neurones profonds a permis d'atteindre des performances élevées dans la classification automatique des cellules sanguines, ce qui nous a amenés à étudier dans notre deuxième approche les modèles de deep learning. [Acevedo et al., 2019] ont proposé un système de classification automatique de huit (8) groupes de cellules sanguines périphériques avec une grande précision en utilisant comme approche l'apprentissage par transfert avec des réseaux de neurones convolutifs. Les auteurs ont proposé deux conceptions basées sur deux architectures CNN VGG16 et InceptionV3, déjà pré-entraînées sur l'ensemble de données ImageNet.

Dans le premier cas, les réseaux ont été utilisés comme extracteurs de caractéristiques et ces caractéristiques ont été utilisées pour entraîner un classificateur SVM. Dans le second cas, les mêmes réseaux ont été affinés sur l'ensemble de données afin d'obtenir deux modèles de bout en bout.

Le dataset utilisé est le PBC (Peripheral Blood Cell) composé de 17 092 images numériques de cellules sanguines périphériques normales obtenues à partir de frottis sanguins et colorées avec MGG. Dans le premier cas, les accuracy des tests expérimentaux obtenues étaient de 86% et 90% lors de l'extraction de caractéristiques avec VGG16 et InceptionV3 respectivement. D'autre part, dans l'expérience de réglage fin, des valeurs d'accuracy de 96% et 95% ont été obtenues en utilisant respectivement VGG16 et InceptionV3.

En faisant référence à d'autres techniques basées sur l'architecture CNN, [Togaçar et al., 2020] ont proposé une méthode composée de trois étapes : dans un premier temps, les modèles CNN AlexNet, GoogLeNet et ResNet50 sont utilisés comme extracteurs de fonctionnalités. Ensuite, les caractéristiques obtenues à partir des couches de ces modèles CNN ont été combinées. Dans un deuxième temps, les méthodes de sélection de caractéristiques MIC et Ridge Regression sont utilisées. Enfin, dans la troisième étape, les caractéristiques sélectionnées par les méthodes MIC et Ridge Regression ont été combinées et classées par la méthode QDA (Analyse Discriminante Quadratique). L'ensemble de données utilisé se compose de quatre catégories : éosinophiles, lymphocytes, monocytes et neutrophiles; avec un total de 12 364 images. Le taux d'accuracy obtenu était de 97.95% dans la classification des globules blancs.

Dans le même contexte, [Çınar and Tuncer, 2021] ont proposé un modèle hybride CNN basé sur l'apprentissage profond pour la classification des globules blancs. Le modèle présenté utilise les architectures CNN AlexNet et GoogLeNet. Les vecteurs de caractéristiques de la dernière couche de regroupement des deux architectures CNN ont été fusionnés, et le vecteur de caractéristiques résultant a été classé par SVM. Deux ensembles de données différents, provenant du site web Kaggle et de la base de données LISC sont classés. L'ensemble de données du site web Kaggle contient 12 500 images de cellules sanguines. Les types de cellules qu'il contient sont: les éosinophiles, les lymphocytes, les monocytes et les neutrophiles. Dans l'ensemble de données LISC, il y'a au total 189 images de globules blancs d'une taille de 720×576 pixels.

Avant l'entraînement, les images ont été augmentées en les faisant une rotation de 90, 180 et 270 degrés. L'accuracy et le F1_score obtenus étaient de 99.73%, 99% et 98.23%, 98% respectivement pour les deux ensembles de données.

En utilisant le dataset PBC, [Chola et al., 2022] ont proposé un modèle basé sur le transfert learning, nommé BCNet. Le CNN de base utilisé pour l'extraction des caractéristiques est l'EfficientNet-B0, pré-entraîné sur l'ensemble de données ImageNet; qui par la suite, a été modifié par l'ajout de différentes couches pour concaténer et déterminer les caractéristiques profondes extraites, puis les compresser avec Global Average Pooling. Afin de classifier les huit (08) types de cellules sanguines périphériques, deux couches entièrement connectées ayant respectivement 1024 et 8 neurones ont été utilisées. Pour éviter le problème de sur-ajustement (overfitting) et améliorer la capacité de généralisation du modèle, les auteurs ont utilisé une couche de dropout de 0.5 % sur toutes les couches de convolution et denses. Trois optimiseurs différents, à savoir ADAM, RMSprop et SGDM ont été utilisés pour mener l'expérience. Les meilleures performances d'évaluation ont été obtenues par l'optimiseur RMSprop, avec une accuracy de 98.51% et un F1_score de 96.24%. ADAM et SGD ont tous les deux obtenu respectivement une accuracy de 98.47%, 98.24% et des F1_score de 96.03%, 95.30%.

Pour identifier diverses cellules sanguines dans des images microscopiques du sang, [Alkafrawi and Dakhell, 2022] ont proposé un modèle de classification basé sur AlexNet. En utilisant des réseaux neuronaux de convolution, les tests ont été réalisés sur un ensemble de données de 17 000 échantillons de frottis sanguins obtenus de l'Hôpital Clinic de Barcelone. Le modèle AlexNet est composé de cinq couches de convolution, trois couches de MaxPooling et trois couches entièrement connectées. Le modèle basé sur AlexNet a présenté un taux d'erreur minimale de 0.0049 et une accuracy de 95.08%.

[Cheuque et al., 2022] ont présenté un modèle CNN à plusieurs niveaux dans le but de résoudre le problème de classification des globules blancs. Un schéma multi-niveau et hybride a été proposé, dans lequel la première étape correspond à l'utilisation d'un réseau Faster R-CNN appliqué pour identifier la région d'intérêt des globules blancs, ainsi que pour la séparation des cellules Mononucléaires des cellules Polymorphonucléaires. En deuxième étape, deux CNN ont été développés : un pour classer les cellules mononucléaires en lymphocytes et monocytes et un autre pour séparer les cellules polymorphonucléaires en neutrophiles et éosinophiles. Les CNN implémentés ont l'architecture MobileNet comme modèle de base. Cinq (5) ensembles de données différents ont été utilisés, dont la description suit:

- L'ensemble de données de détection de cellules sanguines (BCD);
- L'ensemble de données de numération globulaire complète (CBC);
- L'ensemble de données sur les globules blancs (WBC);
- L'ensemble de données Kaggle Blood Cell Images (KBC);
- L'ensemble de données Leucocyte Images for Segmentation and Classification (LISC).

Le modèle a obtenu de bons indicateurs de performance, avec une accuracy, une précision, un recall et un F1_score de 98.4%.

Dans un autre travail sur les images cytologiques, [Zhu et al., 2023] ont conçu un réseau basé sur l'architecture CNN ResNet50 pré-entraînée sur l'ensemble de données ImageNet. Après certaines modifications, cette architecture a été utilisée comme étant un extracteur de caractéristiques, puis ces caractéristiques extraites sont introduites dans l'ETRN, un modèle proposé qui est une combinaison de trois RNN, afin d'améliorer la performance de la classification multi-classes des cellules sanguines. Les auteurs ont utilisé un ensemble de données publiques sur les cellules sanguines qui comprend 12 500 images classées en quatre types différents : les neutrophiles, les éosinophiles, les monocytes et les lymphocytes.

Pour contrer le problème de données limitées et d'overfitting, les auteurs ont proposé un nouveau réseau contradictoire génératif pour les cellules sanguines (BCGAN).

Le BCGAN proposé génère des images synthétiques de cellules sanguines lorsque le discriminateur est incapable de trouver les différences entre les images générées et les images réelles. Les performances moyennes obtenues par le modèle étaient de 95.05%, 93.25%, 97.75%, 93.72% et 95.38% respectivement pour l'accuracy, la sensibilité, la précision, la spécificité et le F1_score.

[Sharma et al., 2023] ont construit un modèle basé sur l'apprentissage par transfert utilisant l'architecture DenseNet121 pour classer les cellules sanguines en quatre classes différentes. Le modèle est simulé avec quatre tailles de lots (batches), à savoir 8, 16, 32, 64 avec l'optimiseur Adam et pendant 10 époques. L'ensemble de données a été extrait de Kaggle et contient 12 444 images, réparties en quatre (4) catégories d'images: éosinophiles, lymphocytes, monocytes et neutrophiles. Ces images ont été redimensionnées de 320×240 à 224×224 . Des techniques de normalisation et d'augmentation telles que le retournement, la rotation, la luminosité et le zoom ont été appliquées sur l'ensemble de données. L'apprentissage avec un batch-size de 8 donne des meilleurs résultats par rapport aux autres. Le modèle proposé a atteint une accuracy de 98.84%, une précision de 99.33%, une sensibilité de 98.85% et une spécificité de 99.61%.

Afin d'améliorer la performance de classification, [Lu et al., 2024] ont proposé une méthode basée sur l'auto-encodeur masqué (MAE) et l'apprentissage actif (AL). Cette méthode vise à étiqueter de manière sélective un sous-ensemble d'images de haute qualité, en se concentrant sur la classification multi-classes des cellules sanguines, crucial pour diagnostiquer les maladies du sang. En cas de limitations budgétaires ou de ressources d'étiquetage, l'approche automatise la sélection d'échantillons en classant les pertes auto-supervisées, utilisant l'échantillonnage par lots en fonction de l'incertitude de l'échantillon. Le processus implique l'utilisation d'une grande quantité de données non étiquetées pour les tâches de masquage et de restauration via l'algorithme auto-supervisé MAE. Ensuite, le réseau est affiné avec un petit nombre d'échantillons étiquetés pour améliorer les performances de classification. Enfin, une méthode d'apprentissage actif sélectionne les échantillons les plus précieux pour l'étiquetage, en combinant la perte auto-supervisée et l'entropie de l'information de prédiction. Cette approche améliore considérablement les performances du modèle lorsque les échantillons étiquetés sont rares. Le taux d'accuracy obtenu était de 96.36%.

Dans un autre travail, [Tarimo et al., 2024] ont réalisé un modèle bidirectionnel en deux (2) étapes combinant les méthodes You Only Look Once (YOLO: détection rapide d'objets) et Vision Transformer (ViT: puissantes capacités de représentation d'images). La méthode implique la détection d'objets et

le recadrage pour identifier les régions de White Blood Cells (WBC), produisant deux ensembles de données distincts de cellules et de leur noyau. Ensuite, ils ont procédé à l'extraction et à la concaténation de caractéristiques des ensembles de données fusionnés de WBC et de régions nucléaires avant de les classer à l'aide du modèle ViT pré-entraîné.

L'ensemble de données utilisé est composé de 4 445 images de cellules sanguines réparties sur 16 classes. Les images originales étaient de taille 360x366, mais avant l'entraînement, elles ont été redimensionnées en une taille de 224x224. Le modèle a atteint une accuracy de 96.449% sur les 16 classes.

Dans cette sous section nous avons présenté des travaux de la littérature et qui sont regroupés en deux catégories; la première basée sur l'apprentissage automatique et la deuxième sur le Deep Learning. Le tableau 3.1 représente le récapitulatif de l'ensemble des méthodes résumées ci-dessus.

Auteurs	Dataset	Théories/ Méthodologies	Accuracy
[Benomar et al., 2014]	Images réelles (27 images de cellules sanguines) acquises au sein du service d'hémobiologie (CHU de Tlemcen)	Extraction de caractéristiques avec attributs d'Haralick et SVM comme classifieur	Noyau:94.11% Cytoplasme: 44.09% Globules rouges:89.49%
[Benazzouz et al., 2017]	Images réelles (194 images de cellules sanguines) acquises au sein du service d'hémobiologie (CHU de Tlemcen)	Extraction des noyaux avec le seuillage d'otsu, l'algorithme de croissance de région pour délimiter le cytoplasme et le SVM pour classer les cellules	75.25%
[Benomar et al., 2018]	Ensemble de données de test composés de 87 images contenant 155 globules blancs	La segmentation par caractéristiques multiples et la classification à l'aide de l'algorithme du forêt aléatoire	95.86%
[Acevedo et al., 2019]	Peripheral Blood Cell (PBC) composé de 17 092 images de cellules sanguines périphériques	Méthode1: VGG16 et InceptionV3 comme extracteur et SVM comme classifieur Méthode2: VGG16 et InceptionV3 sont entraînés avec l'ensemble de cellules sanguines	Méthode1: VGG16:86% InceptionV3:90% Méthode2: VGG16:96% InceptionV3:95%
[Togaçar et al., 2020]	Ensemble de données de Kaggle (12 364 images de cellules sanguines)	AlexNet, GoogLeNet et ResNet50 comme extracteur, MIC et Ridge Regression pour combiner les caractéristiques et QDA comme classifieur	97.95%
[Çınar and Tuncer, 2021]	Ensemble de données de Kaggle (12 500 images de cellules); Ensemble de données Leucocyte Images for Segmentation and Classification LISC (189 images de globules blancs)	AlexNet et GoogleNet comme extracteur et le SVM comme classifieur	Ensemble données1:99.73% Ensemble données2:98.23%

Auteurs	Dataset	Théories/ Méthodologies	Accuracy
[Chola et al., 2022]	Peripheral Blood Cell (PBC) composé de 17 092 images de cellules sanguines périphériques	EfficientNet-B0 est utilisé comme modèle de base pour l'extraction et 2 couches denses pour la classification	98.51%
[Alkafrawi and Dakhell, 2022]	Peripheral Blood Cell (PBC) composé de 17 092 images de cellules sanguines périphériques	AlexNet est utilisé comme modèle de base	95.08%
[Cheuque et al., 2022]	Ensemble de données de détection de cellules sanguines(BCD):2340 images de cellules; Ensemble de données de numération globulaire complète(CBC):360 images de frottis sanguins; Ensemble de données sur les globules blancs (WBC):400 images de globules blancs; Ensemble de données de Kaggle Blood Cell Images (KBC):12 500 image de cellules sanguines; Ensemble de données Leucocyte Images for Segmentation and Classification (LISC): 250 images de frottis sanguins	Sépare les Mononucléaires des polymorphonucléaires avec Faster R-CNN et MobileNet comme classifieur	98.4%
[Zhu et al., 2023]	Ensemble de données de Kaggle (12 500 images de cellules sanguines)	ResNet50 comme extracteur et une combinaison de trois RNN comme classifieur	95.05%
[Sharma et al., 2023]	Ensemble de données de Kaggle (12 444 images de cellules sanguines)	Utilise DenseNet121 comme modèle de base	98.84%
[Tarimo et al., 2024]	L'ensemble de données est composé de 4 445 images de cellules sanguines acquises à l'hôpital universitaire de Soonchunhyang Bucheon en Corée du Sud	Identifie le noyau des cellules avec YOLO et ViT comme classifieur	96.449%

Table 3.1: Récapitulatif de certaines études proposées dans la littérature pour la classification des images cytologiques

3.4 Conclusion

la classification d'images joue un rôle crucial dans de nombreux domaines, de la reconnaissance faciale à la médecine. Propulsée par l'évolution de l'intelligence artificielle et l'avancée continue de l'apprentissage automatique, la classification d'images offre des solutions de plus en plus sophistiquées pour l'analyse et l'interprétation visuelle de la réalité. Dans le dernier chapitre, nous allons étaler la partie résultats et discussion en présentant les modèles proposés, les langages utilisés et enfin faire une analyse des résultats.

Chapitre 4 : Implémentation

IMPLÉMENTATION

4.1 Introduction

L'analyse de la morphologie cellulaire constitue un pilier fondamental dans le diagnostic de diverses maladies hématologiques. Cette analyse est particulièrement centrée sur les leucocytes ou globules blancs, qui sont essentiels pour le maintien de la santé humaine et servent d'indicateurs clés dans le diagnostic de maladies graves comme le paludisme, la leucémie, le VIH/sida, le myélome, les troubles de coagulation et d'autres infections virales. Cependant, la distinction entre les différents types de cellules sanguines périphériques normales ou anormales représente une tâche ardue nécessitant expérience et compétences. Ainsi, notre étude propose un système de classification automatique de cellules sanguines. Ce système peut identifier huit (8) types différents de cellules sanguines périphériques avec une haute précision. L'approche est basée sur le transfert learning en utilisant les CNN.

4.2 Outils utilisés

Pour la réalisation de notre système, nous avons utilisé Google Colab intégré à Google Drive, ce qui permet de stocker, partager et accéder facilement aux notebooks.

4.2.1 Langage utilisé (Python)

Python est le langage de programmation informatique le plus populaire et le plus utilisé, notamment dans le domaine de la data science et du machine learning. De plus, python est un langage open source, multiplateforme et orienté objet, qui fonctionne sur divers systèmes d'exploitation tels que windows, macOS et linux, ce qui en fait un choix idéal pour les développeurs travaillant sur différents environnements. Il dispose de structures de données de haut niveau interprété (pas d'étape de compilation). Contrairement à HTML, CSS ou JavaScript, son usage n'est donc pas limité qu'au développement web. Il peut être utilisé pour tout type de programmation et de développement logiciel.



Figure 4.1: Logo de python [Web 38]

4.2.2 Google Colab (Colaboratory)

Google Colab est une plateforme de cloud fournie par Google, basée sur Jupiter Notebook. Elle offre un environnement de développement complet sur python, avec de nombreuses bibliothèques populaires pré-installées telles que TensorFlow, PyTorch ...etc. De plus, elle permet d'accéder gratuitement à des ressources de calcul accélérées, notamment des GPU et des TPU, ce qui la rend particulièrement adaptée pour des tâches liées à l'apprentissage automatique et à la science des données. La Figure 4.2 représente l'interface de Google Colab.

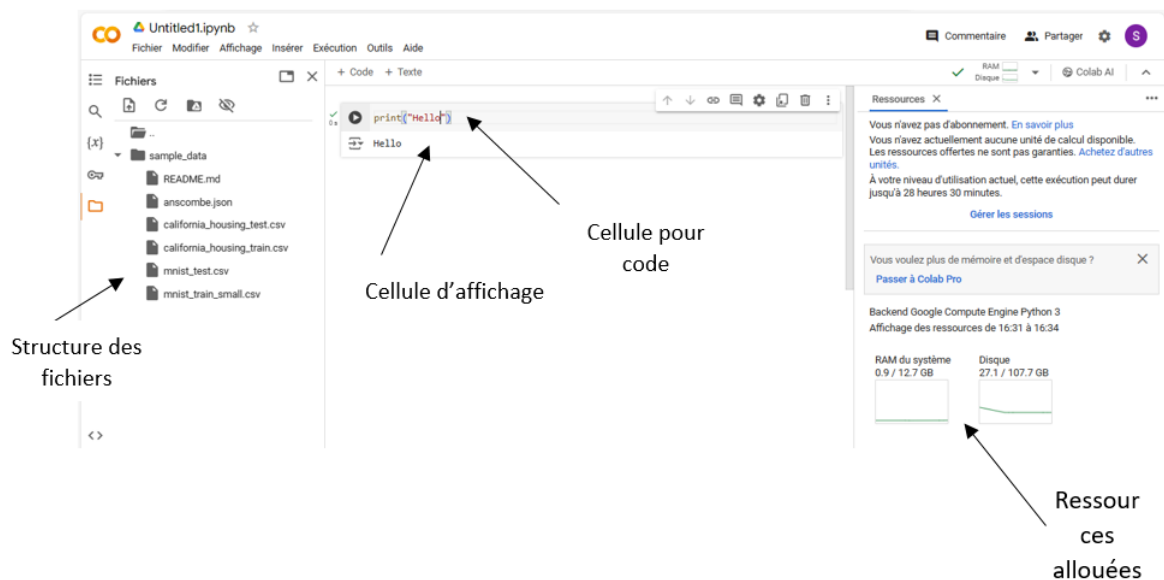


Figure 4.2: Interface de Google Colab

4.2.3 Les bibliothèques

- **TensorFlow**

Créé par l'équipe Google Brain en 2011, TensorFlow est un framework de programmation pour le calcul numérique, renommé et rendu open source par Google en 2015. Son nom s'inspire notamment du fait que les opérations courantes des réseaux de neurones s'effectuent principalement à l'aide d'une table multidimensionnelle de données, appelée tenseurs (Tensor) qui est l'équivalent d'une matrice.

Pour faire simple, TensorFlow est une bibliothèque de machine learning, il s'agit d'une boîte à outils permettant de résoudre des problèmes mathématiques extrêmement complexes avec aisance. Elle permet aux chercheurs de développer des architectures d'apprentissage expérimentales et de les transformer en logiciels [Web 39].



Figure 4.3: Logo de Tensorflow [Web 40]

- **Keras**

Est une API (Application Programming Interface) de réseau de neurones écrite en langage python. Il s'agit d'une bibliothèque open source, exécutée par-dessus des frameworks tels que Theano et TensorFlow. Elle a été développée dans le cadre d'expérimentations et de prototypages rapides grâce à sa convivialité, sa modularité et sa facilité d'extensibilité, ce qui l'a rendue populaire auprès des chercheurs qui souhaitent construire et expérimenter différents modèles de réseaux neuronaux.



Figure 4.4: Logo de Keras [Web 41]

- **Numerical Python (Numpy)**

Est une bibliothèque open source liée au langage python. Elle met à disposition des utilisateurs des tableaux multidimensionnels (ndarray), ainsi qu'une grande multitude de fonctions afin d'effectuer des opérations mathématiques et statistiques. Numpy est principalement utilisé dans des domaines où des calculs numériques efficaces sont nécessaires tels que la science des données, l'apprentissage automatique, l'ingénierie ...etc.



Figure 4.5: Logo de Numpy [Web 42]

- **Scikit-learn**

Également appelée Sklearn, est une bibliothèque open source d'apprentissage automatique permettant aux utilisateurs d'effectuer différentes tâches notamment la régression, la classification, la sélection de modèles, le pré-traitement, la réduction de la dimensionnalité et le regroupement. Elle est basée sur les bibliothèques NumPy, SciPy et Matplotlib, qui sont d'autres bibliothèques python largement utilisées et cruciales pour l'analyse et la visualisation des données.



Figure 4.6: Logo de Scikit-learn [Web 43]

- **Matplotlib**

Est une bibliothèque de Data Visualization en python, inspirée de Matlab et développée par John Hunter dans les années 2000. Le neurobiologiste cherchait un moyen de visualiser les signaux électriques du cerveau de patients épileptiques [Web 44]. Elle offre une grande flexibilité et permet de créer une variété de graphiques tels que des diagrammes en barres, des graphiques linéaires, des diagrammes circulaires, des nuages de points, etc [Web 45].



Figure 4.7: Logo de Matplotlib [Web 46]

- **Gradio**

Est une bibliothèque Python qui vous permet de créer rapidement des composants d'interface utilisateur personnalisables autour de vos modèles d'apprentissage automatique, de vos modèles d'apprentissage profond et d'autres fonctions. Il vous permet de créer une interface utilisateur simple et partageable pour votre modèle en seulement quelques lignes de code [Web 47].

Il est rapide, simple à mettre en place et prêt à être utilisé et partageable en tant que connexion publique à laquelle tout le monde peut accéder pour exécuter le modèle à distance et en parallèle dans votre machine. Gradio fonctionne avec un large éventail de médias : texte, images, vidéo et son. Mis à part les modèles ML, il peut très bien être utilisé comme on peut s'y attendre pour les plongements de code python [Web 48].



Figure 4.8: Logo de Gradio [Web 48]

4.3 Dataset utilisé

L'ensemble de données que nous allons utiliser pour réaliser notre classification est un dataset disponible en ligne, réalisé par [Acevedo et al., 2020], connu sous le nom de PBC (Peripheral Blood Cell) et publié par le journal ELSEVIER pour des travaux d'expérimentation et de recherche. Il est composé de 17 092 images de cellules normales individuelles, qui ont été acquises à l'aide de l'analyseur CellaVision DM96 dans le laboratoire central de la clinique hospitalière de Barcelone [Acevedo et al., 2020]. L'ensemble de données est organisé dans les huit (8) groupes suivants : neutrophiles, éosinophiles, basophiles, lymphocytes, monocytes, granulocytes immatures (promyélocytes, myélocytes et métamyélocytes), érythroblastes et plaquettes ou thrombocytes (voir Figure 4.9). Les images ont été capturées chez

des personnes sans infection, maladie hématologique ou oncologique et exemptes de tout traitement pharmacologique au moment de la prise de sang. Ce jeu de données est le premier ensemble accessible au public avec un grand nombre de cellules sanguines périphériques normales.

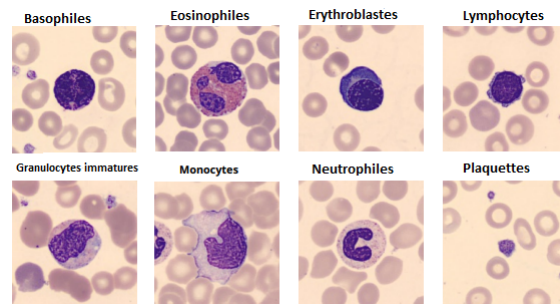


Figure 4.9: Exemple des huit (8) types de cellules normales du sang périphérique [Acevedo et al., 2020]

L'acquisition du jeu de données :

Des images numériques de cellules sanguines périphériques normales ont été obtenues à partir d'échantillons prélevés dans le laboratoire central de la clinique hospitalière de Barcelone. Afin d'obtenir toutes les numérations globulaires, des échantillons de sang ont été analysés dans l'instrument Advia 2120. Ensuite, des frottis ont été préparés automatiquement à l'aide de la machine à colorer les lames Sysmex SP1000i, avec coloration MGG. Enfin, l'analyseur automatique CellaVision DM96 a été utilisé pour obtenir des images de cellules individuelles au format jpg et d'une taille de 360×363 pixels (voir Figure 4.10). Les images obtenues ont été étiquetées et stockées par les pathologistes cliniques [Acevedo et al., 2020].

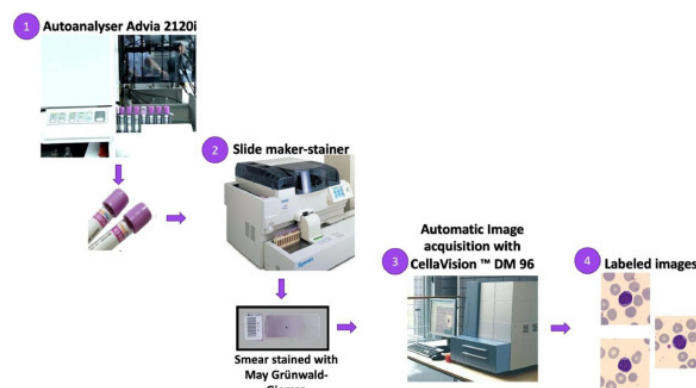


Figure 4.10: Étapes d'acquisition du dataset (1: Traitement des échantillons, 2: Coloration au MGG, 3: Acquisition automatique de l'image, 4: Labelisation) [Acevedo et al., 2020]

4.4 Méthodes proposées

Dans le cadre de notre projet, nous avons élaboré trois (3) modèles CNN avec VGG16 [Simonyan and Zisserman, 2015], InceptionV3 [Szegedy et al., 2015] et ResNet50 [He et al., 2016] en utilisant la méthode d'apprentissage par transfert, qui est une technique précieuse en apprentissage automatique, permettant de réutiliser des modèles existants, initialement formés pour une tâche spécifique, afin de résoudre un problème différent. Cette approche présente de nombreux avantages en économisant le temps et les ressources nécessaires à la construction de nouveaux modèles neuronaux à partir de zéro. Elle est particulière-

ment utile dans les domaines de la vision par ordinateur et du traitement du langage naturel, où elle contribue à améliorer la précision et les performances des modèles résultants.

- Les modèles CNN utilisés pour construire nos trois modèles ont été pré-entraînés sur la base de données ImageNet [Deng et al., 2009], qui est un projet de recherche visant à constituer une vaste base d'images annotées (des images et leurs labels) (voir Figure 4.11);
- Dans le but de diminuer le temps de calcul et de lancer l'apprentissage des différents modèles sur notre dataset (environ 17 092 images de dimension 360x363 pixels), nous avons opté pour une réduction de la dimension des images originales avec une dimension d'entrée de 128x128 pixels pour les trois modèles au lieu de 224x224 pixels pour VGG16 et ResNet50 et 299x299 pixels pour InceptionV3;
- Ensuite, pour chaque modèle, nous avons retiré la partie supérieure initialement dédiée à la classification afin de la reconstruire manuellement et de l'adapter spécifiquement à nos données (`include_top = False`), comme le montre la Figure 4.11;
- De plus, nous avons activé l'apprentissage des poids pour toutes les couches (layers) de convolution de chaque modèle, permettant ainsi aux modèles d'apprendre les caractéristiques spécifiques de nos images;
- Enfin, nous avons appliqué `GlobalAveragePooling2D` aux outputs (sorties) de chaque modèle dans le but de réduire le nombre de paramètres, de compresser les données et de produire une sortie de dimension fixe qui peut être directement utilisée comme entrée pour une couche de classification.

```

base_model= keras.applications.VGG16(
    weights= 'imagenet',
    input_tensor= None,
    input_shape=(128,128,3),
    include_top= False
)
base_model.trainable= True

[ ] base_model_2= keras.applications.ResNet50(
    weights= 'imagenet',
    include_top= False,
    input_tensor= None,
    input_shape= (128,128,3)
)
base_model_2.trainable=True

[ ] base_model_3= keras.applications.InceptionV3(
    weights= 'imagenet',
    include_top= False,
    input_tensor= None,
    input_shape= (128,128,3)
)
base_model_3.trainable=True

```

Figure 4.11: Importation des modèles

4.4.1 Modèle 1 (VGG16 et InceptionV3)

Le premier modèle est une concaténation du modèle VGG16 (que nous avons déjà présenté dans le chapitre 3) et InceptionV3, comme le montre la Figure 4.13.

InceptionV3: est une amélioration du modèle Inception, qui est originellement connu sous le nom de GoogleNet. Développée par les chercheurs de Google, cette version a été introduite pour augmenter la précision tout en réduisant la complexité computationnelle du modèle original. Elle a introduit des factorisations de convolutions, décomposant les convolutions 5x5 en deux convolutions 3x3 successives pour réduire encore plus le nombre de paramètres. Le modèle est formé sur plus d'un million d'images de la base de données ImageNet, avec une profondeur de 42 couches et une taille d'entrée d'image de 299x299 pixels (voir Figure 4.12).

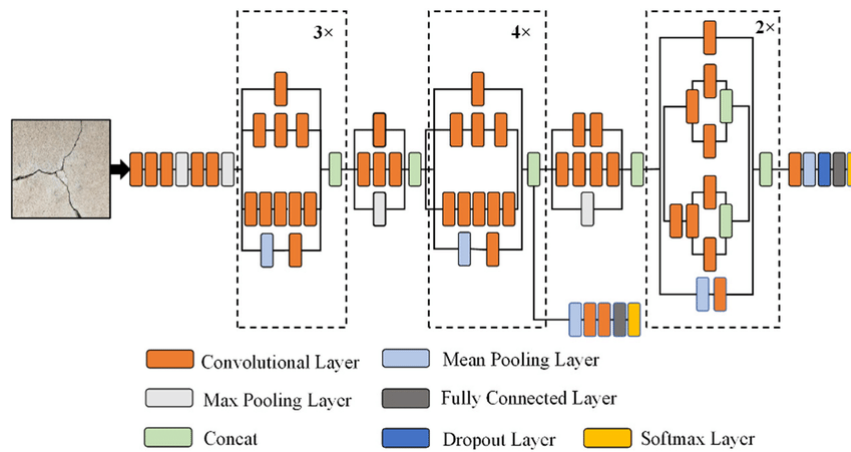


Figure 4.12: Architecture du modèle InceptionV3 [Web 50]

- Comme structure de la partie classification du Modèle 1, il y'a une seule couche dense (entièrement connectée) ayant 256 neurones avec comme fonction d'activation "ReLU", suivie d'une couche de BatchNormalization qui permet d'améliorer la stabilité, la vitesse, les performances du processus d'entraînement des réseaux de neurones et réduit également le risque du sur-apprentissage (overfitting); puis d'une couche dropout de 50%, qui désactive aléatoirement 50% des neurones pendant l'entraînement, ce qui entraîne une meilleure généralisation du modèle et une réduction du sur-apprentissage;
- Enfin, la couche de sortie qui a comme nombre de neurones, le nombre de classes de notre dataset (8) avec une fonction d'activation softmax, qui donne la probabilité d'appartenance aux différentes classes.

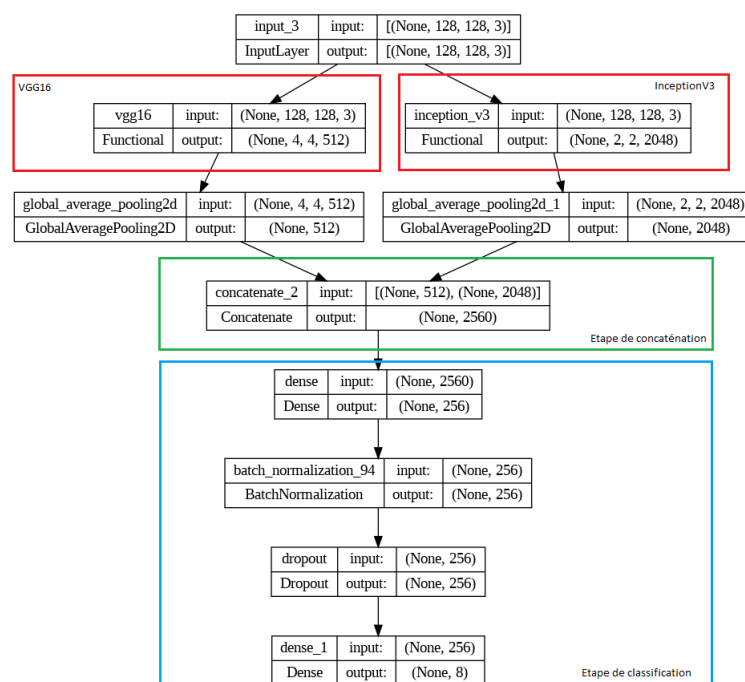


Figure 4.13: Architecture du Modèle 1

4.4.2 Modèle 2 (InceptionV3 et ResNet50)

- Comme deuxième modèle, nous avons concaténé le modèle InceptionV3 et ResNet50 (voir chapitre 3) comme le montre la Figure 4.14, et pour la partie classification, nous avons utilisé deux (2) couches denses (entièrement connectées) ayant respectivement 256 et 128 neurones avec "ReLU" comme fonction d'activation. Chacune de ces couches est suivie d'une couche de BatchNormalization et d'une dropout de 50%;
- Ensuite, une dernière couche qui représente la couche de sortie avec comme nombre de neurones huit (8) et une fonction d'activation "softmax".

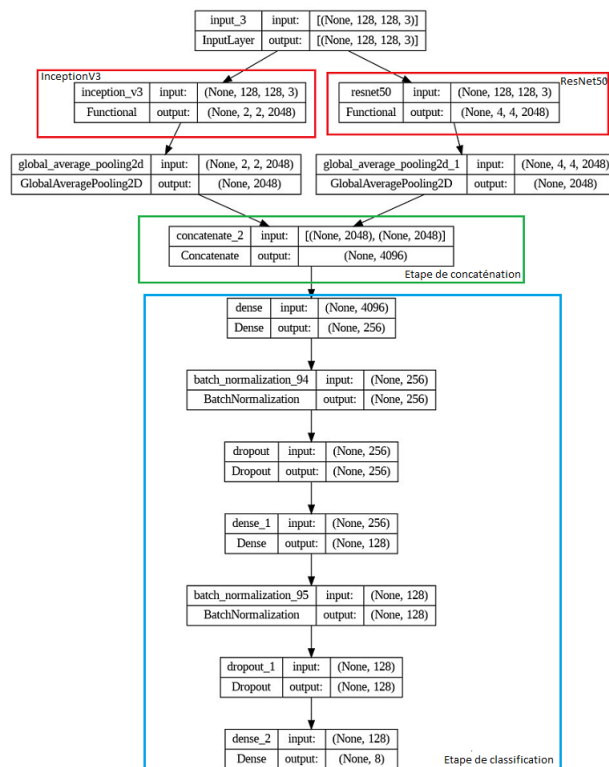


Figure 4.14: Architecture du Modèle 2

4.4.3 Modèle 3 (VGG16 et ResNet50)

- Dans ce modèle, nous avons combiné VGG16 et ResNet50, comme illustré dans la Figure 4.15, en ajoutant trois (3) couches denses de 512, 256 et 125 neurones avec une activation "ReLU". Chacune de ces couches est suivie d'une couche de BatchNormalization et d'une dropout de 50%;
- Après cela, une couche de sortie avec activation "softmax" a été ajoutée afin de construire notre classifieur.

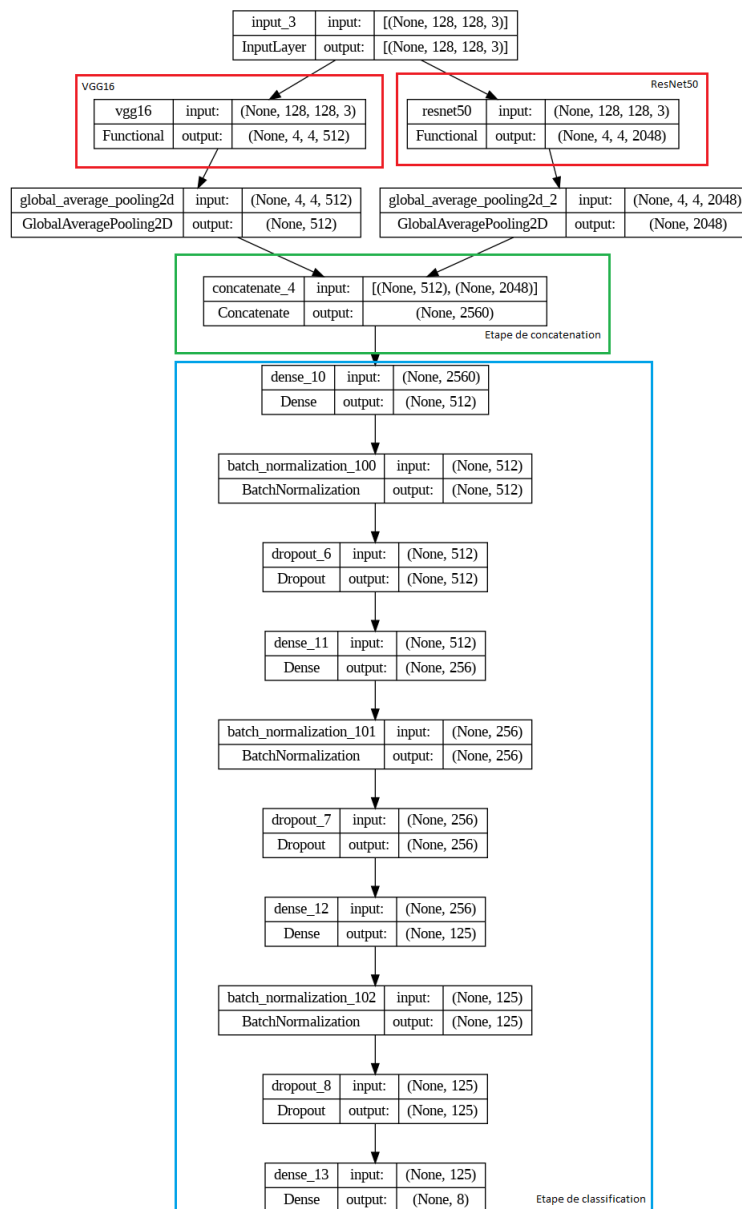


Figure 4.15: Architecture du Modèle 3

4.5 Pré-traitement

4.5.1 Chargement du dataset

Dans cette partie, nous avons chargé le dataset et appliqué une fonction que nous avons programmée; appelée `read_images()`, comme le montre la Figure 4.16. Cette fonction est conçue pour lire chaque image, convertir sa couleur de BGR (Bleu, Green, Red) en RGB, un format largement utilisé par la majorité des bibliothèques de traitement d'images et des outils de visualisation tels que matplotlib. De plus, elle redimensionne les images en une taille uniforme de 128x128 pixels, sachant que la dimension initiale est de 360x363 pixels (pour éviter les problèmes de mémoire). La fonction renvoie une liste contenant toutes les images du dataset.

```
def read_images(data):
    lst_images = []
    for i in range(len(data)):
        img = cv2.imread(data[i])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (128, 128))
        lst_images.append(img)
    return lst_images
```

Figure 4.16: La fonction `read_images()`

Après le chargement du dataset, nous avons constaté un déséquilibre dans la distribution du nombre d'images par classe (imbalanced dataset), comme illustré dans la Figure 4.17. En effet, les classes neutrophiles, éosinophiles, granulocytes immatures(IG) et plaquettes contiennent un nombre important d'images par rapport aux autres classes, ce qui provoque des problèmes d'apprentissage tels que le sur-apprentissage (overfitting), où ces classes seront mieux classées par rapport aux restes des classes. Pour résoudre ce problème, nous avons utilisé le principe de `class_weights`, qui permet d'attribuer des poids à chaque classe afin de pénaliser les classes majoritaires au profit des classes minoritaires pendant l'entraînement, ci-dessous les poids attribués à chaque classe:

0: 0.6418454935622318, 1: 0.6853803849679193, 2: 1.7532239155920282, 3: 1.7594117647058825, 4: 1.5045271629778671, 5: 0.7381539980256664, 6: 1.37707182320442, 7: 0.9102251978088862.

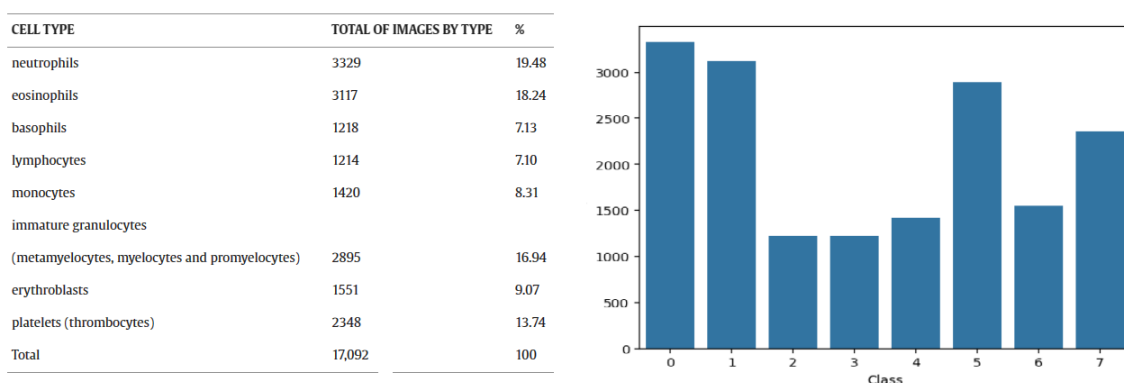


Figure 4.17: La distribution des huit (8) classes (0: Neutrophiles, 1: Éosinophiles, 2: Basophiles, 3: Lymphocytes, 4: Monocytes, 5: Granulocytes Immatures(IG), 6: Érythroblastes, 7: Plaquettes)

4.5.2 Répartition des données

Les images ainsi chargées sont ensuite converties en un tableau NumPy, puis réparties en sous-ensembles de données d'apprentissage, de validation et de test. Les données d'apprentissage représentent 70% du dataset, tandis que les 30% restants sont partagés à parts égales entre les données de test et de validation, comme le montre la Figure 4.18.

```
#Conversion en tableau numpy
Y = np.asarray(labels)

X = np.asarray(images)

#Division des données en données de train, test, validation
X_train, X_temp, Y_train, Y_temp = train_test_split(X, Y,
                                                    test_size= 0.30, stratify= Y, random_state=101)

print("X_train : ",X_train.shape," X_temp : ",X_temp.shape)

X_test, X_val, Y_test, Y_val = train_test_split(X_temp, Y_temp,
                                                test_size= 0.50, stratify= Y_temp, random_state=101)

print("X_test : ",X_test.shape," X_val : ",X_val.shape)
```

Figure 4.18: Division des données en données d'apprentissage, de validation et de test

4.5.3 Normalisation des données

Afin d'uniformiser nos données et d'accélérer l'apprentissage, nous avons appliqué une technique de normalisation qui représente chaque pixel sur une échelle de $[0, 1]$, comme le montre la Figure 4.19, avec une conversion des données vers le type float32. Cette étape permet de garantir que les valeurs des pixels des images se situent dans une plage cohérente, ce qui facilite l'apprentissage des modèles.

```
#Changement du type en float32
X_train= X_train.astype('float32')

X_val= X_val.astype('float32')

X_test= X_test.astype('float32')

#Normalisation
X_train= X_train/255

X_val= X_val/255

X_test= X_test/255

#Conversion des vecteurs labels en format binaire
Y_train= keras.utils.to_categorical (Y_train, 8)

Y_val= keras.utils.to_categorical ( Y_val, 8)

Y_test= keras.utils.to_categorical ( Y_test, 8)
```

Figure 4.19: Normalisation des données

La fonction `to_categorical()` transforme le vecteur d'étiquettes en une matrice où chaque ligne correspond à une étiquette sous forme de vecteur binaire. Chaque colonne dans cette matrice binaire correspond à une catégorie et seul l'index correspondant à l'étiquette est marqué par "1", tous les autres éléments étant "0" (voir Figure 4.20).

Codages								Classes
1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	2
0	0	0	1	0	0	0	0	3
0	0	0	0	1	0	0	0	4
0	0	0	0	0	1	0	0	5
0	0	0	0	0	0	1	0	6
0	0	0	0	0	0	0	1	7

Figure 4.20: Catégorisation des huit (8) classes des cellules

4.6 Apprentissage des modèles

Pour entraîner nos différents modèles (3 architectures), le choix des valeurs des hyperparamètres est une étape importante afin de garantir des résultats optimaux. Dans notre travail, ces paramètres ont été choisis expérimentalement de la manière suivante:

- **Optimizer** : un optimiseur de gradient stochastique (SGD) a été utilisé afin d'affiner les réseaux;
- **Loss Function** : nous avons utilisé *categorical_crossentropy* en tant que fonction de perte pour le Modèle 1 et le Modèle 3 et *sparse_categorical_crossentropy* pour le Modèle 2;
- **Epochs / batch size** : La taille des lots (batches) et le nombre d'époques (epochs) ont été fixés respectivement à 64 et 100.

En outre, pour améliorer la performance de nos modèles et éviter le sur-apprentissage (overfitting), nous avons ajouté quelques améliorations (voir Figure 4.21) telles que:

1. **EarlyStopping**: est particulièrement utile pour arrêter l'entraînement du modèle dès que sa performance cesse de s'améliorer sur le jeu de données de validation, permettant ainsi de conserver les meilleurs poids obtenus et de réduire le temps de formation inutile;
2. **ReduceLRonPlateau**: ajuste le pas d'apprentissage (learning rate) pendant l'entraînement, en réduisant ce taux lorsque la métrique surveillée (accuracy) ne s'améliore plus;
3. **Data augmentation**: génère de nouvelles images en appliquant des opérations de zoom, de rotation, de décalage ...etc, dans le but de rendre le modèle plus robuste face aux variations des images lors de la phase de test. La Figure 4.22 présente quelques images après l'application de la technique d'augmentation des données sur une image originale.

```

early = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

learning = keras.callbacks.ReduceLRonPlateau(factor=0.00001, patience=5,
                                             monitor="val_accuracy", min_lr=0.0000001)

data_generator = ImageDataGenerator(width_shift_range=0.1,
                                   height_shift_range = 0.1,
                                   zoom_range = 0.1,
                                   rotation_range = 10,
                                   horizontal_flip = True,
                                   vertical_flip = True,
                                   shear_range = 0.2,
                                   fill_mode='nearest')

data_generator.fit(X_train)

```

Figure 4.21: Techniques d'améliorations appliquées

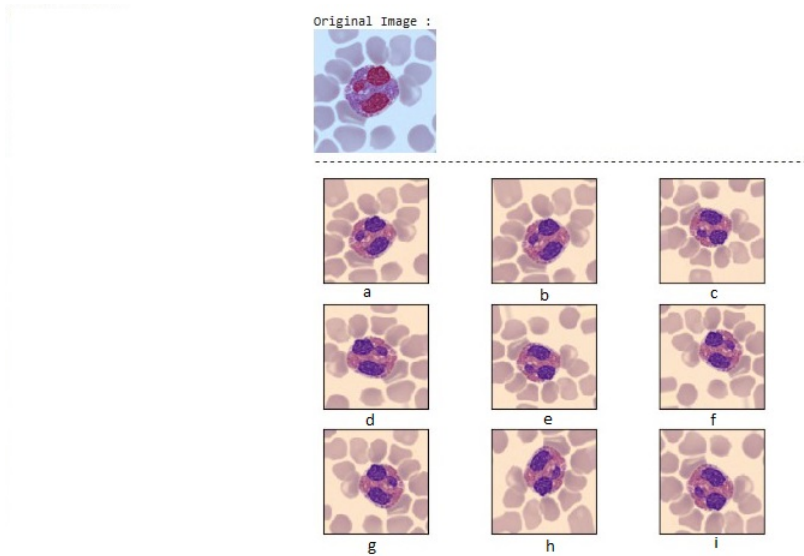


Figure 4.22: Exemple d'images après augmentation (zoom, rotation, vertical flip, horizontal flip ...etc.)

Comme métrique d'évaluation pendant l'apprentissage, nous avons choisi l'accuracy pour superviser le taux d'apprentissage, les Figures 4.23, 4.24 et 4.25 montrent quelques performances à certains moments (epochs) de l'apprentissage.

```

Epoch 1/100
186/186 [=====] - 95s 307ms/step - loss: 0.7180 - accuracy: 0.8171 - val_loss: 1.3300 - val_accuracy: 0.7730
Epoch 2/100
186/186 [=====] - 45s 243ms/step - loss: 0.2366 - accuracy: 0.9383 - val_loss: 0.2990 - val_accuracy: 0.9098
Epoch 3/100
186/186 [=====] - 45s 242ms/step - loss: 0.1243 - accuracy: 0.9602 - val_loss: 0.1705 - val_accuracy: 0.9438
Epoch 4/100
186/186 [=====] - 45s 243ms/step - loss: 0.0891 - accuracy: 0.9713 - val_loss: 0.0704 - val_accuracy: 0.9746
Epoch 5/100
186/186 [=====] - 45s 240ms/step - loss: 0.0790 - accuracy: 0.9743 - val_loss: 0.1351 - val_accuracy: 0.9547
Epoch 6/100
186/186 [=====] - 46s 244ms/step - loss: 0.0676 - accuracy: 0.9782 - val_loss: 0.2461 - val_accuracy: 0.9211
Epoch 7/100
186/186 [=====] - 45s 242ms/step - loss: 0.0622 - accuracy: 0.9790 - val_loss: 0.1787 - val_accuracy: 0.9453
Epoch 8/100
186/186 [=====] - 45s 242ms/step - loss: 0.0572 - accuracy: 0.9801 - val_loss: 0.0717 - val_accuracy: 0.9746
Epoch 9/100
186/186 [=====] - 46s 245ms/step - loss: 0.0543 - accuracy: 0.9809 - val_loss: 0.2220 - val_accuracy: 0.9348
Epoch 10/100
186/186 [=====] - 45s 242ms/step - loss: 0.0416 - accuracy: 0.9866 - val_loss: 0.1084 - val_accuracy: 0.9617
Epoch 11/100
186/186 [=====] - 45s 242ms/step - loss: 0.0485 - accuracy: 0.9823 - val_loss: 0.0573 - val_accuracy: 0.9793
Epoch 12/100
186/186 [=====] - 45s 241ms/step - loss: 0.0384 - accuracy: 0.9848 - val_loss: 0.0804 - val_accuracy: 0.9695
Epoch 13/100
186/186 [=====] - 45s 243ms/step - loss: 0.0361 - accuracy: 0.9876 - val_loss: 0.1524 - val_accuracy: 0.9563
Epoch 14/100
186/186 [=====] - 45s 241ms/step - loss: 0.0350 - accuracy: 0.9866 - val_loss: 0.1355 - val_accuracy: 0.9600

```

Figure 4.23: Apprentissage du Modèle 1

```

Epoch 1/100
186/186 [=====] - 81s 244ms/step - loss: 0.3442 - accuracy: 0.8949 - val_loss: 9.4133 - val_accuracy: 0.2172
Epoch 2/100
186/186 [=====] - 45s 239ms/step - loss: 0.0768 - accuracy: 0.9794 - val_loss: 3.3713 - val_accuracy: 0.4629
Epoch 3/100
186/186 [=====] - 44s 238ms/step - loss: 0.0608 - accuracy: 0.9831 - val_loss: 1.4394 - val_accuracy: 0.6215
Epoch 4/100
186/186 [=====] - 45s 240ms/step - loss: 0.0568 - accuracy: 0.9827 - val_loss: 0.2237 - val_accuracy: 0.9367
Epoch 5/100
186/186 [=====] - 44s 237ms/step - loss: 0.0411 - accuracy: 0.9874 - val_loss: 0.1117 - val_accuracy: 0.9727
Epoch 6/100
186/186 [=====] - 45s 239ms/step - loss: 0.0487 - accuracy: 0.9847 - val_loss: 0.0950 - val_accuracy: 0.9699
Epoch 7/100
186/186 [=====] - 45s 239ms/step - loss: 0.0364 - accuracy: 0.9884 - val_loss: 0.0554 - val_accuracy: 0.9824
Epoch 8/100
186/186 [=====] - 45s 240ms/step - loss: 0.0381 - accuracy: 0.9887 - val_loss: 0.1004 - val_accuracy: 0.9695
Epoch 9/100
186/186 [=====] - 44s 237ms/step - loss: 0.0317 - accuracy: 0.9904 - val_loss: 0.0671 - val_accuracy: 0.9773
Epoch 10/100
186/186 [=====] - 45s 240ms/step - loss: 0.0329 - accuracy: 0.9898 - val_loss: 0.0763 - val_accuracy: 0.9777
Epoch 11/100
186/186 [=====] - 45s 239ms/step - loss: 0.0292 - accuracy: 0.9910 - val_loss: 0.1310 - val_accuracy: 0.9641
Epoch 12/100
186/186 [=====] - 45s 239ms/step - loss: 0.0268 - accuracy: 0.9924 - val_loss: 0.0841 - val_accuracy: 0.9719
Epoch 13/100
186/186 [=====] - 44s 238ms/step - loss: 0.0301 - accuracy: 0.9897 - val_loss: 0.2134 - val_accuracy: 0.9473
Epoch 14/100
186/186 [=====] - 44s 237ms/step - loss: 0.0282 - accuracy: 0.9906 - val_loss: 0.0659 - val_accuracy: 0.9785
Epoch 15/100
186/186 [=====] - 44s 236ms/step - loss: 0.0285 - accuracy: 0.9908 - val_loss: 0.1354 - val_accuracy: 0.9637
Epoch 16/100
186/186 [=====] - 45s 240ms/step - loss: 0.0221 - accuracy: 0.9930 - val_loss: 0.1171 - val_accuracy: 0.9648
Epoch 17/100
186/186 [=====] - 45s 243ms/step - loss: 0.0214 - accuracy: 0.9935 - val_loss: 0.0674 - val_accuracy: 0.9803

```

Figure 4.24: Apprentissage du Modèle 2

```

Epoch 1/100
186/186 [=====] - 79s 323ms/step - loss: 0.5190 - accuracy: 0.7554 - val_loss: 0.1368 - val_accuracy: 0.9446
Epoch 2/100
186/186 [=====] - 58s 311ms/step - loss: 0.1658 - accuracy: 0.9461 - val_loss: 0.0566 - val_accuracy: 0.9735
Epoch 3/100
186/186 [=====] - 58s 312ms/step - loss: 0.1157 - accuracy: 0.9626 - val_loss: 0.0489 - val_accuracy: 0.9762
Epoch 4/100
186/186 [=====] - 58s 312ms/step - loss: 0.0894 - accuracy: 0.9702 - val_loss: 0.0465 - val_accuracy: 0.9797
Epoch 5/100
186/186 [=====] - 58s 311ms/step - loss: 0.0802 - accuracy: 0.9718 - val_loss: 0.0330 - val_accuracy: 0.9860
Epoch 6/100
186/186 [=====] - 58s 311ms/step - loss: 0.0748 - accuracy: 0.9730 - val_loss: 0.0710 - val_accuracy: 0.9680
Epoch 7/100
186/186 [=====] - 58s 311ms/step - loss: 0.0607 - accuracy: 0.9774 - val_loss: 0.0982 - val_accuracy: 0.9524
Epoch 8/100
186/186 [=====] - 58s 311ms/step - loss: 0.0613 - accuracy: 0.9769 - val_loss: 0.0351 - val_accuracy: 0.9832
Epoch 9/100
186/186 [=====] - 58s 311ms/step - loss: 0.0581 - accuracy: 0.9767 - val_loss: 0.0372 - val_accuracy: 0.9840
Epoch 10/100
186/186 [=====] - 58s 311ms/step - loss: 0.0519 - accuracy: 0.9800 - val_loss: 0.0531 - val_accuracy: 0.9743
Epoch 11/100
186/186 [=====] - 58s 310ms/step - loss: 0.0527 - accuracy: 0.9792 - val_loss: 0.2688 - val_accuracy: 0.8658
Epoch 12/100
186/186 [=====] - 58s 311ms/step - loss: 0.0464 - accuracy: 0.9822 - val_loss: 0.0372 - val_accuracy: 0.9824
Epoch 13/100
186/186 [=====] - 58s 313ms/step - loss: 0.0437 - accuracy: 0.9833 - val_loss: 0.0321 - val_accuracy: 0.9840
Epoch 14/100
186/186 [=====] - 59s 314ms/step - loss: 0.0392 - accuracy: 0.9843 - val_loss: 0.3093 - val_accuracy: 0.8397
Epoch 15/100
186/186 [=====] - 58s 312ms/step - loss: 0.0422 - accuracy: 0.9819 - val_loss: 0.0871 - val_accuracy: 0.9583
Epoch 16/100
186/186 [=====] - 58s 311ms/step - loss: 0.0451 - accuracy: 0.9819 - val_loss: 0.1276 - val_accuracy: 0.9372
Epoch 17/100
186/186 [=====] - 58s 310ms/step - loss: 0.0405 - accuracy: 0.9825 - val_loss: 0.0402 - val_accuracy: 0.9809

```

Figure 4.25: Apprentissage du Modèle 3

4.7 Résultats et Discussion

Les trois modèles ont été évalués sur le jeu de données d'images évoqué dans la section 4.3. Les expériences ont été réalisées sur des sous-ensembles distincts pour l'apprentissage, la validation et le test. Le nombre de paramètres entraînaibles pour chaque modèle utilisé est de 37 141 224, 46 386 472, 39 726 759 respectivement pour le Modèle 1 (VGG16 et InceptionV3), le Modèle 2 (InceptionV3 et ResNet50) et enfin le Modèle 3 (VGG16 et ResNet50).

Durant la phase d'apprentissage, nous avons configuré le paramètre `EarlyStopping` à 10 pour le Modèle 1. Ainsi, l'apprentissage s'est arrêté au bout de la 27^{ème} époque sur 100, avec un taux d'accuracy de 99.17% pour l'entraînement et 98.55% pour la validation. Les pertes (taux d'erreurs) observées respectivement étaient de 0.0230 et 0.0402. L'apprentissage a duré 53 minutes.

Concernant le Modèle 2, l'`EarlyStopping` est défini à 20. Les taux d'accuracy pour l'entraînement et la validation étaient respectivement de 99.35% et 98.36% avec des pertes de 0.0234 et 0.0639, qui ont

été observées à la 32ème époque sur 100, pour une durée de 24 minutes.

Enfin pour le Modèle 3, nous avons également utilisé une valeur de 20 pour l'EarlyStopping. L'apprentissage a cessé à la 30ème époque sur 100, avec des taux d'accuracy de 99.34% et 98.83% et des pertes de 0.0172 et 0.0330 respectivement pour l'entraînement et la validation, la durée totale était de 49 minutes.

Les figures 4.26 , 4.27 et 4.28 illustrent les performances des modèles lors de la phase d'apprentissage où l'on trouve la fonction perte (Loss) et l'accuracy sur l'ensemble d'entraînement et de validation pour les trois modèles. La faible variance dans les graphes indique que nos modèles ne sont pas sur_ajustés (overfitting).

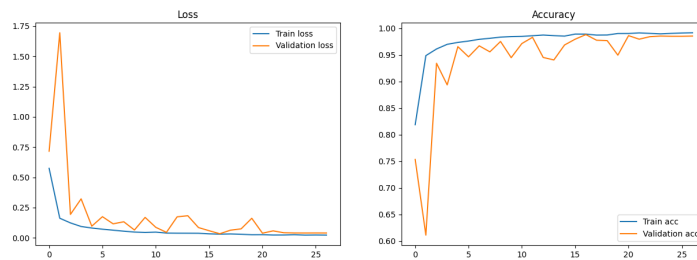


Figure 4.26: Performances de l'apprentissage du Modèle 1 (VGG16 et InceptionV3)

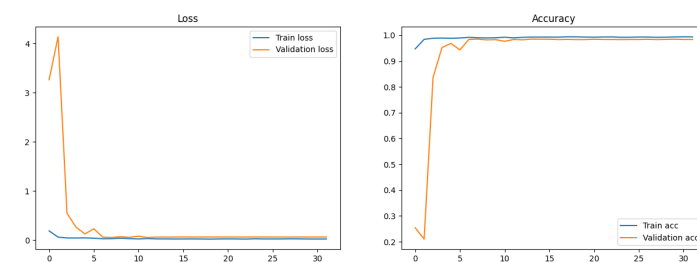


Figure 4.27: Performances de l'apprentissage du Modèle 2 (InceptionV3 et ResNet50)

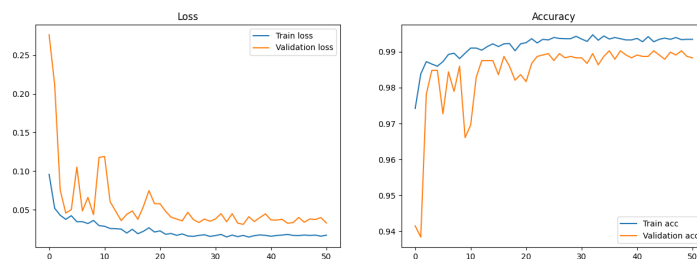


Figure 4.28: Performances de l'apprentissage du Modèle 3 (VGG16 et ResNet50)

Les courbes obtenues sont plutôt satisfaisantes, notamment celle du Modèle 2 où l'on voit qu'il y a une parfaite harmonie entre la courbe d'apprentissage et celle de validation, pareil pour les courbes de la fonction perte qui décroissent au fur et à mesure que l'indice d'accuracy de l'apprentissage et de la validation augmente. L'erreur d'entraînement et l'erreur de validation sont faibles et proches l'une de l'autre pour les trois modèles. Comme le montre les figures, au bout d'un certain nombre d'époques, les courbes de perte et d'accuracy deviennent constantes. Cela indique que les modèles ont atteint un point de convergence où il n'y a plus d'amélioration dans leurs performances, donc on peut en déduire que les modèles généralisent bien sur de nouvelles données.

Pour prouver cela, nous avons sauvegardé les poids de nos trois modèles CNN (fichier extension .h5) et avons évalué leurs performances sur nos données de test (2 564 images). Les résultats de précision (courbes ROC) démontrent que les modèles fonctionnent bien en classifiant correctement les différents types de cellules sanguines périphériques. La Figure 4.29 montre la courbe de caractéristique de fonctionnement du récepteur (ROC) de nos modèles sur les données de test encore jamais vues, atteignant ainsi une surface sous la courbe (AUC) égale à 1.000, 1.000 et 0.999 respectivement pour les trois modèles.

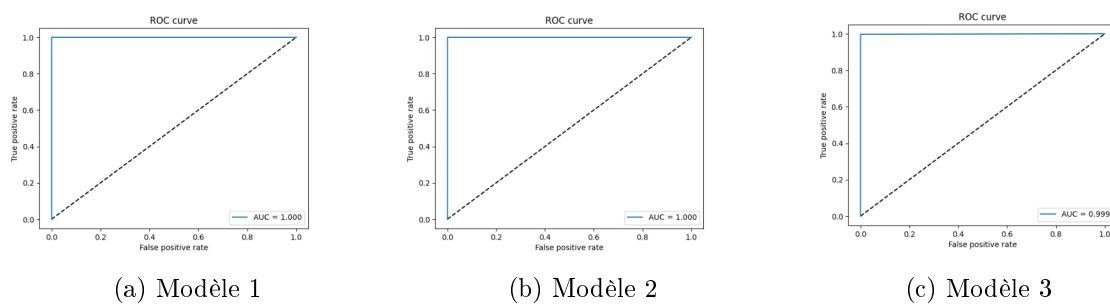


Figure 4.29: Les courbes ROC

De plus, nous avons affiché les matrices de confusion pour chaque modèle afin d'analyser et visualiser les erreurs de classification. La matrice de confusion permet de comparer les prédictions du modèle avec les véritables classes (valeurs réelles). Chaque ligne de la matrice représente les instances de la classe prédite, tandis que chaque colonne représente les instances de la classe réelle (ou vice versa). Le tableau 4.1 représente la matrice de confusion pour un problème de classification binaire.

	Classe réelle positive	Classe réelle négative
Classe prédite positive	True Positif (TP)	False Positif (FP)
Classe prédite négative	False Negatif (FN)	True Negatif (TN)

Table 4.1: Matrice de confusion binaire

Par définition en classification binaire, un résultat est dit :

- Vrai Positif (True Positif "TP") : lorsque la classe prédite est positive et la classe réelle est également positive;
- Faux Positif (False Positif "FP") : lorsque la classe prédite est positive mais la classe réelle est négative;

- Vrai Négatif (True Negatif "TN"): lorsque la classe prédite est négative et la classe réelle est également négative;
- Faux Négatif (False Negatif "FN"): lorsque la classe prédite est négative mais la classe réelle est positive.

Les résultats illustrés par les Figures 4.30 , 4.31 et 4.32 représentent la matrice de confusion pour chaque modèle, où l'on trouve les huit (8) classes (types des cellules) sur les axes X et Y avec une diagonale qui représente les vrais positifs.

Avec l'architecture du Modèle 1, un total de vingt-huit (28) images sur 2 564 ont été mal classées (1.09%), comme le montre la Figure 4.30 . Il s'agit notamment de six (6) images de neutrophiles qui ont été classées comme des granulocytes immatures , deux (2) images de lymphocytes comme étant des monocytes, trois (3) images de monocytes ont été mal classées, une comme lymphocyte et deux (2) comme granulocytes immatures. En plus, onze (11) images de granulocytes immatures ont été mal classées, sept (7) comme neutrophiles, trois (3) comme basophiles et une comme érythroblaste. Enfin, deux (2) images d'érythroblastes ont été classées comme étant des granulocytes immatures.

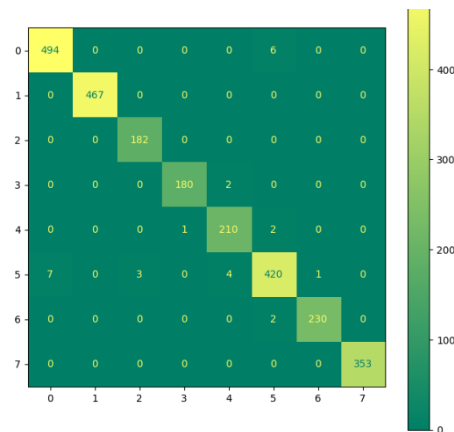


Figure 4.30: Matrice de confusion du Modèle 1

Pour l'architecture du Modèle 2, quarante (40) images ont été mal classées sur les 2 564 (1.56%). Parmi elles, neuf (9) images de neutrophiles ont été classées comme des granulocytes immatures, neuf (9) images de lymphocytes dont six (6) comme étant des monocytes et trois (3) comme des érythroblastes, sept (7) images de monocytes, une comme lymphocyte et six (6) comme granulocytes immatures, treize (13) images de granulocytes immatures ont été mal classées, neuf (9) comme neutrophiles, une comme basophile, une comme monocyte et deux (2) comme érythroblastes et pour finir deux (2) images d'érythroblastes ont été classées comme des lymphocytes (voir Figure 4.31).

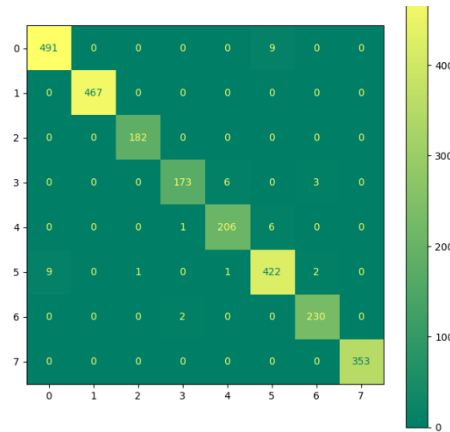


Figure 4.31: Matrice de confusion du Modèle 2

Avec le Modèle 3, un total de quarante-quatre (44) images sur 2 564 ont été mal classées (1.71%). Plus précisément, dix (10) images de neutrophiles, dont une classée comme basophile et neuf (9) comme des granulocytes immatures, une image de lymphocyte comme monocyte, six (6) images de monocytes, une comme lymphocyte et cinq (5) comme étant des granulocytes immatures, vingt-six (26) images de granulocytes immatures dont seize (16) classées comme neutrophiles, huit (8) comme basophiles et deux (2) comme érythroblastes et enfin une image d'érythroblaste a été classée comme basophile. (voir Figure 4.32)

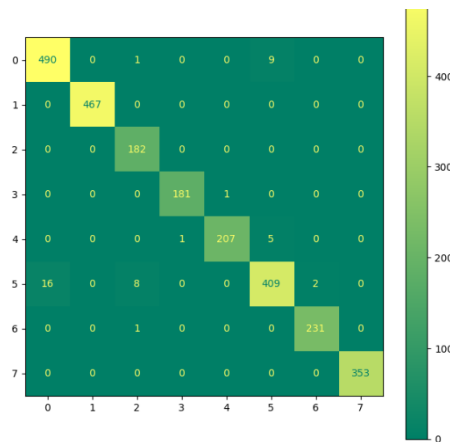


Figure 4.32: Matrice de confusion du Modèle 3

La classification erronée de ces cellules peut être attribuée à leur grande similitude de taille, de forme et de leur variabilité intra-classe car même les cellules du même type peuvent présenter une grande variabilité dans leur apparence en raison de divers facteurs biologiques (voir Figure 4.33). Cette variabilité peut rendre la tâche de classification plus difficile pour le modèle.

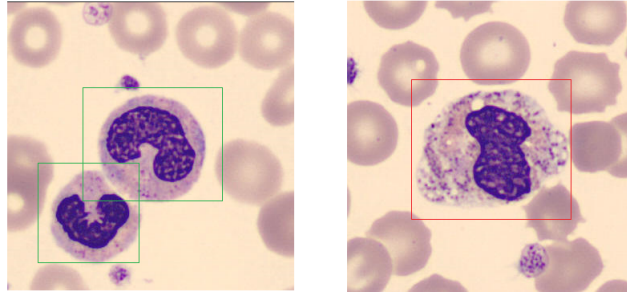


Figure 4.33: Deux images de même type cellulaire (neutrophile)

Lors de l'évaluation de nos modèles, trois métriques ont été considérées qui sont:

- Precision: est le nombre total de vrais positifs divisé par le nombre total d'éléments étiquetés comme appartenant à la classe positive;

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- Recall: est le nombre total de vrais positifs divisé par le nombre total d'éléments qui appartiennent réellement à la classe positive;

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- F1_score: est calculé en combinant la précision et le rappel (recall) d'un modèle. Il est particulièrement utile lorsque les classes sont déséquilibrées.

$$F1_score = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \quad (4)$$

La Figure 4.34 montre les performances de nos trois modèles sur les données de test, où l'on trouve les statistiques pour chaque type de cellules (8 types: classe 0 jusqu'à 7) ainsi que la moyenne des performances pour chaque modèle.

	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.99	0.99	0.99	500	0	0.98	0.98	0.98	500	0	0.97	0.98	0.97	500
1	1.00	1.00	1.00	467	1	1.00	1.00	1.00	467	1	1.00	1.00	1.00	467
2	0.98	1.00	0.99	182	2	0.99	1.00	1.00	182	2	0.95	1.00	0.97	182
3	0.99	0.99	0.99	182	3	0.98	0.95	0.97	182	3	0.99	0.99	0.99	182
4	0.97	0.99	0.98	213	4	0.97	0.97	0.97	213	4	1.00	0.97	0.98	213
5	0.98	0.97	0.97	435	5	0.97	0.97	0.97	435	5	0.97	0.94	0.95	435
6	1.00	0.99	0.99	232	6	0.98	0.99	0.99	232	6	0.99	1.00	0.99	232
7	1.00	1.00	1.00	353	7	1.00	1.00	1.00	353	7	1.00	1.00	1.00	353
accuracy			0.99	2564	accuracy			0.98	2564	accuracy			0.98	2564
macro avg	0.99	0.99	0.99	2564	macro avg	0.98	0.98	0.98	2564	macro avg	0.98	0.99	0.98	2564
weighted avg	0.99	0.99	0.99	2564	weighted avg	0.98	0.98	0.98	2564	weighted avg	0.98	0.98	0.98	2564

(a) Modèle 1

(b) Modèle 2

(c) Modèle 3

Figure 4.34: Rapport de classification

Le Modèle 2 a atteint des performances légèrement supérieures à celles des deux autres modèles lors de l'apprentissage, mais le Modèle 1 généralise mieux sur de nouvelles données, comme le montre la figure de l'évaluation, car au moment de l'évaluation le Modèle 1 obtient une précision supérieure à celle des autres. Il est possible que les données spécifiques utilisées favorisent les caractéristiques capturées par VGG16 et InceptionV3. Les précisions globales obtenues sont 99% , 98% et enfin 98%

respectivement pour le Modèle 1, le Modèle 2 et le Modèle 3.

Dans le but d’approfondir notre évaluation, Le tableau 4.2 montre une comparaison de nos modèles avec d’autres modèles pertinents proposés dans la littérature et qui utilisent la même base d’images (8 classes).

Auteurs	Dataset	Taille	Modèles	Accuracy
[Acevedo et al., 2019]	PBC	17 092	VGG16 et InceptionV3 comme extracteur de caractéristiques SVM comme classifieur	86% et 90%
[Acevedo et al., 2019]	PBC	17 092	VGG16 et InceptionV3	96% et 95%
[Chola et al., 2022]	PBC	17 092	EfficientNet-B0 + 2 couches Dense	98.51%
[Alkafrawi and Dakhell, 2022]	PBC	17 092	AlexNet	95.08%
Notre Modèle 1	PBC	17 092	VGG16 + InceptionV3	99%
Notre Modèle 2	PBC	17 092	InceptionV3 + ResNet50	98%
Notre Modèle 3	PBC	17 092	VGG16 + ResNet50	98%

Table 4.2: Tableau de comparaison

Par rapport aux travaux [Acevedo et al., 2019], nos modèles ont atteint des précisions largement supérieures, cette amélioration peut être justifiée par la concaténation des modèles CNN, qui permet de combiner leurs avantages pour extraire en mieux les caractéristiques complexes. En plus de cela, cette combinaison compense les erreurs spécifiques de chaque modèle, ce qui peut améliorer la précision globale et la capacité de généralisation.

En ce qui concerne les modèles proposés par [Chola et al., 2022] et [Alkafrawi and Dakhell, 2022], on constate que notre Modèle 1 a obtenu une précision de 99% dépassant ainsi la précision obtenue par ces derniers. Donc on peut conclure que notre modèle (VGG16 + InceptionV3) est plus adapté pour classer correctement les huit (8) classes du dataset Pheripheral Blood Cell (PBC) que les autres modèles évoqués précédemment.

4.7.1 Prédiction avec l’interface Gradio

Pour la création de l’interface graphique, nous avons opté pour la bibliothèque Gradio de Google (voir Figure 4.36), celle-ci nous permet de créer une interface utilisateur (voir Figure 4.35), afin d’exploiter nos modèles déjà sauvegardés.


```

class_names = { 0:'Neutrophile', 1:'Eosinophile', 2:'Basophile', 3:'Lymphocyte', 4:'Monocyte',
               5:'Granulocyte immature', 6:'Erythroblast', 7:'Platelet'}

def predict_input_image(img):
    image = cv2.resize(img, (128, 128))
    image = image/255.0
    image = image.reshape((-1,128,128,3))
    prediction= base.predict(image)[0]
    predict = np.argmax(prediction)
    print(predict)
    return {class_names[i]: float(prediction[i]) for i in range(8)}

css = """
h1{text-align: center;
display: block}
"""

with gr.Blocks(css = css) as demo:
    gr.Markdown("# Classification d'image Cytologique")
    gr.Interface(fn=predict_input_image, inputs=gr.Image(), outputs=gr.Label(num_top_classes=8))

if __name__ == "__main__":
    demo.launch(debug='True')

```

Figure 4.35: Implémentation de l'interface utilisateur



Figure 4.36: Page d'accueil (à gauche: les inputs, à Droite: les outputs)

Pour prédire la classe d'une image, on clique sur « Déposer l'image ici » puis sélectionner l'image dans l'emplacement correspondant. Une fois l'image sélectionnée, l'utilisateur envoie l'image au modèle pré-entraîné en cliquant sur le bouton « Submit » et enfin le modèle répond comme suit (voir Figure 4.37).

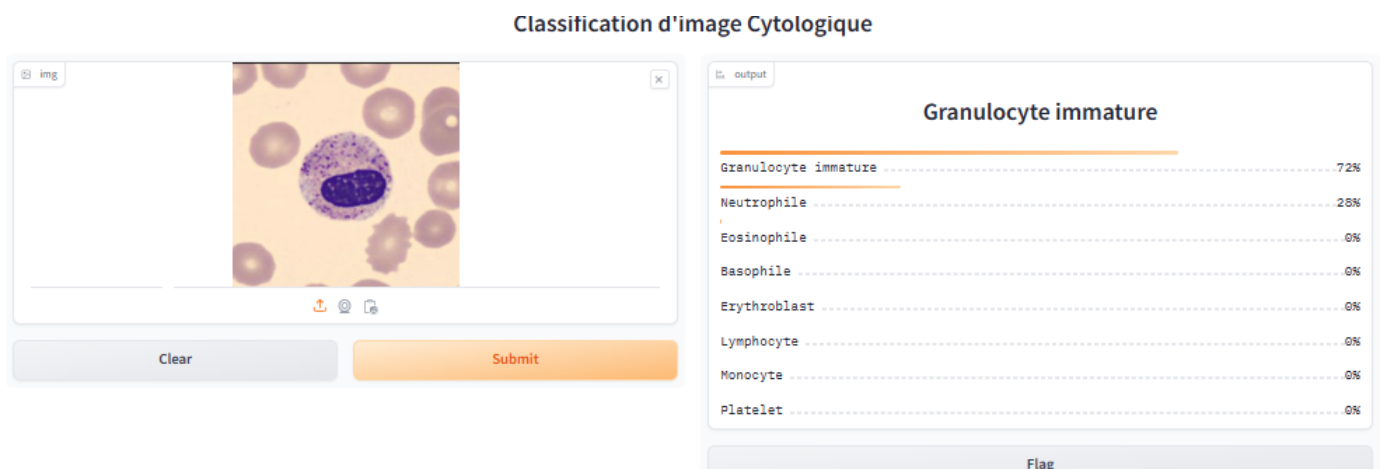


Figure 4.37: Exemple de prédiction (à gauche : image d'entrée, à droite : résultat de classification)

Dans notre exemple, nous avons introduit une image de type granulocyte immature et notre modèle a donné un pourcentage de 72%, comme le montre la Figure 4.37, ce qui montre qu'il a bien reconnu l'image d'entrée. De plus il a attribué un très faible pourcentage aux autres classes, démontrant ainsi le bon fonctionnement de notre modèle.

4.7.2 Difficultés rencontrées

Pendant la réalisation de notre système de classification automatique, nous avons rencontré certains problèmes notamment le problème de ressources, car les ressources allouées par la plateforme google colab étaient limitées et n'arrivaient pas à supporter la capacité de nos données. Lorsqu'on essayait de charger les données avec la taille par défaut des architectures CNN utilisés, c'est à dire 224x224 et 299x299, la session plantait en affichant ce message (voir Figure 4.38), raison pour laquelle nous avons diminué cette taille.

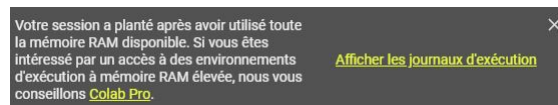


Figure 4.38: Saturation de la RAM

Initialement, la durée accordé par colab pour l'utilisation de la ressource GPU (qui est nécessaire pour l'apprentissage) était de 12h par jour. Cependant, en raison de nos utilisations courantes de la plateforme ainsi que la quantité importante de paramètres des modèles pré-entraînés utilisés tels que VGG16 qui compte à lui seul 138 000 000 paramètres [Web 49], cette durée d'utilisation a été réduite à moins d'une heure par jour. Afin de remédier à ces problèmes et de pouvoir utiliser aisément les ressource, nous avons été obligés de payer un abonnement quotidien (par unités).

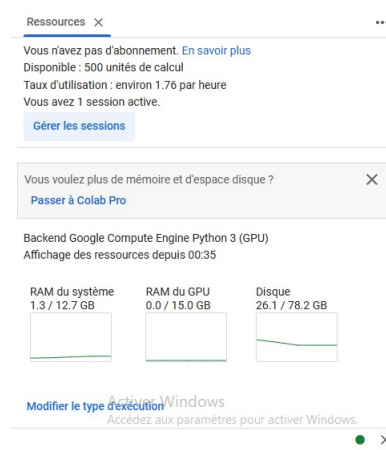


Figure 4.39: Preuve d'achat des unités

4.8 Conclusion

Lors de la conception de notre système de classification des images cytologiques, nous avons constaté l'importance cruciale du choix du réseau de base pour l'extraction des caractéristiques. En plus, la concaténation de différents réseaux s'est avérée essentielle pour caractériser efficacement les différentes données. Le bon ajustement des hyperparamètres tels que le nombre de couches, le nombre de neurones par couche, le nombre d'époques, la taille du batch, les optimiseurs, la fonction de perte ...etc, est primordial pour garantir des performances de qualité. Les résultats obtenus ont permis d'évaluer l'efficacité et la précision de nos modèles.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

CONCLUSION GÉNÉRALE ET PERSPECTIVES

L'analyse et la classification des cellules sanguines sont essentielles pour le diagnostic de presque toutes les maladies hématologiques. Actuellement, c'est un sujet de recherche dans de nombreux domaines où les chercheurs tentent de diagnostiquer les maladies qui affectent les cellules sanguines.

Traditionnellement, la classification des cellules sanguines à l'aide d'images microscopiques du sang est effectuée manuellement par le médecin, ce qui est une tâche fastidieuse et exposée à des erreurs humaines.

L'objectif de ce projet était de réaliser un système de classification automatique, efficace et avec une grande capacité de généralisation des différentes cellules sanguines périphériques à l'aide du Deep Learning. Pour cette fin, nous avons proposé trois méthodes (modèles) basées sur l'apprentissage par transfert en utilisant les modèles convolutifs : VGG16, ResNet50 et InceptionV3 pré-entraînés sur le dataset ImageNet. Le Modèle 1 fusionne le VGG16 et InceptionV3, le Modèle 2 combine le ResNet50 et InceptionV3 et enfin le Modèle 3 associe le VGG16 et ResNet50, le but de ces concaténations était de combiner l'avantage de ces trois modèles CNN.

On trouve dans ce manuscrit les différentes étapes de notre approche et contribution. La première étape consiste au chargement des données suivie des étapes de pré-traitements. L'étape la plus importante est le choix des réseaux de base et des hyperparamètres adéquats afin d'obtenir des modèles robustes et précis. Et enfin, l'étape finale est la classification des huit (8) types de cellules sanguines périphériques. L'expérience a été réalisée avec la base de données PBC normales de [Acevedo et al., 2020] et les résultats obtenus sont très satisfaisants, avec une précision globale maximale de 99% sur l'ensemble des données de test. À notre connaissance, aucun travail basé sur ce dataset avec l'approche de la concaténation n'a obtenu un pourcentage aussi élevé.

Dans l'optique de l'amélioration de notre système, plusieurs points nécessitent un approfondissement, ouvrant ainsi la voie à de nouvelles perspectives et futurs travaux de recherche dans le domaine médical. Enrichir la base de données en y ajoutant d'autres transformations complexes telles que le changement de couleur et les transformations géométriques pour permettre au système de reconnaître les images cellulaires sanguines dans des contextes variés. Nous envisageons également d'utiliser d'autres modèles CNN (Vision Transformer ViT ...etc) avec d'autres fonctions d'optimiseurs et de perte pour faire une étude comparative des différents modèles. Ainsi, améliorer d'avantage le système afin de l'intégrer à temps réel et de le mettre au service des praticiens pour accélérer le diagnostic à travers une plateforme en ligne.

BIBLIOGRAPHIE

- [Acevedo et al., 2019] Acevedo, A., Alférez, S., Merino, A., Puigví, L., and Rodellar, J. (2019). Recognition of peripheral blood cell images using convolutional neural networks. *Computer methods and programs in biomedicine*, 180:105020.
- [Acevedo et al., 2020] Acevedo, A., Merino González, A., Alférez Baquero, E. S., Molina Borrás, Á., Boldú Nebot, L., and Rodellar Benedé, J. (2020). A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in brief*, 30(article 105474).
- [Alkafrawi and Dakhell, 2022] Alkafrawi, I. M. and Dakhell, Z. A. (2022). Blood cells classification using deep learning technique. In *2022 International Conference on Engineering & MIS (ICEMIS)*, pages 1–6. IEEE.
- [Barbe, 2001] Barbe, L. (2001). *Mécanisme d’adhérence des leucocytes aux fibres synthétiques. Application à la filtration sanguine*. PhD thesis, Université Paris-Diderot de Paris VII.
- [Bellman, 1978] Bellman, R. (1978). *An Introduction to Artificial Intelligence: Can Computers Think?* Boyd & Fraser Publishing Company.
- [Benali et al., 2022] Benali, N., Benyaya, C., Hamdani, D., and Kina, F. (2022). Analyse d’images texturées application à l’imagerie médicale. Master’s thesis, Université Echahid Hamma Lakhdar d’El -Oued.
- [Benazzouz et al., 2017] Benazzouz, M., Baghli, I., Benomar, M. L., and Chikh, M. A. (2017). Automated identification of plasma cell in bone marrow images. In *2017 International Conference on Mathematics and Information Technology (ICMIT)*, pages 8–12. IEEE.
- [Benomar, 2018] Benomar, M. L. (2018). *Combinaison adaptative des informations texture et couleur pour la segmentation d’images médicales*. PhD thesis, Université de Tlemcen-Abou Bekr Belkaid.
- [Benomar et al., 2014] Benomar, M. L., Benazzouz, M., and Chikh, M. A. (2014). Segmentation d’images microscopiques basée sur les attributs textures. In *ICA2IT International Conference on Artificial Intelligence and Information Technology. Ouargla, Algeria*.
- [Benomar et al., 2018] Benomar, M. L., Chikh, A., Descombes, X., and Benazzouz, M. (2018). Multi-feature-based approach for white blood cells segmentation and classification in peripheral blood and

- bone marrow images. *International Journal of Biomedical Engineering and Technology*, 35(3):223–241.
- [Cheuque et al., 2022] Cheuque, C., Querales, M., Leon, R., Salas, R., and Torres, R. (2022). An efficient multi-level convolutional neural network approach for white blood cells classification. *Diagnostics*, 12(2):248.
- [Chola et al., 2022] Chola, C., Muaad, A. Y., Bin Heyat, M. B., Benifa, J. B., Naji, W. R., Hemachandran, K., Mahmoud, N. F., Samee, N. A., Al-Antari, M. A., Kadah, Y. M., et al. (2022). Bcnet: A deep learning computer-aided diagnosis framework for human peripheral blood cell identification. *Diagnostics*, 12(11):2815.
- [Çınar and Tuncer, 2021] Çınar, A. and Tuncer, S. A. (2021). Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid alexnet-googlenet-svm. *SN Applied Sciences*, 3:1–11.
- [Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- [Féré, 2018] Féré, M. (2018). *M3S-Développement de la spectroscopie Raman en cytopathologie: Application au diagnostic de la leucémie lymphoïde chronique*. PhD thesis, Université de Reims Champagne-Ardenne.
- [Gafour et al., 2020] Gafour, Y. et al. (2020). *Apprentissage automatique pour une classification des images basée sur les descripteurs*. PhD thesis, Université Djillali Liabes de Sidi Bel Abbes.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Joshi, 2022] Joshi, P. (2022). Predicting movie genres using nlp – an awesome introduction to multi-label classification. *Analytics Vidhya*.
- [Kohler, 2010] Kohler, C. (2010). Les cellules sanguines. *Collège universitaire et hospitalier des histologistes, embryologistes cytologistes et cytogénéticiens*, 2011.
- [Krizhevsky et al., 2017] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.
- [Kumar, 2022] Kumar, A. (2022). Overfitting & underfitting in machine learning. <https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/>. *Acesso em*, 15:27.
- [Lu et al., 2024] Lu, Q., Wang, B., He, Q., Zhang, Q., Guo, L., Li, J., Li, J., and Ma, Q. (2024). A blood cell classification method based on mae and active learning. *Biomedical Signal Processing and Control*, 90:105813.
- [Mahloul and Mecieb, 2023] Mahloul, A. and Mecieb, S. (2023). Système d’aide au diagnostic du cancer du cerveau. Master’s thesis, Université Ibn Khaldoun de Tiaret.

- [Mersaoui and Fouri, 2018] Mersaoui, I. and Fouri, M. (2018). Etude sur l'imagerie médicale: pré-traitement, segmentation et amélioration des exploitations. Master's thesis, Université Saad Dahlab de Blida.
- [Nabi, 2022] Nabi, J. (2022). Multiclass classification with an imbalanced data set. *Built In*.
- [Sara, 2020] Sara, K. (2020). Introduction à l'imagerie médicale. Master's thesis, Université Mohamed Khider de Biskra.
- [Semmame, 2020] Semmame (2020). La microscopie. Technical report, frère Mentouri Constantine1.
- [Sharma et al., 2023] Sharma, S., Gupta, S., Gupta, D., Juneja, S., Gupta, P., Dhiman, G., Kautish, S., et al. (2023). Deep learning model for the automatic classification of white blood cells. *Computational Intelligence and Neuroscience*, 2022.
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Souad and Tinhinane, 2017] Souad, H. and Tinhinane, I. (2017). Segmentation d'images couleur par analyse d'histogrammes bidimensionnels 2d. Master's thesis, Université Mouloud Mammeri de Tizi-Ouzou.
- [Sourour, 2021] Sourour, K. (2021). Classification multi label des catégories fonctionnelles des gènes de levure. Master's thesis, Université Badji Mokhtar d'Annaba.
- [Szegedy et al., 2015] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2015). Rethinking the inception architecture for computer vision.
- [Tagliasacchi and Carboni, 1997] Tagliasacchi, D. and Carboni, G. (1997). Let's observe the blood cells. Technical report.
- [Tahiri et al., 2023] Tahiri, M. A., Bencherqui, A., Karmouni, H., Amakdouf, H., Sayyouri, M., Qjidaa, H., et al. (2023). White blood cell automatic classification using deep learning and optimized quaternion hybrid moments. *Biomedical Signal Processing and Control*, 86:105128.
- [Taileb, 2022] Taileb, A. (2022). Apprentissage profond pour la classification d'images. Master's thesis, Université Larbi Tébessi de Tébessa.
- [Tarimo et al., 2024] Tarimo, S. A., Jang, M.-A., Ngasa, E. E., Shin, H. B., Shin, H., and Woo, J. (2024). Wbc yolo-vit: 2 way-2 stage white blood cell detection and classification with a combination of yolov5 and vision transformer. *Computers in Biology and Medicine*, 169:107875.
- [Togaçar et al., 2020] Togaçar, M., Ergen, B., and Comert, Z. (2020). Classification of white blood cells using deep features obtained from convolutional neural network models based on the combination of feature selection methods. *Applied Soft Computing*, 97:106810.
- [Vandenbroucke, 2000] Vandenbroucke, N. (2000). *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés: application à l'analyse d'images de football*. PhD thesis, Université de Lille 1.

- [Yan and Su, 2009] Yan, X. and Su, X. (2009). *Linear regression analysis: theory and computing*. world scientific.
- [Zhu et al., 2023] Zhu, Z., Ren, Z., Lu, S., Wang, S., and Zhang, Y. (2023). Dlbnet: A deep learning network for classifying blood cells. *Big Data and Cognitive Computing*, 7(2):75.

Webographie

- [Web 1]. L'imagerie médicale [consulté le 21 février 2024]. Disponible sur :
<https://www.cea.fr/comprendre/Pages/sante-sciences-du-vivant/essentiel-sur-imagerie-medicale.aspx#:~:text=L'imagerie%20m%C3%A9dicale%20a%20fait,acquiert%20en%201897%20un%20radioscope.>
- [Web 2]. Codage HSL (TSL) [consulté le 28 février 2024]. Disponible sur :
<https://web.maths.unsw.edu.au/~lafaye/CCM/video/hsl-tsl.htm>.
- [Web 3]. Imagerie médicale [consulté le 21 février 2024]. Disponible sur :
https://www.larousse.fr/encyclopedie/medical/imagerie_mdicale/13805.
- [Web 4]. Histologie : tout savoir sur l'étude des tissus biologique [consulté le 22 février 2024].
Disponible sur
[:https://www.passeportsante.net/fr/specialites-medicales/Fiche.aspx?doc=histologie-savoir-sur-etude-tissus-biologiques#:~:text=L'histologie%20est%20une%20branche,ailleurs%20parler%20d'anatomie%20microscopique.](https://www.passeportsante.net/fr/specialites-medicales/Fiche.aspx?doc=histologie-savoir-sur-etude-tissus-biologiques#:~:text=L'histologie%20est%20une%20branche,ailleurs%20parler%20d'anatomie%20microscopique.)
- [Web 5]. Cytologie [consulté le 22 février 2024]. Disponible sur
[:https://www.cdllaboratories.com/fr/services/women/cytology/](https://www.cdllaboratories.com/fr/services/women/cytology/).
- [Web 6]. Développement de la spectroscopie Raman en cytopathologie : Application au diagnostic de la leucémie lymphoïde chronique [consulté le 24 Février 2024]. Disponible sur
[:https://www.theses.fr/2018REIMM2028](https://www.theses.fr/2018REIMM2028).
- [Web 7]. Le tissu sanguin [consulté le 24 Février 2024]. Disponible sur :https://fac.umc.edu.dz/vet/Cours_Ligne/cours_20_21/Histologie_A1/Tissu_sanguin.pdf.
- [Web 8]. La composition du sang [consulté le 24 Février 2024]. Disponible sur
[:https://fr.khanacademy.org/science/biologie-a-l-ecole/x5047ff3843d876a6:1er-degre-sciences-de-base/x5047ff3843d876a6:bio-1er-degre-les-systemes-circulatoire-et-pulmonaire/a/components-of-the-blood](https://fr.khanacademy.org/science/biologie-a-l-ecole/x5047ff3843d876a6:1er-degre-sciences-de-base/x5047ff3843d876a6:bio-1er-degre-les-systemes-circulatoire-et-pulmonaire/a/components-of-the-blood).
- [Web 9]. Comment réaliser un frottis sanguin? [consulté le 28 février 2024]. Disponible sur
[:https://www.lapvso.com/comment-realiser-un-frottis-sanguin/](https://www.lapvso.com/comment-realiser-un-frottis-sanguin/).

- [Web 10]. Coloration de May-Grünwald-Giemsa en hématologie [consulté le 28 février 2024]. Disponible sur :https://www.humeau.com/media/blfa_files/ME_Protocoles-utilisation-hematologie_34332007054-34332007060_FR_250516.pdf
- [Web 11]. Les lymphocytes : rôles, pathologies, traitements [consulté le 28 Février 2024]. Disponible sur :<https://www.passeportsante.net/fr/parties-corps/Fiche.aspx?doc=lymphocytes>.
- [Web 12]. Notions de base sur l'apprentissage profond [consulté le 5 Mars 2024]. Disponible sur :<https://rpubs.com/alexvezeau/SCI1035-CH1>.
- [Web 13]. Système expert [consulté le 5 Mars 2024]. Disponible sur :<https://www.gladir.com/CODER/IA/systeme-expert.htm>.
- [Web 14]. Régression linéaire [consulté le 6 Mars 2024]. Disponible sur :https://fr.wikipedia.org/wiki/R%C3%A9gression_lin%C3%A9aire.
- [Web 15]. Régression logistique [consulté le 6 Mars 2024]. Disponible sur :<https://spss.espaceweb.usherbrooke.ca/regression-logistique/>.
- [Web 16]. Logistic Regression: From Intuition to Sigmoid function [consulté le 6 Mars 2024]. Disponible sur :<https://medium.com/@mkk.rakesh/logistic-regression-from-intuition-to-sigmoid-function-b9b977c8f97d>.
- [Web 17]. SVM [consulté le 6 Mars 2024]. Disponible sur :
[https://dataanalyticspost.com/Lexique/svm#:~:text=SVM%20\(Support%20Vector%20Machine%20ou,ou%20de%20d%C3%A9tection%20d'anomalie](https://dataanalyticspost.com/Lexique/svm#:~:text=SVM%20(Support%20Vector%20Machine%20ou,ou%20de%20d%C3%A9tection%20d'anomalie).
- [Web 18]. KMeans Clustering for Classification [consulté le 6 Mars 2024]. Disponible sur:<https://towardsdatascience.com/kmeans-clustering-for-classification-74b992405d0a>.
- [Web 19]. APPRENTISSAGE PAR RENFORCEMENT [consulté le 6 Mars 2024]. Disponible sur :
<https://dataanalyticspost.com/Lexique/apprentissage-par-renforcement/>.
- [Web 20]. Deep Learning Spreads [consulté le 6 Mars 2024]. Disponible sur :
<https://semiengineering.com/deep-learning-spreads/>.
- [Web 21]. Les réseaux de neurones convolutifs [consulté le 7 Mars 2024]. Disponible sur :
<https://history.hashnode.dev/les-reseaux-de-neurones-convolutifs>.
- [Web 22]. LES RÉSEAUX DE NEURONES CONVOLUTIFS [consulté le 7 Mars 2024]. Disponible sur :
<https://www.natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-reseaux-de-neurones-convolutifs>.
- [Web 23]. Mieux comprendre le deep learning appliqué à la reconnaissance d'images [consulté le 10 Mars 2024]. Disponible sur :
<https://france.devoteam.com/paroles-dexperts/mieux-comprendre-le-deep-learning-applique-a-la-reconnaissance-dimages/>.
- [Web 24]. Comprendre les Réseaux de Neurones Convolutifs (CNN) [consulté le 10 Mars 2024]. Disponible sur :
<https://yannicksergeobam.medium.com/comprendre-les-r%C3%A9seaux-de-neurones-convolutifs-cnn-d5f14d963714>.
- [Web 25]. Couche de Convolution: Tout savoir sur les CNN – Meilleur Guide [consulté le 11 Mars 2024]. Disponible sur :
<https://inside-machinelearning.com/cnn-couche-de-convolution/>.
- [Web 26]. Mish vs ReLU : quelle est la meilleure fonction d'activation ? [consulté le 11 Mars 2024]. Disponible sur :
<https://penseeartificielle.fr/mish-vs-relu-meilleure-fonction-activation/>.

- [Web 27]. Difference between AlexNet and GoogleNet [consulté le 12 Mars 2024]. Disponible sur :<https://www.geeksforgeeks.org/difference-between-alexnet-and-googlenet/>.
- [Web 28]. Evaluation of cancer outcome assessment using MRI: A review of deep-learning methods [consulté le 12 Mars 2024]. Disponible sur :<https://www.birpublications.org/doi/10.1259/bjro.20210072>.
- [Web 29]. Illustrated: 10 CNN Architectures [consulté le 13 Mars 2024]. Disponible sur :<https://connect.aisingapore.org/2019/09/illustrated-10-cnn-architectures/>.
- [Web 30]. Recurrent Neural Network (RNN) : de quoi s'agit-il ? [consulté le 13 Mars 2024]. Disponible sur :<https://datascientest.com/recurrent-neural-network#:~:text=Les%20r%C3%A9seaux%20de%20neurones%20r%C3%A9currents,parole%20ou%20des%20s%C3%A9ries%20temporelles>.
- [Web 31]. Deep Learning : les réseaux de neurones récurrents [consulté le 13 Mars 2024]. Disponible sur :<https://datavalue-consulting.com/deep-learning-reseaux-neurones-recurrents-rnn/>.
- [Web 32]. RNN or Recurrent Neural Network for Noobs [consulté le 13 Mars 2024]. Disponible sur :<https://medium.com/hackernoon/rnn-or-recurrent-neural-network-for-noobs-a9afbb00e860>.
- [Web 33]. Les Autoencoders – modèles d'apprentissage non supervisé [consulté le 13 Mars 2024]. Disponible sur :<https://datascientest.com/les-autoencoders-modeles-dapprentissage-non-supervise>.
- [Web 34]. Autoencoder [consulté le 13 Mars 2024]. Disponible sur :<https://fr.mathworks.com/discovery/autoencoder.html>.
- [Web 35]. L'apprentissage par transfert : qu'est-ce que c'est ? [consulté le 20 Mars 2024]. Disponible sur :<https://datascientest.com/en/transfer-learning-what-is-it>.
- [Web 36]. Classification Binaire [consulté le 16 Avril 2024]. Disponible sur :https://datafranca.org/wiki/Classification_binaire#:~:text=D%C3%A9finition,non%20une%20propri%C3%A9t%C3%A9%20%2F%20fonctionnalit%C3%A9%20donn%C3%A9e.
- [Web 37]. Logistic Regression pour Machine Learning – Une Introduction Simple [consulté le 16 Avril 2024]. Disponible sur :<https://mrmint.fr/logistic-regression-machine-learning-introduction-simple>.
- [Web 38]. Introduction au langage python [consulté le 1 Mai 2024]. Disponible sur :<https://kachou92.over-blog.com/2019/09/introduction-au-langage-python.html>.
- [Web 39]. TensorFlow : tout savoir sur la bibliothèque Machine Learning open source [consulté le 1 Mai 2024]. Disponible sur :<https://www.lebigdata.fr/tensorflow-definition-tout-savoir>.
- [Web 40]. TensorFlow [consulté le 1 Mai 2024]. Disponible sur :<https://quintagroup.com/cms/python/tensorflow>.
- [Web 41]. Keras, une bibliothèque pour construire un modèle simple de réseau de neurones [consulté le 1 Mai 2024]. Disponible sur : <https://www.lemondeinformatique.fr/les-dossiers/lire-keras-une-bibliotheque-pour-construire-un-modele-simple-de-reseau-de-neurones-1006.html>.
- [Web 42]. NumPy logo refresh #37 [consulté le 1 Mai 2024]. Disponible sur : <https://github.com/numpy/numpy.org/issues/>.

- [Web 43]. Scikit Learn logo [consulté le 1 Mai 2024]. Disponible sur :<https://logosdownload.com/scikit-learn>.
- [Web 44]. Matplotlib : la visualisation de données en Python [consulté le 1 Mai 2024]. Disponible sur :<https://www.intelligence-artificielle-school.com/ecole/technologies/matplotlib/>.
- [Web 45]. Créer des graphiques avec la librairie Matplotlib et Python [consulté le 1 Mai 2024]. Disponible sur :<https://blog-mes-outils-automatiques.fr/introduction-librairie-matplotlib-python/>.
- [Web 46]. Matplotlib logo [consulté le 1 Mai 2024]. Disponible sur :<https://matplotlib.org/3.6.0/gallery/misc/logos2.html>.
- [Web 47]. Comment utiliser PyGWalker avec Gradio [consulté le 1 Mai 2024]. Disponible sur :<https://docs.kanaries.net/fr/pygwalker/tutorials/use-pygwalker-in-gradio>.
- [Web 48]. Créer une interface pour vos modèles d'apprentissage automatique à l'aide de la bibliothèque Python Gradio [consulté le 1 Mai 2024]. Disponible sur :<https://www.analyticsvidhya.com/blog/2021/04/create-interface-for-your-machine-learning-models-using-gradio-python-library/>.
- [Web 49]. Guide du débutant sur l'implémentation de VGG16 dans Keras [consulté le 22 Mai 2024]. Disponible sur :<https://builtin.com/machine-learning/vgg16>.
- [Web 50]. Figure1: Architecture of InceptionV3 • INCEPTION-RESNETV2 [consulté le 25 Mai 2024]. Disponible sur : https://www.researchgate.net/figure/figure1-Architecture-of-InceptionV3-INCEPTION-RESNETV2-The-Inception-ResNetV2_fig1_376510348.