

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de fin de cycle
en vue de l'obtention du diplôme de master
en Mathématiques appliquées



جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Option : Science des données et aide à la décision

Thème :

**Machine Learning pour l'analyse des images médicales
Cas d'étude : CHU Béjaïa**

Présenté par :

**CHEBIHI Salma
CHIBANE Zebida**

Soutenu devant le jury composé de :

Dr. ASLI Larbi	Université de Béjaïa	Président
Dr. ELSAKAAN Nadim	Université de Béjaïa	Examineur
Dr. BOUDRIOUA Asma	Université de Béjaïa	Examinatrice
Dr. HAMZA Lamia	Université de Béjaïa	Encadrant
Mme. BAIR Narimene	Université de Béjaïa	Co-Encadrant

Année universitaire : 2023/2024

Remerciements

En premier lieu, nous remercions Allah, le Tout-Puissant, pour nous avoir donné la force et la patience pour accomplir ce travail jusqu'à son achèvement.

Nous tenons également à exprimer notre sincère gratitude à notre directrice de recherche, Dr **HAMZA Lamia**, pour son encadrement et son soutien pendant toute cette période pleine de défis. Sa guidance éclairée et sa patience ont été cruciales pour la réussite de ce projet.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs de l'Université, en particulier notre co-encadrant **Mme. BAIR Narimene** et aussi **Dr. ASLI Larbi** et **Dr. EL-SAKAAN Nadim** pour leur soutien et leurs conseils précieux qui ont été cruciaux pour notre progression.

Nous remercions également les membres du jury qui nous feront l'honneur d'évaluer notre travail, et de soulever leurs critiques constructives qui sont importantes pour but d'apporter un plus à notre projet.

Nous adressons également nos remerciements à toutes les personnes qui nous ont soutenus, de près ou de loin, tout au long de la réalisation de ce projet.

Merci à tous.

Dédicaces

Je dédie ce travail :

À **mes parents**, pour leur amour, leur soutien interminable et les sacrifices qu'ils ont consentis pour m'offrir les meilleures opportunités d'éducation. Votre force et votre dévouement sont les fondations de tout ce que j'entreprends.

À **ma soeur** et à **mon frère**, pour leur soutien intensif tout au long de mes études.

À ma binôme **CHIBANE Zebida**, merci pour ta patience et ton soutien dans toutes les difficultés que nous avons rencontrées, que Dieu t'accorde tout ce que tu souhaites.

À ma cousine **Ahlem**, à ma meilleure amie, **BOUDJAOUI Imene** et à mes amis, **ABDOUNE Sarah**, **OUGUERGOUZ Mehdi**, **Redjal Zineddine**, **HAMADI liza**, **MEBARKI aida** et **CHIKH AMER Chiraz** pour leur amitié sincère et les encouragements répétés. Vous avez rendu chaque obstacle plus surmontable et chaque moment en un souvenir précieux.

Et à tous ceux qui m'aiment et que j'aime, de près ou de loin, ont contribué dans mon mémoire de fin d'étude.

#FREE_PALESTINE

CHEBIHI Salma.

Dédicaces

Je remercie **Allah** pour tout et avant tout, grâce à lui j'ai pu réaliser ce travail, que je dédie :

À **mes très chers parents** qui m'ont soutenu durant toutes mes études, qui ont fait de moi ce que je suis aujourd'hui je leur serais éternellement reconnaissante.

À **mes frères** Toufik et Sofiane, pour leur aide précieuse, à **ma soeur** Meriem et son mari Khelaf, pour leur amour, leur soutien indéfectible, et leurs précieux conseils.

À **ma binôme T. Salma**, je te remercie pour ta patience, ton encouragement, ton soutien tout au long de ce projet, et aussi pour ton amitié.

À mes cousines, à ma meilleure amie Kamelia, pour être toujours là à mes côtés, et à mes amis, Mehdi, Zineddine, Djidji, Dabi, Ikram, Belkacem et Fadi, pour leur amitié sincère et leur encouragement.

À tous ceux qui m'ont soutenu de près ou de loin, Merci.

#FREE_PALESTINE

CHIBANE Zebida.

Table des matières

Remerciements	I
Dédicaces	II
Dédicaces	III
Liste des figures	VII
Liste des tables	VIII
Liste des abréviations et notations	IX
Introduction générale	1
1 Contexte et terminologies	3
Introduction	3
1.1 Présentation de l'organisme d'accueil CHU	4
1.2 Services du CHU de Béjaïa	4
1.3 Organigramme du CHU	5
1.4 Concepts de base	6
1.4.1 Imagerie médicale	6
1.4.2 Pneumonie	7
1.4.3 Intelligence artificielle	7
1.4.4 Machine Learning	8
1.4.5 Algorithmes de ML utilisés dans notre projet	9
1.4.6 Deep Learning	16
1.4.7 Réseau neuronal convolutif	16
1.5 Optimisation et évaluation des modèles	18
1.5.1 GridSearch	18
1.5.2 Validation croisée	18
1.6 Métriques d'évaluation	18
1.6.1 Exactitude (Accuracy)	19
1.6.2 Précision (Precision)	19
1.6.3 Rappel (Recall)	19
1.6.4 F1 Score	19
1.6.5 Spécificité (Specificity)	20
1.6.6 Le taux de faux positifs (FPR)	20
1.7 Matrice de Confusion	20
1.8 La courbe de ROC et AUC	20
1.9 Problématique	21
1.10 Solution proposée	21
Conclusion	21

2	Travaux connexes	22
	Introduction	22
2.1	Modèles de CNN	23
2.1.1	Modèle VGG16	23
2.1.2	Modèle VGG19	24
2.2	Travaux connexes	25
2.2.1	Méthodes basées sur VGG	25
2.2.2	Méthodes basées sur des modèles ensemblistes ou d'autres architectures CNN	26
2.2.3	Méthodes basées sur des approches spécifiques ou non traditionnelles	27
2.3	Analyse comparative	29
	Conclusion	30
3	Méthodes et résolutions	31
	Introduction	31
3.1	Notre approche	32
3.2	Configuration de l'environnement de travail	33
3.3	Collecte de données	33
3.4	Chargement des données	33
3.5	Prétraitement des données	34
3.5.1	Redimensionnement des images	34
3.5.2	Conversion en niveaux de gris	34
3.5.3	Normalisation	34
3.5.4	Création des étiquettes	34
3.5.5	Exploration des données	34
3.6	Visualisation des données	34
3.7	Processus d'extraction des caractéristiques	35
3.7.1	Chargement des modèles VGG16 et VGG19	36
3.7.2	Prétraitement des images	36
3.7.3	Définition de la fonction d'extraction des caractéristiques	36
3.7.4	Chargement et prétraitement des images	36
3.7.5	Combinaison des images et des étiquettes	36
3.7.6	Extraction des caractéristiques pour toutes les images	36
3.8	Processus d'entraînement des modèles de ML	36
3.8.1	Préparation des données	37
3.8.2	Sélection des modèles	37
3.8.3	La recherche des hyperparamètres	38
3.8.4	Résultat de recherche des hyperparamètres	39
3.8.5	Interprétation des paramètres optimaux	39
3.8.6	Création et entraînement des modèles	41
3.8.7	Évaluation et validation des modèles	42
3.8.8	Sauvegarde des modèles	42
3.8.9	Prédiction sur de nouvelles données	42
	Conclusion	42

4 Résultats et évaluation	43
Introduction	43
4.1 Environnement de développement	44
4.1.1 Langage de Programmation	44
4.1.2 Plateforme de Développement	44
4.1.3 Bibliothèques utilisées	44
4.2 Évaluation des modèles de ML avec VGG16	45
4.2.1 Matrices de confusion	46
4.2.2 Analyse des courbes ROC	47
4.2.3 Analyse des autres métriques de performance	49
4.3 Évaluation des modèles de ML avec VGG19	50
4.3.1 Matrices de confusion	50
4.3.2 Analyse des courbes ROC	51
4.3.3 Analyse des autres métriques de performance	52
4.4 Comparaison entre les deux architecture VGG16 et VGG19	53
4.5 Synthèse	54
Conclusion	55
Conclusion générale	56
Bibliographie	61
Annexes	62
Résumé	67

Liste des figures

1.1	Logo du CHU de Béjaïa	4
1.2	Organigramme du CHU Béjaïa	5
1.3	L'intelligence artificielle	7
1.4	Machine Learning	8
1.5	Exemple du KNN	10
1.6	Exemple du Random Forest	12
1.7	Exemple du SVM	13
1.8	Exemple du Gradient Boosting	15
1.9	Exemple de la structure du Deep Learning	16
1.10	Exemple de la structure d'un CNN	17
1.11	Processus de validation croisée [14]	18
2.1	Exemple de la structure d'un VGG16 [26]	23
2.2	Exemple de la structure d'un VGG19	24
3.1	Schéma de proposition	32
3.2	Échantillon d'images pour chaque classe ('PNEUMONIA' et 'NORMAL')	35
3.3	Répartition des étiquettes ('PNEUMONIA' et 'NORMAL')	35
4.1	Matrices de confusion des différents algorithmes	46
4.2	Courbes ROC pour différents modèles (VGG16)	48
4.3	Comparaison des performances pour différents modèles (VGG16)	49
4.4	Matrices de confusion des différents algorithmes	50
4.5	Courbes ROC pour différents modèles (VGG19)	51
4.6	Comparaison des performances pour différents modèles (VGG19)	52
4.7	Comparaison de la précision des architectures VGG19 et VGG16	53
4.8	Comparaison des métriques des architectures VGG19 et VGG16	54
4.9	Architecture du cloud computing	65

Liste des tableaux

1.1	Un exemple de matrice de confusion	20
2.1	Tableau récapitulatif des travaux	29
3.1	Grille de recherche des hyperparamètres pour différents modèles	38
3.2	Paramètres optimaux de chaque algorithme ML avec VGG16 et VGG19	39

Liste des abréviations et notations

AUC	Area Under Curve
CHU	Centre Hospitalo Universitaire
CNN	Convolutional Neural Network
CT-scan	Computed Tomography Scan
DICOM	Digital Imaging and COmmunications in Medicine
DL	Deep Learning
GB	Gradient Boost
IA	Intelligence Artificielle
IaaS	Infrastructure as a Service
IRM	Imagerie par Résonance Magnétique
KNN	K-Nearest Neighbors
ML	Machine Learning
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
PaaS	Platform as a Service
RF	Random Forest
ROC	Receiver Operating Characteristic
SaaS	Software as a Service
SVM	Support Vector Machine
VGG	Visual Geometry Group

Introduction générale

Dans le domaine médical, la vitesse et la précision des diagnostics sont cruciales pour traiter efficacement les patients. Bien que l'imagerie médicale, telle que les radiographies et les scanners, soit essentielle pour détecter des maladies, les méthodes traditionnelles d'interprétation manuelle par des professionnels formés peuvent être lentes et sujettes à des erreurs, notamment en cas de fatigue ou de surcharge de travail. Avec les progrès technologiques récents, l'intelligence artificielle (IA) offre de nouvelles possibilités pour améliorer et accélérer le processus diagnostique.

Le Centre Hospitalo-Universitaire (CHU) de Béjaïa fait face à des défis particuliers, notamment en ce qui concerne le diagnostic de la pneumonie. Les limitations liées à la gestion et au stockage des images, spécifiquement des images au format DICOM (Digital Imaging and Communications in Medicine), ainsi que les délais d'interprétation, peuvent retarder les diagnostics et, par conséquent, le traitement. Cela souligne la nécessité d'explorer des solutions innovantes pour surmonter ces obstacles.

La problématique centrale de ce mémoire est de déterminer comment améliorer les capacités diagnostiques au CHU de Béjaïa en termes de précision et de rapidité. Les radiologues du CHU sont confrontés à un volume élevé de cas à diagnostiquer, ce qui peut entraîner des retards significatifs, surtout lors de la saison grippale. De plus, la fatigue et la charge de travail élevée peuvent augmenter le risque d'erreurs dans l'interprétation des images radiographiques. Par conséquent, notre recherche vise à surpasser les limites des méthodes diagnostiques actuelles en introduisant des technologies avancées de Machine Learning. L'objectif est d'automatiser une partie du processus de diagnostic de la pneumonie, afin d'analyser et de classer les images radiographiques de manière plus précise et efficace. Cette démarche permettrait de réduire la dépendance à l'interprétation manuelle et augmenter la cohérence des diagnostics.

Nous proposons d'explorer des modèles de Machine Learning, en particulier les réseaux de neurones convolutifs (CNN). Nous comparons deux architectures, VGG16 et VGG19, qui ont été choisies en raison de leurs efficacités et leurs performances prometteuses dans la reconnaissance d'images à grande échelle, notamment en obtenant de bons résultats dans des compétitions telles que l'ILSVRC (ImageNet Large-Scale Visual Recognition Challenge). Ces modèles ont démontré une capacité à extraire efficacement les caractéristiques importantes des images, ce qui les rend adaptés à des tâches de classification d'images médicales telles que la détection de la pneumonie à partir de radiographies thoraciques. En combinant ces architectures avec divers algorithmes de Machine Learning, nous cherchons à déterminer la combinaison la plus efficace pour but de classer correctement les images et accélérer le processus de diagnostic.

Notre approche méthodologique englobe la collecte et le prétraitement des images radiographiques, incluant leur normalisation et redimensionnement des images, pour uniformiser les entrées. Nous appliquons par la suite, la combinaison des deux modèles de réseaux de neurones convolutifs (CNN) avec des algorithmes de Machine Learning. Cette dernière est évaluée en utilisant des métriques telles que la précision, le rappel, et le score F1 pour juger son efficacité.

Notre étude est structurée en quatre chapitres principaux :

1. **Le premier chapitre** présente le contexte et les terminologies essentielles, incluant une présentation du CHU de Béjaïa, les concepts de base de l'imagerie médicale, de l'Intelligence Artificielle et du Machine Learning. Il aborde également les métriques d'évaluation et le cloud computing.
2. **Le deuxième chapitre** se concentre sur la modélisation et les travaux connexes, en comparant notamment les modèles de CNN VGG16 et VGG19, ainsi que diverses méthodes basées sur ces architectures.
3. **Le troisième chapitre** détaille notre approche méthodologique, incluant la collecte, le prétraitement des données, l'extraction des caractéristiques, et l'utilisation de divers algorithmes de machine learning .
4. Enfin, **le quatrième chapitre** présente les résultats et l'évaluation de nos modèles, incluant l'environnement de développement utilisé et une analyse comparative détaillée des performances des différentes approches.

1

Contexte et terminologies

Sommaire

Introduction	3
1.1 Présentation de l'organisme d'accueil CHU	4
1.2 Services du CHU de Béjaïa	4
1.3 Organigramme du CHU	5
1.4 Concepts de base	6
1.5 Optimisation et évaluation des modèles	18
1.6 Métriques d'évaluation	18
1.7 Matrice de Confusion	20
1.8 La courbe de ROC et AUC	20
1.9 Problématique	21
1.10 Solution proposée	21
Conclusion	21

Introduction

Ce premier chapitre introduit le Centre Hospitalier Universitaire de Béjaïa (CHU), soulignant les défis spécifiques rencontrés dans le domaine de l'imagerie médicale et du diagnostic de la pneumonie. Nous examinons les possibilités offertes par des technologies avancées comme l'intelligence artificielle et le cloud computing pour surmonter ces problématiques. De plus, nous définissons les concepts et les métriques d'évaluation essentiels pour comprendre ces technologies, établissant les bases nécessaires aux développements futurs de cette étude.

1.1 Présentation de l'organisme d'accueil CHU

Le secteur sanitaire de Béjaïa comprend plusieurs structures de santé, notamment l'Hôpital Khellil Amrane situé dans la cité Smina, l'Hôpital Frantz Fanon dans l'ancienne ville de Bejaïa, à côté de Bordj Moussa, et l'Hôpital Targa Ouzamour (clinique mère-enfant).

Ce secteur est géré par l'Hôpital Khellil Amrane, situé au centre de la commune de Béjaïa, qui dispose d'un budget de fonctionnement et d'une autonomie de gestion. Mis en service en 1991, cet hôpital a marqué un tournant dans l'amélioration des services de santé de la région. Suite à l'ouverture de la faculté de médecine, l'Hôpital Khellil Amrane est devenu le siège du Centre Hospitalo-Universitaire (CHU) de Béjaïa en 2011 [1]. La figure 1.1 représente le CHU de Béjaïa.



FIGURE 1.1 – Logo du CHU de Béjaïa

1.2 Services du CHU de Béjaïa

Le CHU de Béjaïa est chargé de missions de diagnostic, d'exploration, de soins, de prévention, de formation, d'études et de recherche, en collaboration avec les établissements d'enseignement supérieur en sciences médicales. L'hôpital offre une gamme complète de services médicaux, incluant la cellule d'accueil et d'orientation des cancéreux, l'anesthésie, la réanimation, la chirurgie générale, la médecine interne, le bloc opératoire central, le laboratoire central, la pédiatrie, la cardiologie, la neurochirurgie, l'orthopédie traumatologie, les urgences médicales chirurgicales et l'imagerie médicale [1].

1.3 Organigramme du CHU

La figure 1.2 représente l’organigramme du CHU, divisé en plusieurs niveaux de gestion et de spécialisations :

- **Niveau supérieur** : Composé du comité consultatif, du Directeur général, du conseil scientifique et du secrétariat général jouent un rôle essentiel dans la direction stratégique et la définition des politiques globales du CHU.
- **Gestion des hôpitaux** : Hôpital Frontz Fanon, Hôpital Khelil Amrane, et Hôpital Targua Ouzemour.
- **Directions spécialisées** : Incluent les activités médicales et paramédicales, les ressources humaines, les finances et contrôle, et les moyens matériels.
- **Services spécialisés** : Chaque direction englobe divers services spécifiques comme la gestion des activités médicales, la pharmacie, la gestion administrative des patients, la formation, le personnel, la comptabilité et le contrôle des coûts, ainsi que la maintenance des équipements.
- **Service imagerie médicale** : Ce Service est rattaché à l’Hôpital Khelil Amrane sous la direction des activités médicales et paramédicales, est crucial pour les diagnostics médicaux.

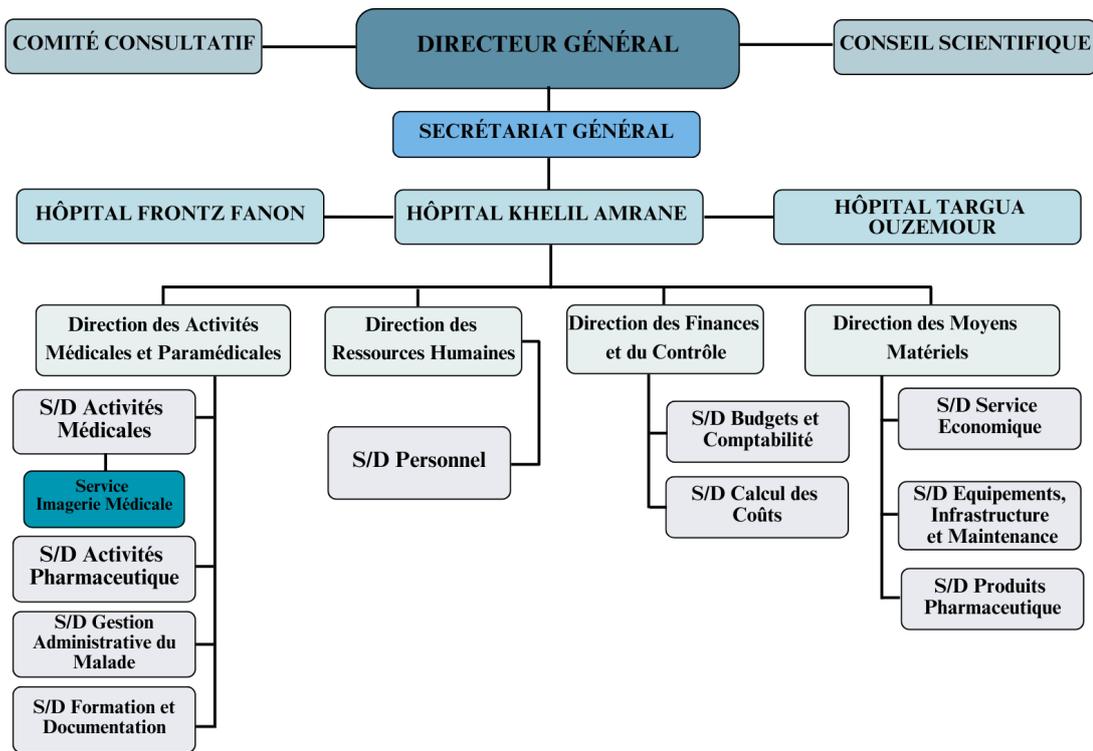


FIGURE 1.2 – Organigramme du CHU Béjaïa

1.4 Concepts de base

Dans cette section nous présentons les principes fondamentaux de l'imagerie médicale, les caractéristiques essentielles de la pneumonie, le Machine Learning, le Deep Learning et le Cloud Computing.

1.4.1 Imagerie médicale

L'imagerie médicale est une spécialité médicale qui consiste à produire des images du corps humain et à les interpréter à des fins diagnostiques, thérapeutiques et de suivi de l'évolution des pathologies [2]. Les avancées technologiques comme la tomographie assistée par ordinateur (**CT-scan : Computed Tomography Scan**), imagerie par résonance magnétique (**IRM : Imagerie par Résonance Magnétique**) et les rayons X (**X-ray**) ont rendu l'analyse de ces images plus complexe. Alors qu'elle était auparavant réalisée par des experts humains, les progrès du Machine Learning (ML) et du Deep Learning (DL) permettent désormais de développer des systèmes d'aide au diagnostic assistés par ordinateur, améliorant la précision et l'efficacité de l'analyse des images médicales.

a. Radiographie par Rayons X :

La radiographie par rayons X est l'une des techniques d'imagerie médicale les plus fréquemment utilisées. Elle repose sur l'utilisation de rayonnements électromagnétiques pour visualiser l'intérieur du corps humain. Quand les rayons X traversent le corps, ils forment une image en deux dimensions et les différents tissus absorbent les rayons X de manière différente selon leur densité. En raison de la densité élevée des os, ils absorbent davantage de rayons X et sont visibles en blanc sur la radio, tandis que les tissus mous apparaissent en gris et l'air en noir [3].

La radiographie est utilisée dans plusieurs tâches, telles que le diagnostic des fractures, l'identification des tumeurs, l'évaluation dentaire, et la détection des infections pulmonaires.

b. Images chest x-ray :

La radiographie du thorax est l'examen radiographique diagnostique le plus couramment pratiqué. Elle permet d'obtenir des images du cœur, des poumons, des voies respiratoires, des vaisseaux sanguins et des os de la colonne vertébrale et du thorax [4].

c. CT-Scan :

La tomographie assistée par ordinateur est une technique d'imagerie puissante qui utilise les rayons X et des algorithmes informatiques avancés pour créer des images transversales détaillées du corps humain [5].

d. IRM :

L'IRM est une technique non invasive qui permet d'observer l'anatomie des organes, l'élasticité des tissus, et d'autres caractéristiques internes du corps sans nécessiter l'injection de produit de contraste. Elle est particulièrement précieuse pour le diagnostic et le suivi de diverses pathologies [6].

1.4.2 Pneumonie

La pneumonie est une infection des poumons qui touche spécifiquement les alvéoles pulmonaires. Elle peut être provoquée par différents agents infectieux tels que des bactéries, des virus ou des champignons. Les symptômes typiques incluent la toux, la fièvre, des frissons, une respiration difficile, et dans certains cas, une douleur thoracique exacerbée par la toux ou la respiration profonde [7].

1.4.2.1 Diagnostic de la pneumonie

Le diagnostic de la pneumonie repose principalement sur l'imagerie médicale, en particulier la radiographie pulmonaire, qui permet de détecter des infiltrats ou de zones de consolidation au niveau des poumons, signes typiques de la pneumonie. Bien que d'autres tests complémentaires puissent être réalisés, tels qu'un examen clinique approfondi, des analyses sanguines et des cultures d'expectorations pour identifier l'agent pathogène responsable, l'interprétation de la radiographie pulmonaire est considéré comme l'examen de première intention et l'élément clé pour confirmer la présence d'une pneumonie [8].

1.4.2.2 Relation entre la pneumonie et les images chest X-ray

La radiographie thoracique (chest X-ray) joue un rôle crucial dans le diagnostic de la pneumonie, qui se manifeste par une inflammation des sacs aériens (poumons). Grâce aux images X-ray du thorax, il est possible de voir les opacités pulmonaires, qui sont des zones anormales de densité élevée signalant la présence de liquide, de cellules inflammatoires, etc. Ces opacités, généralement représentées par des taches blanches ou des zones obscurcies, permettent aux médecins de localiser et de déterminer l'étendue de l'infection, ce qui permet de distinguer la pneumonie d'autres maladies pulmonaires telles que l'œdème pulmonaire ou les tumeurs [9].

1.4.3 Intelligence artificielle

L'intelligence artificielle est le domaine de l'informatique qui permet de développer des systèmes capables d'accomplir des tâches qui requièrent normalement l'intelligence humaine. Elle intègre des méthodes telles que le Machine Learning et le Deep Learning pour automatiser la prise de décision [10]. La figure 1.3 représente un schéma illustratif des relations entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond.

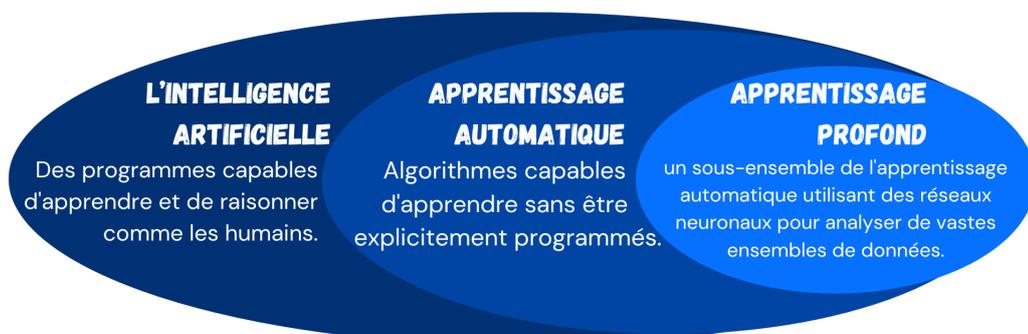


FIGURE 1.3 – L'intelligence artificielle

1.4.4 Machine Learning

Le Machine Learning (ML) équipe les systèmes informatiques avec la capacité d'apprendre à partir de données et de faire des prédictions ou prendre des décisions sans être programmés de manière explicite pour chaque tâche. Il s'appuie sur différents types d'algorithmes statistiques pour identifier des patterns dans les données et ajuster les actions en conséquence, notamment l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [10]. Ces algorithmes permettent aux systèmes d'évoluer et de s'améliorer de façon autonome au fur et à mesure qu'ils sont exposés à plus de données.

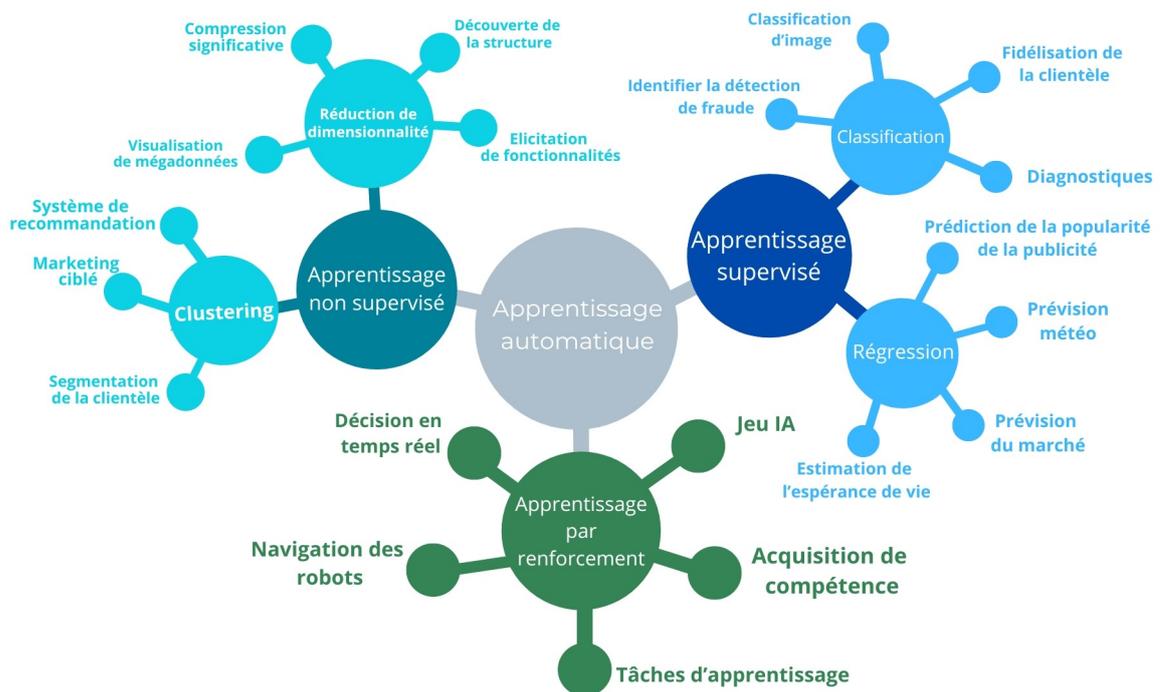


FIGURE 1.4 – Machine Learning

1.4.4.1 Modèles d'apprentissage

Il existe plusieurs types de modèles d'apprentissage automatique, chacun étant adapté à différents types de tâches et de données, à savoir : les modèles d'apprentissage supervisé, non supervisé et par renforcement (voir la figure 1.4).

1. Modèles d'apprentissage supervisé

Ces modèles utilisent des données étiquetées, où la sortie correcte est connue pour chaque exemple. Le modèle apprend à associer les données d'entrée à ces sorties correctes. L'apprentissage supervisé est utilisé pour :

- **Classification** : Déterminer la catégorie à laquelle une nouvelle observation appartient, comme filtrer les e-mails en "spam" ou "non spam".
- **Régression** : Estimer les relations entre les variables, utile pour la prédiction et la prévision.

2. Modèles d'apprentissage non supervisé

Ces modèles travaillent avec des données non étiquetées et doivent découvrir la structure sous-jacente des données par eux-mêmes. Ils sont utilisés pour :

- **Clustering** : Identifier des groupes ou des structures dans les données.
- **Réduction de la dimensionnalité** : Simplifier les données en réduisant le nombre de variables tout en conservant l'essentiel de l'information.

3. Modèles d'apprentissage par renforcement

Ces modèles apprennent par interaction avec leur environnement, recevant des récompenses ou des punitions. Le but est de maximiser la récompense cumulée, et ils sont couramment utilisés dans :

- **Jeux** : Apprendre à jouer et à gagner.
- **Contrôle de robots** : Apprendre à effectuer des tâches spécifiques.

1.4.5 Algorithmes de ML utilisés dans notre projet

Dans cette section, nous présentons les algorithmes que nous allons utiliser pour réaliser notre solution.

1. Naïve Bayes

Le classificateur Naïve Bayes est un algorithme de classification basé sur le théorème de Bayes, avec l'hypothèse d'indépendance entre les caractéristiques. Il permet de prédire une classe/catégorie, en se basant sur un ensemble donné de caractéristiques, en utilisant la probabilité. Malgré sa simplicité, le classificateur est étonnamment efficace et est fréquemment employé car il surpasse les méthodes de classification plus complexes [11].

Fonctionnement :

- **Estimation des probabilités à priori** : Calculer la probabilité à priori $P(C_k)$ pour chaque classe C_k .
- **Estimation des probabilités conditionnelles** : Calculer la probabilité conditionnelle $P(x_i|C_k)$ pour chaque caractéristique x_i donnée la classe C_k .
- **Utilisation du paramètre `var_smoothing`** : Ajouter une petite valeur à la variance de chaque caractéristique pour stabiliser les calculs et éviter les divisions par zéro.
- **Application du théorème de Bayes** : Calculer la probabilité à posteriori $P(C_k|\mathbf{x})$ pour chaque classe étant donné une observation \mathbf{x} .
- **Attribution de la classe** : Attribuer la classe avec la probabilité à posteriori la plus élevée.

Paramètres principaux :

- **`var_smoothing`** : Ce paramètre ajoute une petite valeur à la variance de chaque caractéristique pour stabiliser les calculs. Il permet d'éviter les divisions par zéro et les valeurs très petites, ce qui peut améliorer la robustesse du modèle.

Formule mathématique :

Le théorème de Bayes est exprimé comme suit :

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k) \cdot P(C_k)}{P(\mathbf{x})} \quad (1.1)$$

Pour le classificateur Naive Bayes, la probabilité à posteriori pour une classe C_k est proportionnelle au produit des probabilités à priori de la classe et des probabilités conditionnelles des caractéristiques données de la classe :

$$\mathbf{P}(C_k | \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \propto \mathbf{P}(C_k) \prod_{i=1}^n \mathbf{P}(x_i | C_k) \quad (1.2)$$

La classe C_k prédite est celle qui maximise cette probabilité.

2. KNN

Le K-Nearest Neighbor (KNN) est un algorithme de classification et de régression. Il détermine la probabilité qu'un point de données appartienne à un groupe [12], en fonction de la majorité des k voisins les plus proches, en utilisant une mesure de distance comme la distance euclidienne.

Fonctionnement :

- **Choix du nombre de voisins k** : Déterminer le nombre de voisins à considérer.
- **Calcul de la distance** : Calculer la distance entre l'instance à prédire et toutes les instances d'entraînement. La distance euclidienne est couramment utilisée :

$$\mathbf{d}(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^n (x_j - x_{ij})^2} \quad (1.3)$$

- **Sélection des k voisins les plus proches** : Identifier les k instances avec les plus petites distances.
- **Vote majoritaire** : Attribuer la classe la plus fréquente parmi les k voisins pour la classification, ou calculer la moyenne des valeurs pour la régression.

La figure 1.5 illustre un exemple du fonctionnement du KNN :

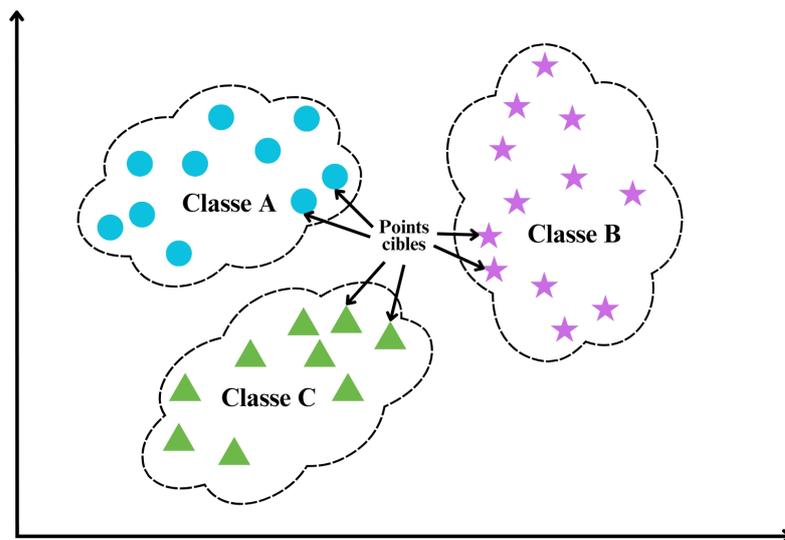


FIGURE 1.5 – Exemple du KNN

Paramètres principaux :

- **n_neighbors** : Nombre de voisins à considérer pour la classification. Ce paramètre détermine combien de voisins seront utilisés pour classer un point donné.
- **weights** : Méthode pour pondérer les voisins. Les options incluent 'uniform' (tous les voisins ont le même poids) et 'distance' (les voisins les plus proches ont plus de poids).
- **metric** : Métrique de distance utilisée pour identifier les voisins les plus proches (par exemple, 'euclidean', 'manhattan').

Formule mathématique :

Pour la classification, l'étiquette prédite \hat{y} est :

$$\hat{y} = \operatorname{argmax}_c \sum_{i \in \mathcal{N}_k(x)} \mathbf{1}(y_i = c) \quad (1.4)$$

où $\mathcal{N}_k(x)$ représente les k voisins les plus proches de x , et $\mathbf{1}(\cdot)$ est l'indicatrice.

Pour la régression, la valeur prédite \hat{y} est :

$$\hat{y} = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i \quad (1.5)$$

3. Forêts Aléatoires

Les forêts aléatoires, également appelées « forêts à décision aléatoire », sont une approche d'apprentissage qui combine différents algorithmes afin d'obtenir des résultats remarquables pour la classification, la régression et d'autres tâches et contrôler le sur-ajustement.

L'algorithme débute en créant un arbre de décision (voir **Annexe A**) et en enregistrant une entrée en haut. Il parcourt ensuite l'arborescence, les données étant segmentées en ensembles de plus en plus petits, en fonction de variables spécifiques [12].

Fonctionnement :

- **Bootstrap aggregation (Bagging)** : Random Forest utilise la technique de bagging pour créer plusieurs sous-ensembles de données d'entraînement par échantillonnage avec remplacement.
- **Construction des arbres de décision** : Un arbre de décision est construit pour chaque sous-ensemble en utilisant une sélection aléatoire de caractéristiques.
- **Agrégation des prédictions** : Les prédictions de tous les arbres sont combinées pour produire une prédiction finale. Pour la classification, c'est la classe majoritaire qui est choisie, et pour la régression, c'est la moyenne des prédictions.

La figure 1.6 illustre un exemple du fonctionnement des forêts aléatoires :

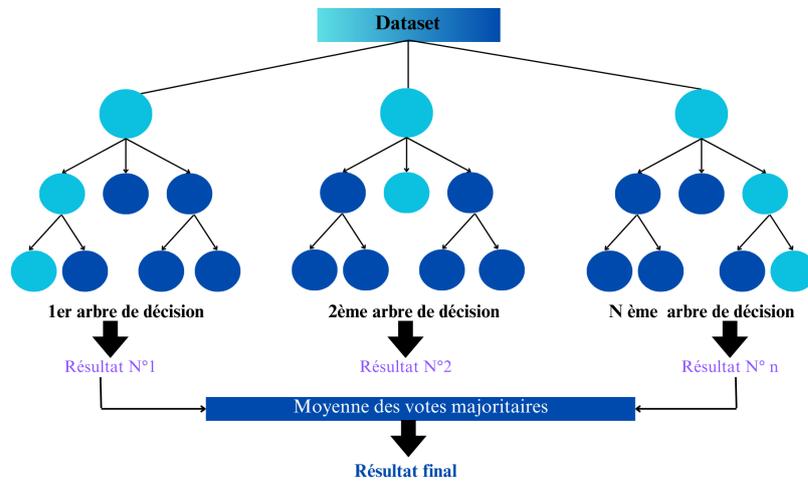


FIGURE 1.6 – Exemple du Random Forest

Paramètres principaux :

- **n_estimators** : Nombre d'arbres dans la forêt.
- **max_features** : Nombre maximum de caractéristiques considérées pour diviser un nœud.
- **max_depth** : Profondeur maximale des arbres. Un plus grand nombre rend le modèle plus complexe.
- **criterion** : Mesure de la qualité d'une division (par exemple, 'gini' pour l'indice de Gini, 'entropy' pour le gain d'information).

Formule mathématique :

Pour la classification, la prédiction finale \hat{y} est déterminée par le vote majoritaire des B arbres :

$$\hat{y} = \operatorname{argmax}_c \sum_{b=1}^B \mathbf{1}(h_b(\mathbf{x}) = c) \quad (1.6)$$

où $h_b(x)$ est la prédiction du b -ième arbre et $\mathbf{1}(\cdot)$ est la fonction indicatrice.

Pour la régression, la prédiction finale \hat{y} est la moyenne des prédictions des B arbres :

$$\hat{y} = \frac{1}{B} \sum_{b=1}^B h_b(x) \quad (1.7)$$

4. Machine à Vecteurs de Support

Les algorithmes Support Vector Machine (SVM) sont des modèles d'apprentissage supervisé utilisés pour la classification et la régression. Ils cherchent à trouver un hyperplan optimal dans un espace multidimensionnel qui sépare de manière adéquate toutes les données tout en maximisant la marge entre l'hyperplan et les points de données les plus proches de chaque classe [13].

Fonctionnement :

- **Maximisation de la marge** : SVM construit un hyperplan qui maximise la distance entre les points de données les plus proches de chaque classe, ces points étant appelés vecteurs de support.
- **Noyaux** : Pour les données non linéairement séparables, SVM utilise des fonctions de noyau pour transformer l'espace des caractéristiques et trouver un hyperplan dans un nouvel espace transformé.

La figure 1.7 illustre un exemple du fonctionnement du SVM :

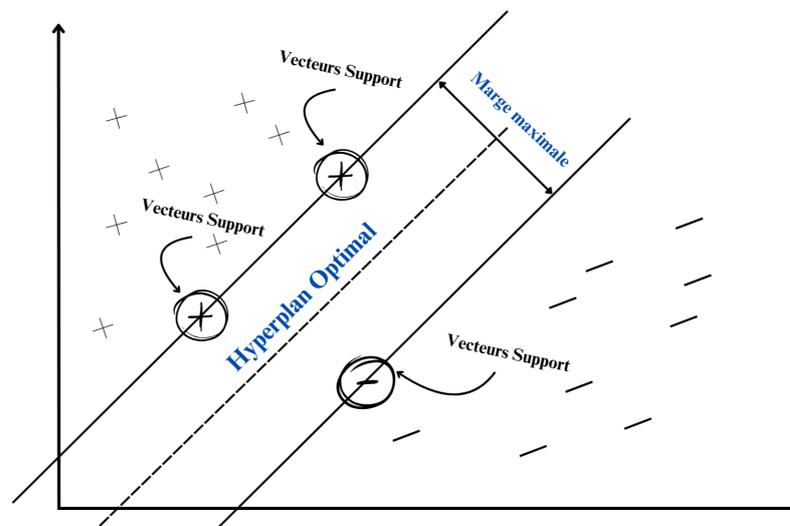


FIGURE 1.7 – Exemple du SVM

Paramètres principaux :

- **C** : Paramètre de régularisation qui contrôle la marge d'erreur admise sur les données d'entraînement. Des valeurs plus élevées de C cherchent à classer correctement toutes les données d'entraînement, au risque de sur-ajuster.
- **kernel** : Spécifie le type de noyau utilisé dans l'algorithme. Des exemples incluent `linear`, `poly` (polynomial), `rbf` (radial basis function), et `sigmoid`. Le choix du noyau permet de traiter des relations non-linéaires entre les caractéristiques.
- **gamma** : Coefficient de noyau pour les noyaux `rbf`, `poly` et `sigmoid`. Il détermine l'influence d'un seul exemple d'entraînement. Un faible gamma signifie que chaque point a une portée large, ce qui peut simplifier la frontière de décision. Un gamma élevé signifie que chaque point a une portée étroite, ce qui peut rendre la frontière de décision plus complexe.

L'hyperplan est défini par la formule 1.9 :

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (1.8)$$

où \mathbf{w} est le vecteur de poids et b est le biais. La marge est maximisée sous la contrainte :

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \forall i \quad (1.9)$$

L'optimisation de l'hyperplan se fait en résolvant le problème suivant :

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (1.10)$$

sous les contraintes :

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \quad (1.11)$$

$$\xi_i \geq 0 \quad (1.12)$$

où ξ_i sont les variables de relaxation pour les marges.

Les noyaux utilisés dans SVM sont définis comme suit :

- **Linear** : $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- **Polynomial** : $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)^d$
- **RBF (Radial Basis Function)** : $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$
- **Sigmoid** : $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)$

5. AdaBoost

Adaboost est une technique de renforcement très populaire qui vise à combiner plusieurs classificateurs faibles pour construire un classificateur fort [14]. En d'autres termes, cela permet de corriger les erreurs itérativement et d'améliorer les performances globales. AdaBoost est particulièrement efficace pour améliorer les performances des modèles simples.

Fonctionnement :

- **Initialisation des poids** : Attribuer des poids égaux à toutes les instances de l'ensemble d'entraînement.
- **Formation des classificateurs faibles** : Former des classificateurs faibles séquentiellement, chaque classificateur se concentrant davantage sur les instances mal classées par les précédents.
- **Mise à jour des poids** : Augmenter les poids des instances mal classées et diminuer ceux des bien classées, en fonction de l'erreur de classification.
- **Combinaison des classificateurs** : Combiner les classificateurs faibles en une seule prédiction finale pondérée.

Paramètres principaux :

- **n_estimators** : Nombre de classificateurs faibles à utiliser.
- **learning_rate** : Taux d'apprentissage qui réduit la contribution de chaque classificateur.

Formule mathématique :

Le classificateur final $H(x)$ est une combinaison pondérée des classificateurs faibles $h_t(x)$:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (1.13)$$

où α_t est le poids attribué au t -ième classificateur faible, déterminé par l'erreur de classification.

6. Gradient Boosting

L'algorithme d'apprentissage ensembliste appelé « Gradient Boosting Classifier » repose sur les arbres de décision. Il a été mis en œuvre pour éviter le surapprentissage [12]. Cette méthode garantit une diminution constante de l'erreur, ce qui améliore la performance globale du modèle.

Fonctionnement :

- **Initialisation** : Commencer avec une prédiction initiale simple, souvent la moyenne des valeurs pour la régression ou la classe majoritaire pour la classification.
- **Construction itérative des arbres** : Ajouter des arbres de décision séquentiellement, chaque nouvel arbre corrigeant les erreurs des arbres précédents en se basant sur le gradient de la fonction de perte.
- **Mise à jour des résidus** : Calculer les erreurs résiduelles après chaque itération et ajuster les modèles suivants pour minimiser ces erreurs.
- **Combinaison des modèles** : Combiner les prédictions de tous les arbres pour former une prédiction finale.

La figure 1.8 illustre un exemple du fonctionnement de Gradient Boosting :

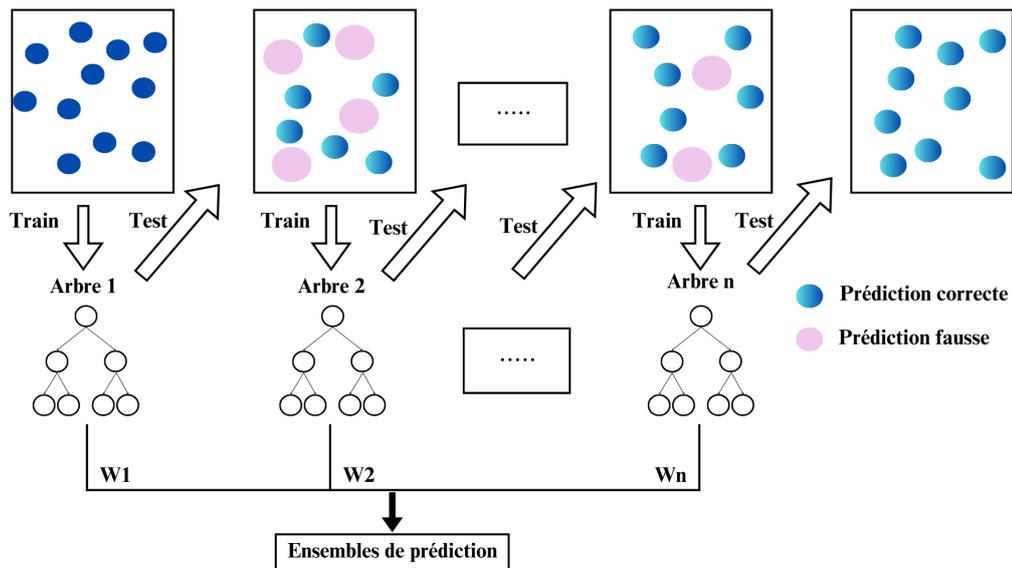


FIGURE 1.8 – Exemple du Gradient Boosting

Paramètres principaux :

- **n_estimators** : Nombre d'arbres dans la séquence d'apprentissage.
- **learning_rate** : Taux d'apprentissage qui réduit la contribution de chaque arbre.
- **max_depth** : Profondeur maximale des arbres. Un plus grand nombre rend le modèle plus complexe.

Formule mathématique :

La prédiction est donnée par :

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \nu f_i(x_i), \quad f_i \in \mathcal{F} \quad (1.14)$$

où ν est le taux d'apprentissage, f_i est le t -ième modèle faible (arbre de décision) et \mathcal{F} est l'espace des modèles faibles.

1.4.6 Deep Learning

Le Deep Learning (DL) est un sous-ensemble spécialisé du Machine Learning qui utilise des réseaux de neurones artificiels profonds pour modéliser des hiérarchies complexes de données. Cette méthode permet aux modèles de s'auto-améliorer et d'apprendre de vastes quantités de données non structurées en extrayant progressivement des caractéristiques de plus en plus significatives à chaque couche du réseau [15]. La figure 1.9 représente un simple schéma illustratif du Deep Learning.

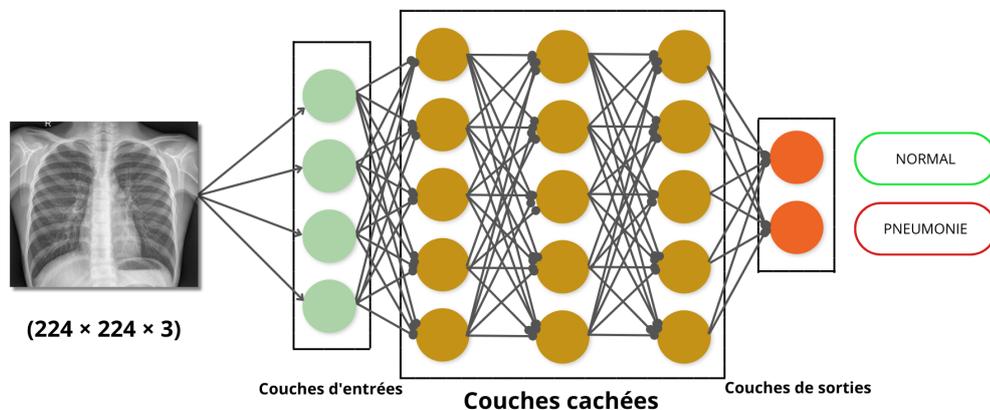


FIGURE 1.9 – Exemple de la structure du Deep Learning

Notre solution se base sur le Réseau neuronal convolutif. Pour cela, la section suivante est dédiée à élaborer cette dernière plus en détail.

1.4.7 Réseau neuronal convolutif

Les réseaux neuronaux convolutifs, en anglais Convolutional Neural Networks (CNNs) sont une classe de modèles de Machine Learning (ML) utilisés principalement pour le traitement des images et la reconnaissance d'objets. Ils contiennent plusieurs couches, y compris des couches convolutives et des couches entièrement connectées [16].

Les images d'entrée sont traitées par les couches convolutionnelles qui les représentent sous la forme de matrices de pixels. Elles présentent une image en gris à deux dimensions. Les couleurs fondamentales (Rouge, Vert, Bleu) sont représentées par une troisième dimension, de profondeur 3.

La partie convolutive d'un CNN opère en tant qu'outil d'extraction des caractéristiques des images. Une série de filtres est utilisée pour générer de nouvelles images, connues sous le nom de cartes de convolutions. Des filtres intermédiaires permettent de diminuer la résolution de l'image en effectuant une opération de maximum local. En fin de compte, les cartes de convolutions sont alignées et combinées en un vecteur de caractéristiques, connu sous le nom de code CNN [16]. La figure 1.10 illustre un exemple de modèle de CNN utilisé pour traiter et classer des images radiographiques, en montrant comment l'image est progressivement transformée à travers plusieurs couches pour aboutir à une prédiction finale.

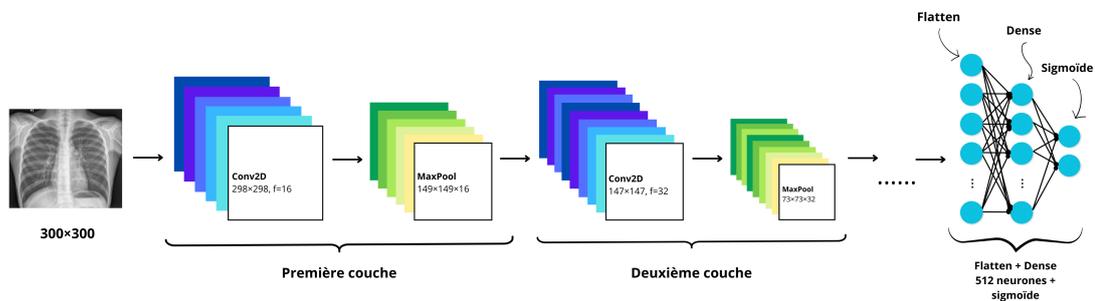


FIGURE 1.10 – Exemple de la structure d'un CNN

Les caractéristiques principales des CNNs incluent :

- **Couches convolutionnelles** : Ces couches utilisent des filtres de convolution pour extraire des caractéristiques locales, comme les contours, les textures et les motifs. Les filtres glissent sur l'image afin de générer des cartes de caractéristiques.
- **ReLU (Rectified Linear Unit)** : Toutes les valeurs négatives reçues en entrées sont remplacées par des zéros dans cette couche. Ces couches d'activation présentent l'avantage de rendre le modèle non linéaire, ce qui facilite la compréhension des relations complexes entre les caractéristiques.
- **Pooling** : La couche de Pooling est une opération appliquée entre deux couches de convolution. Elle réduit la dimensionnalité des cartes de caractéristiques tout en conservant les informations importantes en réduisant le sur-ajustement.
- **Normalisation** : Pour améliorer la généralisation du modèle, on utilise une technique qui normalise les activations des neurones, imitant les neurones biologiques.
- **Couches entièrement connectées (Fully-connected)** : Après les couches convolutionnelles, ces couches sont entièrement connectées à tous les neurones de sorties et jouent le rôle de classificateurs en exploitant les caractéristiques extraites afin de prédire les classes des images.
- **Dropout** : Le dropout permet d'éviter le problème de sur-apprentissage par la désactivation aléatoire d'un pourcentage de neurones qui sont trop liés entre eux pendant l'entraînement. C'est une méthode de régularisation qui permet au modèle de ne pas devenir trop dépendant de certaines caractéristiques et d'améliorer sa capacité de généralisation.

1.5 Optimisation et évaluation des modèles

Pour garantir la meilleure performance des modèles prédictifs, plusieurs techniques d'optimisation et d'évaluation sont mises en œuvre.

1.5.1 GridSearch

GridSearch, ou recherche en grille, est une méthode utilisée en apprentissage automatique pour optimiser les hyperparamètres d'un modèle. Elle consiste à tester systématiquement une grille prédéfinie de valeurs pour chaque hyperparamètre, afin de trouver la combinaison qui maximise les performances du modèle. Chaque combinaison est évaluée en utilisant une technique de validation croisée, ce qui permet de s'assurer que les résultats sont robustes [17].

1.5.2 Validation croisée

La validation croisée est une technique statistique utilisée pour évaluer les performances d'un modèle prédictif en le testant sur plusieurs sous-ensembles de données. Les données sont divisées en plusieurs groupes, ou "**folds**". Pour chaque groupe, le modèle est entraîné sur les autres groupes et testé sur le groupe actuel, permettant ainsi d'évaluer sa capacité à généraliser à de nouvelles données. Cette méthode minimise le risque de surajustement et améliore la fiabilité des prédictions du modèle [18]. La figure 1.11 illustre le processus de validation croisée.

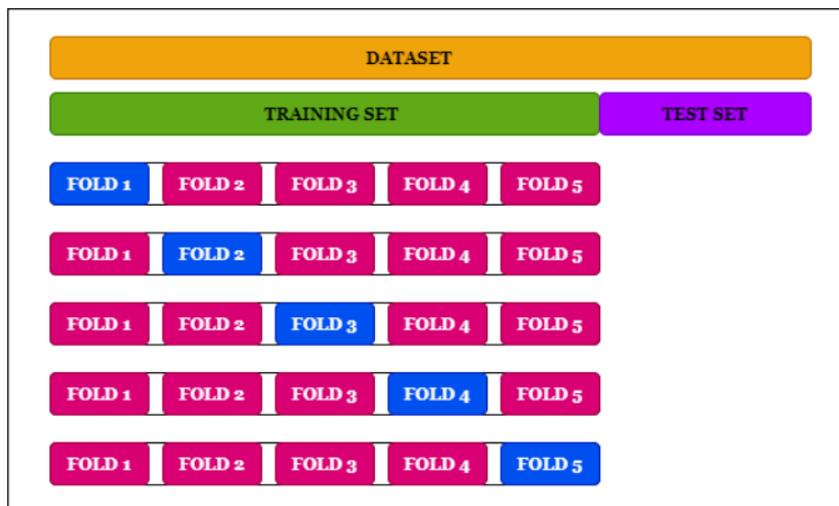


FIGURE 1.11 – Processus de validation croisée [14]

1.6 Métriques d'évaluation

Les métriques d'évaluation en Machine Learning permettent d'évaluer les performances d'un modèle de classification. Pour un problème de classification binaire, le résultat peut être l'une des deux classes possibles : positif ou négatif. Le modèle de classification produit des résultats qui peuvent être évalués en termes de vrais positifs (TP), vrais négatifs (TN), faux positifs (FP) et faux négatifs (FN). Voici comment ces mesures peuvent être définies :

- **Vrais Positifs (TP)** : Le nombre d'instances correctement prédites comme positives par le modèle de Machine Learning.
- **Vrais Négatifs (TN)** : Le nombre d'instances correctement prédites comme négatives par le modèle de Machine Learning.
- **Faux Positifs (FP)** : Le nombre d'instances incorrectement prédites comme positives par le modèle de Machine Learning.
- **Faux Négatifs (FN)** : Le nombre d'instances incorrectement prédites comme négatives par le modèle de Machine Learning.

Les principales métriques d'évaluation utilisées pour analyser les performances des modèles de classification sont les suivantes [19] :

1.6.1 Exactitude (Accuracy)

L'exactitude mesure la proportion de prédictions correctes parmi toutes les prédictions effectuées, tel que indiqué par la formule 1.15 :

$$\text{Exactitude} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1.15)$$

1.6.2 Précision (Precision)

La précision mesure la proportion de vraies prédictions positives parmi toutes les prédictions positives. La formule 1.16 exprime cette métrique :

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (1.16)$$

1.6.3 Rappel (Recall)

Le rappel, ou sensibilité, mesure le taux de vrais positifs identifiés parmi tous les cas positifs réels. La formule 1.17 exprime cette métrique :

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (1.17)$$

1.6.4 F1 Score

Le F1 Score est la moyenne harmonique de la précision et du rappel, offrant une mesure équilibrée lorsque les classes sont déséquilibrées. La formule 1.18 exprime cette métrique :

$$\text{F1} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FN} + \text{FP}} \quad (1.18)$$

1.6.5 Spécificité (Specificity)

La spécificité ou le taux de vrai négatif est une mesure de la capacité d'un modèle à identifier correctement les négatifs, indiquant son efficacité à reconnaître les non-occurrences d'une condition dans des situations où les faux positifs sont particulièrement problématiques. La formule 1.19 exprime cette métrique :

$$\text{Spécificité} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (1.19)$$

1.6.6 Le taux de faux positifs (FPR)

Le taux de faux positifs (FPR : False Positive Rate) mesure la proportion de résultats négatifs réels qui sont incorrectement identifiés comme positifs par un test ou un modèle. La formule 1.20 exprime cette métrique :

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad (1.20)$$

1.7 Matrice de Confusion

La matrice de confusion est un tableau récapitulatif qui montre les performances de l'algorithme de classification en termes de vrais positifs, faux positifs, vrais négatifs et faux négatifs. Permettant ainsi d'évaluer l'efficacité du modèle en identifiant les erreurs commises et en facilitant son optimisation grâce à une visualisation explicite des résultats. Le tableau 1.1 illustre les performances de classification.

		Prédiction	
		Positif	Négatif
Réalité	Positif	VP (Vrais Positifs)	FN (Faux Négatifs)
	Négatif	FP (Faux Positifs)	VN (Vrais Négatifs)

TABLE 1.1 – Un exemple de matrice de confusion

1.8 La courbe de ROC et AUC

Les modèles de classification binaire sont évalués à l'aide de la courbe ROC (Receiver Operating Characteristic), qui représente la relation entre le taux de vrais positifs (Recall) et le taux de faux positifs (1 - spécificité) pour divers seuils de décision.

La mesure de l'aire sous la courbe ROC abrégée en (AUC : Aire under curve) résume cette performance en une seule mesure, allant de 0 à 1, où 1 correspond à une classification parfaite et 0.5 à une efficacité équivalente à un choix aléatoire [20].

1.9 Problématique

Lors de notre stage au (CHU), au sein du service d'imagerie médicale. On a constaté plusieurs défis majeurs auxquels le service est confronté. Tout d'abord, il existe un problème de stockage des images médicales, notamment les radiographies thoraciques, car leur format DICOM¹ est volumineux, cela conduit à une mauvaise gestion des données et à une accessibilité restreinte aux images historiques des patients. Cette situation complique au fil du temps, le diagnostic précis ainsi que l'évaluation de l'évolution des pathologies.

Actuellement, le processus de diagnostic des radiographies thoraciques repose sur l'interprétation manuelle des radiologues. Cela engendre des délais importants dans l'obtention des résultats, surtout lors d'une forte demande et d'un manque de personnel spécialisé. Un diagnostic rapide et précis est essentiel pour la prise en charge efficace des patients, particulièrement ceux atteints de pneumonies.

1.10 Solution proposée

Face à ces enjeux, l'utilisation de techniques avancées des réseaux de neurones, pourrait offrir des perspectives prometteuses pour relever ces défis. Notre projet propose donc d'explorer les différentes techniques d'automatisation pour le processus du diagnostic des radiographies thoraciques, afin d'améliorer sa précision et réduire considérablement les délais d'attente.

En complément, l'adoption d'une solution de stockage dans le Cloud permettra de surmonter les limitations actuelles, en offrant une capacité extensible, facilitant ainsi une meilleure gestion des données. D'autant plus elle permet un accès rapide aux images médicales. Ces améliorations contribueront à une meilleure prise en charge des patients et à une optimisation des ressources humaines et matérielles du service d'imagerie médicale.

Conclusion

Dans ce chapitre initial, nous avons introduit le contexte de notre recherche au CHU de Béjaïa et identifié les principales difficultés dans le diagnostic de la pneumonie par imagerie médicale. Les technologies de l'intelligence artificielle promettent des solutions innovantes, dont l'analyse approfondie se poursuivra dans les chapitres suivants, dans le but d'améliorer la prise en charge des patients.

1. **DICOM** : est la norme internationale pour la transmission, le stockage, l'extraction, l'impression, le traitement et l'affichage d'informations d'imagerie médicale.

2

Travaux connexes

Sommaire

Introduction	22
2.1 Modèles de CNN	23
2.2 Travaux connexes	25
2.3 Analyse comparative	29
Conclusion	30

Introduction

Dans le domaine de la détection et du diagnostic des maladies à partir d'images médicales, les progrès récents de l'intelligence artificielle et de l'apprentissage profond, notamment grâce aux réseaux de neurones convolutifs (CNN), ont permis des avancées significatives. Dans ce chapitre, nous commencerons par expliquer les architectures CNN VGG16 et VGG19. Ensuite, nous nous concentrons sur les travaux de recherche qui utilisent ces modèles pour détecter les maladies pulmonaires et d'autres problèmes à partir de radiographies thoraciques. Nous terminerons par une taxonomie des travaux et une études comparative.

2.1 Modèles de CNN

Le CNN comporte plusieurs modèles pré-entraînés, tels que AlexNet [21], ResNet50 [22], GoogleNet [23], VGG16 et VGG19. Pour notre projet, on s'est focalisé sur VGG16 et VGG19 vu leurs résultats prometteurs.

2.1.1 Modèle VGG16

Le modèle VGG16 est un réseau neuronal convolutionnel pour la reconnaissance d'images à grande échelle. Comparé aux autres modèles, ce dernier a obtenu les meilleures performances pour les tâches de localisation de la compétition annuelle ILSVRC (ImageNet¹ Large-Scale Visual Recognition Challenge) de 2014. Le modèle atteint 92,7% de précision de test ImageNet. Le réseau est profond de 16 couches [24].

2.1.1.1 Architecture du VGG16

Ce modèle prend en entrée des images de tailles (224x224x3). Il est constitué de cinq blocs de couches de convolution accompagnés de couches de MaxPooling. Un filtre de taille 3x3 et un pas (stride) de taille 1 sont utilisés pour chaque couche de convolution. La fonction d'activation ReLU est utilisée pour chaque couche de convolution et la technique du zéro padding est employée de manière à conserver la taille des images après chaque couche de convolution.

Les deux premiers blocs de convolution sont constitués chacun de deux couches de convolution et ont une profondeur de 64 et 128 successivement. Les trois derniers ensembles de convolution comprennent chacun trois couches de convolutions et ont successivement des profondeurs de 256, 512 et 512.

L'opération de Flattening est utilisée à la suite du dernier bloc de couches de convolution. Ensuite, nous avons le classificateur qui comprend 3 couches (deux couches cachées de 4096 neurones chacune et une couche de sortie de 1000 neurones). La dernière couche comporte 1000 neurones, car ce modèle a été conçu pour la classification d'images en 1000 catégories [25]. La figure 2.1 représente l'architecture d'un réseau de neurones convolutionnel VGG16, détaillant les dimensions de chaque couche et les transformations appliquées, de l'entrée à la sortie.

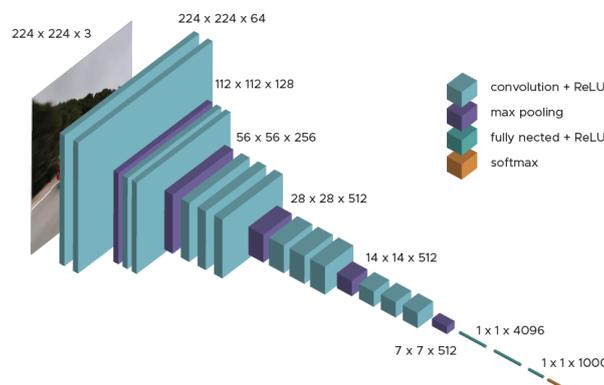


FIGURE 2.1 – Exemple de la structure d'un VGG16 [26]

1. **ImageNet** : est un ensemble de données de plus de 14 millions d'images appartenant à 1000 classes.

2.1.2 Modèle VGG19

Le modèle VGG19 est un réseau neuronal convolucional développé pour l'analyse et la reconnaissance d'images à grande échelle. Il représente une extension du VGG16, incluant trois couches convolutionnelles supplémentaires, ce qui porte le total à 19 couches profondes. Ces ajouts permettent au VGG19 de capturer des caractéristiques plus complexes des images, ce qui s'avère utile pour des tâches de classification plus détaillées. Le modèle a également montré d'excellentes performances sur la base de données ImageNet, avec une précision de classification compétitive, rendant ce modèle particulièrement robuste pour les applications de vision par ordinateur nécessitant une analyse détaillée de l'image.

2.1.2.1 Architecture du VGG19

Le modèle VGG19, prend en charge des images d'entrée de dimensions (224x224x3). Il est constitué de cinq blocs de couches de convolution accompagnés de couches de MaxPooling pour réduire la dimensionnalité. Les filtres de convolution sont uniformément de taille 3x3 avec un pas (stride) de 1, et la fonction d'activation utilisée est ReLU à travers toutes les couches. Il emploie un padding de zéro pour maintenir les dimensions des images après chaque convolution.

Les deux premiers blocs contiennent chacun deux couches de convolution, avec des profondeurs de filtre de 64 et 128, respectivement. Les trois derniers blocs se distinguent par leur complexité accrue, contenant chacun quatre couches de convolution, avec des profondeurs de filtres de 256, 512, et 512, ce qui ajoute trois couches supplémentaires par rapport à des modèles antérieurs moins complexes.

Le Flattening des cartes de caractéristiques s'effectue après le dernier bloc de convolution, menant à un classificateur dense composé de trois couches. Les deux premières sont des couches cachées, chacune ayant 4096 neurones, suivies d'une couche de sortie qui compte 1000 neurones. Cette configuration est destinée à la classification d'images en 1000 catégories différentes [27]. La figure 2.2 représente l'architecture d'un réseau de neurones convolucional VGG19, détaillant les dimensions de chaque couche et les transformations appliquées, de l'entrée à la sortie.

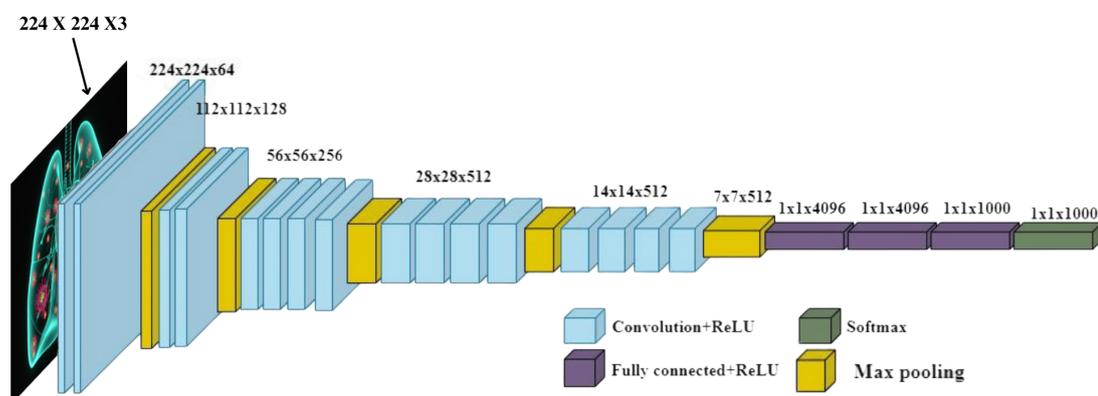


FIGURE 2.2 – Exemple de la structure d'un VGG19

2.2 Travaux connexes

Cette section donne un aperçu des travaux de recherche déjà effectués dans la prédiction des maladies liées au traitement d'images radiographique à l'aide de techniques d'apprentissage automatique et d'apprentissage profond.

2.2.1 Méthodes basées sur VGG

Nilanjan Dey et al. [28] ont proposé une architecture personnalisée de VGG19 pour le diagnostic de la pneumonie à partir de radiographies thoraciques. Leur démarche s'est déroulée en quatre phases principales. Tout d'abord, les chercheurs ont collecté et traité un ensemble de données d'images de radiographie pulmonaire, comprenant à la fois des images normales et des images de pneumonie. Ces images ont été prétraitées et redimensionnées pour être utilisées dans l'étude. Ensuite, pour l'architecture de l'apprentissage en profondeur, les chercheurs ont utilisé des réseaux de neurones profonds pré-entraînés tels que VGG16, VGG19 et ResNet50. Après avoir évalué les performances de ces architectures, ils ont constaté que le modèle VGG19 a les meilleures performances. Pour le personnaliser, les chercheurs ont intégré un schéma d'ensemble de caractéristiques (EFS : Ensemble of Features Scheme). Cette approche innovante combine des caractéristiques extraites manuellement avec des caractéristiques profondes obtenues par apprentissage par transfert (TL : Transfert Learning).

Cette approche visait à améliorer la précision de la classification des images de radiographie pulmonaire. Enfin, la performance du VGG19 personnalisée a été obtenue avec divers classificateurs, notamment SVM, KNN, Random Forest (RF) et Decision Tree (DT).

Les résultats ont montré que le VGG19 personnalisé, lorsqu'il est utilisé avec le classificateur Random Forest, atteignait une précision de classification de 97,94 % sur les radiographies thoraciques.

Zhi-Peng Jiang et al. [29] ont présenté le modèle IVGG13, spécialement développé pour classifier les images radiographiques thoraciques afin d'identifier la pneumonie, utilisant des données provenant de la plateforme Kaggle². Le modèle est une amélioration du VGG16, il réduit la profondeur du réseau par rapport au dernier tout en conservant la capacité de percevoir les caractéristiques importantes des images, et cela en utilisant de petits noyaux de convolution (3x3) et en divisant les couches de convolution et pooling en blocs cinq modulaires, chaque bloc contenant deux couches convolutives et une couche de pooling. Cette approche permet une extraction plus granulaire et efficace des caractéristiques. L'étude met en évidence l'importance cruciale de l'augmentation des données, utilisant des techniques comme le flip horizontal, la rotation et l'ajustement de la luminosité pour compenser les déséquilibres et augmenter le volume de données, améliorant ainsi la robustesse et la précision du modèle. Les performances du modèle IVGG13 ont été clairement supérieures à celles des modèles de CNN bien connus tels que LeNet, AlexNet et GoogLeNet, en se basant sur des critères tels que la précision, le rappel et la mesure F1. Le modèle a atteint une précision de 93%, un rappel de 95% et une mesure F1 de 94%, se comparant favorablement à LeNet (85% F1), AlexNet (88% F1), et GoogLeNet (90% F1). En outre, Zhi-Peng Jiang et al. [29] détaillent les architectures des réseaux AlexNet, VGGNet, et GoogLeNet. Ils expliquent la méthode de préparation et d'augmentation des don-

2. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

nées, et offre une analyse approfondie des résultats expérimentaux qui prouvent l'efficacité de IVGG13 dans l'amélioration de la reconnaissance d'images médicales.

Sana Sahar Guia et al. [30] ont exposé, lors d'une conférence, une nouvelle approche pour la détection de la tuberculose en utilisant la classification d'images de radiographies thoraciques par des méthodes d'apprentissage profond. Ils ont mis en œuvre une combinaison des modèles pré-entraînés VGG16 et VGG19, intégrant des modules d'attention [31] pour capturer efficacement les données spatiales des images. Cette approche tire parti de la capacité des modèles VGG à extraire des caractéristiques visuelles détaillées, tout en utilisant les modules d'attention pour se concentrer sur les régions les plus pertinentes de l'image.

Les techniques de transfert d'apprentissage ont été utilisées pour adapter des réseaux neuronaux profonds, préalablement formés sur le vaste ensemble de données ImageNet, à la tâche spécifique de la détection des signes de tuberculose dans les radiographies thoraciques. Le transfert d'apprentissage permet de réutiliser les poids des couches pré-entraînés sur des images générales, ce qui accélère l'entraînement sur les données spécifiques de tuberculose et améliore les performances du modèle. En incluant des optimiseurs comme Adam [32], qui ajuste le taux d'apprentissage durant l'entraînement, cela améliore la convergence du modèle. L'approche proposée a permis d'atteindre une précision remarquable de 0.9992.

2.2.2 Méthodes basées sur des modèles ensemblistes ou d'autres architectures CNN

Alhassan Mabrouk et al. [33] ont proposé une architecture basée sur des modèles de réseaux de neurones convolutifs (CNN) pré-entraînés, récemment utilisés pour améliorer la performance de nombreuses tâches médicales, plutôt que d'entraîner les modèles CNN à partir de zéro. L'objectif de la méthode proposée est de mettre en lumière la puissance de combiner trois modèles d'apprentissage profond (DL) bien connus - MobileNetV2 [34], DenseNet169 [35] et Vision Transformer (ViT) [36] - pour améliorer la précision et l'efficacité dans la détection et la classification de la pneumonie à partir d'images radiographiques du thorax. Ces modèles, pré-entraînés sur la base de données ImageNet, offrent un point de départ solide grâce à leur capacité déjà développée à détecter et classer des caractéristiques complexes dans les images. Cette approche, baptisée méthode EL (Ensemble Learning) a été affinée pour des ensembles de données radiographiques spécifiques à la pneumonie (de taille 224×224), afin de transformer les images en vecteurs de caractéristiques structurés. Ces vecteurs permettent ainsi de représenter de manière numérique les informations clés des images, facilitant ainsi le processus de classification automatisée pour détecter la présence de la pneumonie. La méthode proposée a atteint des performances de classification satisfaisantes, en surpassant les autres méthodes existantes et obtient une précision de 93,91 % et un F1-score de 93,88 % sur la phase de test.

Rohit Kundu et al. [37] ont appliqué l'apprentissage par transfert profond³ pour gérer la rareté des données disponibles. Ils ont conçu un ensemble de trois modèles de réseaux de neurones convolutifs, à savoir GoogLeNet, ResNet-18 et DenseNet-121. Ces modèles ont été utilisés pour

3. **Apprentissage par transfert profond** : consiste à utiliser des modèles neuronaux pré-entraînés pour améliorer l'apprentissage sur une nouvelle tâche.

la détection de la pneumonie à partir de deux ensembles de données de radiographies thoraciques disponibles publiquement. Le premier, provenant du dataset de Kermany⁴, comprend 5856 radiographies thoraciques issues d'une large population d'adultes et d'enfants, répartis de manière non équilibrée entre les classes "Pneumonie" et "Normal". Le second, a été fourni par la RSNA et a été présenté comme un défi sur Kaggle⁵ pour la détection de la pneumonie. Le cadre d'ensemble adopté par les auteurs intègre de manière efficace les informations discriminantes des différents modèles, ce qui améliore significativement les prédictions comparées à celles obtenues par les modèles individuels. Ils ont développé une technique novatrice pour l'attribution des poids dans l'ensemble, fondée sur des critères d'évaluation tels que la précision, le rappel, le score F1 et l'aire sous la courbe ROC (AUC), plutôt que de se baser uniquement sur l'exactitude des classifieurs. Cette méthode a démontré une grande efficacité, atteignant des taux de précision de 98,81% et 86,85%, ainsi que des sensibilités de 98,80% et 87,02% sur les datasets de Kermany et de RSNA, respectivement.

2.2.3 Méthodes basées sur des approches spécifiques ou non traditionnelles

Xiaosong Wang et al. [38] construisent le dataset "ChestX-ray8"⁶, qui comprend 108 948 images de radios thoracique issues de 32 717 patients, étiquetées avec un ou plusieurs mots-clés de pathologie thoraciques, l'Atélectasie, la Cardiopathie, l'Effusion, l'Infiltration, la Masse, le Nodule, le Pneumothorax et la Pneumonie, ou Normal autrement, dérivées des rapports radiologiques (ainsi que les images). Divers techniques de Deep Learning combinées à des techniques de traitement du langage naturel (NLP) sont utilisées de manière spécifique pour transformer les informations textuelles complexes des rapports radiologiques en données structurées exploitables. Les rapports radiologiques accompagnant les images incluent souvent des termes médicaux qui indiquent la présence ou l'absence de certaines pathologies, ainsi que des expressions de négation ou d'incertitude, comme "aucun signe de pneumonie" ou "possible présence de masse".

Ils ont présenté un cadre de classification d'images multi-étiquettes et qui vise à détecter et à localiser où exactement se trouve la maladie dans l'image radiographique. Malgré le fait que l'ensemble de données soit faiblement étiqueté, où chaque image peut avoir plusieurs étiquettes associées, le système proposé montre des résultats initiaux prometteurs à la fois dans la classification et la localisation des maladies thoraciques courantes.

Pour valider leur méthode, ils ont comparé leur méthode OpenI avec une méthode existante qui est MetaMap [39]. Leur méthode obtient une précision plus élevée de 0,90, un rappel plus élevé de 0,91, et un score F1 plus élevé de 0,90. Ainsi que pour toutes les maladies, des précisions plus élevées, notamment pour "pneumothorax" (0,90 contre 0,32), "infiltration" (0,74 contre 0,25), "Effusion" (0,93 contre 0,74) et "Cardiomegaly" (1 contre 0,99). Cela indique que l'utilisation de la détection de négation et d'incertitude [40] au niveau syntaxique élimine avec succès les cas de faux positifs.

Malgré ces avancées, ils reconnaissent que pour une automatisation complète et précise, ça nécessite des améliorations continues dans les techniques de Deep Learning et une construction

4. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

5. <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge>

6. <https://nihcc.app.box.com/v/ChestXray-NIHCC>

des données médicales plus robustes.

Eduardo Pérez et Sebastian Ventura [41] ont proposé une méthode innovante pour diagnostiquer le mélanome à l'aide d'images. Elle est inspirée de l'apprentissage ensembliste basé sur les modèles de CNN - InceptionV3, DenseNet, VGG16, MobileNet, Xception, et NASNetMobile - et est construite à l'aide d'un algorithme génétique. Cette dernière est conçu pour trouver les membres optimaux de ces ensembles tout en contrastant l'ensemble de la phase d'entraînement des membres potentiels. De plus, les caractéristiques abstraites de tous les modèles de l'ensemble sont fusionnées, ce qui donne une prédiction supplémentaire. Ces prédictions individuelles sont ensuite combinées pour établir un diagnostic final. L'objectif est de trouver les modèles qui contribuent le mieux à l'ensemble, plutôt qu'au niveau individuel. En outre, un nouvel algorithme de segmentation des lésions cutanées, capable d'obtenir efficacement des masques de segmentation fiables sans information préalable, est appliqué. Les expériences ont montré que cette approche a obtenu des performances prometteuses. Les résultats des métriques de performance ont révélé que le modèle de segmentation proposé a obtenu les meilleures performances moyennes avec 81% et 80% de U-Net score (UNS) dans les ensembles de données PH2⁷ et DERM-LIB⁸, respectivement.

7. <https://www.dropbox.com/scl/fi/epzcoqeyr1v9qlv/PH2Dataset.rar?rlkey=6mt2jlvwfkditkyg12xdei6ux&e=1>

8. <https://bit.ly/3gi6rKR>

2.3 Analyse comparative

La tableau 2.1 présente un récapitulatif des articles étudiés.

Article	Modèle utilisé	Dataset	Méthodologie spécifique	Métriques d'évaluation
N. Dey [28]	VGG19 personnalisé	Radiographies thoraciques (normales et pneumonie)	Apprentissage par transfert, Combinaison de caractéristiques manuelles et profondes	Précision : 97,94% (VGG19 avec Random Forest)
Zhi-Ping et al. [29]	IVGG13 (amélioration de VGG16)	Radiographies thoraciques (Kaggle)	Amélioration des architectures existantes, Augmentation des données	Précision : 93% Rappel : 95% F1-score : 94%
S. Guia et al. [30]	VGG16, VGG19 avec modules d'attention	Radiographies thoraciques (Tuberculos)	Combinaison de VGG16 et VGG19, intégration de modules d'attention	Précision : 99%
A. Mabrouk et al. [33]	MobileNetV2, DenseNet169, Vision Transformer (VIT)	Radiographies thoraciques	Apprentissage par transfert, méthode EL (Ensemble Learning)	Précision : 93,91% F1-score : 93,88%
R. Kundu et al. [37]	GoogLeNet, ResNet-18, DenseNet-121	Radiographies thoraciques (Kermany, RSNA)	Apprentissage par transfert, Fusion de classifieurs par moyenne pondérée	Précision : Kermany : 98,81% , RSNA : 86,85%
Xiaosong Wang et al. [38]	Techniques de Deep Learning	ChestX-ray8	Deep Learning avec traitement du langage naturel pour la classification multi-étiquette	Précision : infiltration : 93% Pneumothorax : 90% Effusion : 74%
E. Pérez and S. Ventura [41]	Ensembles de CNNs avec algorithmes génétiques	Images de mélanome (PH2, DERM-LIB)	Algorithmes génétiques, Segmentation des lésions cutanées	U-Net score (UNS) : 81% (PH2), 80% (DERM-LIB)

TABLE 2.1 – Tableau récapitulatif des travaux

D'après les travaux étudiés, les techniques de prédiction des maladies à l'aide de l'apprentissage automatique et de l'apprentissage profond se basent sur trois catégories : les méthodes basées sur les architectures VGG, les approches utilisant des modèles ensemblistes ou d'autres architectures CNN, et les méthodes spécifiques ou non traditionnelles. Les méthodes basées sur VGG, telles que celles proposées par Nilanjan Dey et al [28], Zhi-Peng Jiang et al [29], et Sana Sahar Guia et al. [30], démontrent une forte capacité à améliorer la précision de classification grâce à des techniques avancées d'extraction de caractéristiques et de personnalisation des architectures VGG, atteignant des précisions respectives de 97,94%, 93%, et 99,92%. En parallèle, les méthodes ensemblistes, comme celles de Alhassan Mabrouk et al. [33] et Rohit Kundu et al. [37], combinent plusieurs modèles CNN pour obtenir des performances supérieures en termes de précision et de robustesse, avec des précisions de validation parfaites et des taux de précision atteignant jusqu'à 98,81%. Enfin, les approches spécifiques ou non traditionnelles, illustrées par les travaux de Xiaosong Wang et al. [38], offrent une grande flexibilité et efficacité dans la détection et la classification des maladies grâce à des techniques innovantes comme la classification multi-étiquettes et le traitement du langage naturel, obtenant une précision allant jusqu'à 100% pour certaines maladies. Ainsi Pérez et Ventura [41] utilisent un algorithme génétique pour sélectionner les meilleurs membres de leur ensemble de modèles CNN, fusionnant les caractéristiques abstraites pour établir un diagnostic final, atteignant des scores UNS de 81% et 80% sur les ensembles de données PH2 et DERM-LIB.

Chaque catégorie présente des avantages uniques, permettant ainsi de choisir et de combiner les méthodes en fonction des besoins spécifiques et des contraintes de données pour optimiser les résultats de prédiction des maladies à partir d'images médicales.

Conclusion

Dans ce chapitre, nous avons examiné plusieurs méthodes de prédiction des maladies à partir d'images médicales, à savoir des méthodes basées sur VGG, des modèles ensemblistes, et des techniques spécifiques et non traditionnelles. Par conséquent, nous avons vu comment chacune de ces solutions était plus ou moins précise et applicable. Dans le chapitre suivant, nous proposerons une solution innovante pour la détection de la pneumonie à partir de radiographies thoraciques, en tirant parti des enseignements tirés de ces travaux.

3

Méthodes et résolutions

Sommaire

Introduction	31
3.1 Notre approche	32
3.2 Configuration de l'environnement de travail	33
3.3 Collecte de données	33
3.4 Chargement des données	33
3.5 Prétraitement des données	34
3.6 Visualisation des données	34
3.7 Processus d'extraction des caractéristiques	35
3.8 Processus d'entraînement des modèles de ML	36
Conclusion	42

Introduction

L'utilisation de l'IA dans le domaine du diagnostic médical, ouvre de nouvelles perspectives prometteuses. Dans ce chapitre, nous exposons notre approche pour automatiser ce diagnostic en utilisant **une architecture hybride** qui inclut l'utilisation de modèles de Deep Learning (VGG16 et VGG19) ainsi que différents algorithmes de Machine Learning pour la classification. L'objectif de cette méthode est d'augmenter la précision et la rapidité des diagnostics tout en assurant une gestion sécurisée des données grâce à un **Cloud privé**. Nous décrivons en détail notre étude de cas sur la détection de la pneumonie, qui comprend la collecte et le prétraitement des données, l'extraction des caractéristiques et l'entraînement des modèles.

3.1 Notre approche

Le schéma 3.1 illustre notre proposition d'automatisation du diagnostic des radiographies médicales en intégrant des techniques d'intelligence artificielle (IA) et une solution de stockage cloud privé. Les images médicales DICOM sont d'abord acquises et stockées dans le cloud privé, ce qui permet une gestion optimisée, une sauvegarde sécurisée et un accès rapide aux données historiques des patients du CHU de Béjaïa. Pour la classification de ces images, nous avons proposé une architecture hybride. Cette approche consiste à utiliser des modèles de Deep Learning pour extraire des caractéristiques significatives des images, suivie de l'utilisation d'algorithmes de Machine Learning pour classifier les images. Cette combinaison améliore considérablement la précision et l'efficacité du diagnostic, réduisant les délais et augmentant la fiabilité des résultats. En intégrant cette solution, nous surmontons les défis actuels de stockage et de diagnostic dans le service d'imagerie médicale, tout en optimisant l'utilisation des ressources humaines et technologiques disponibles.

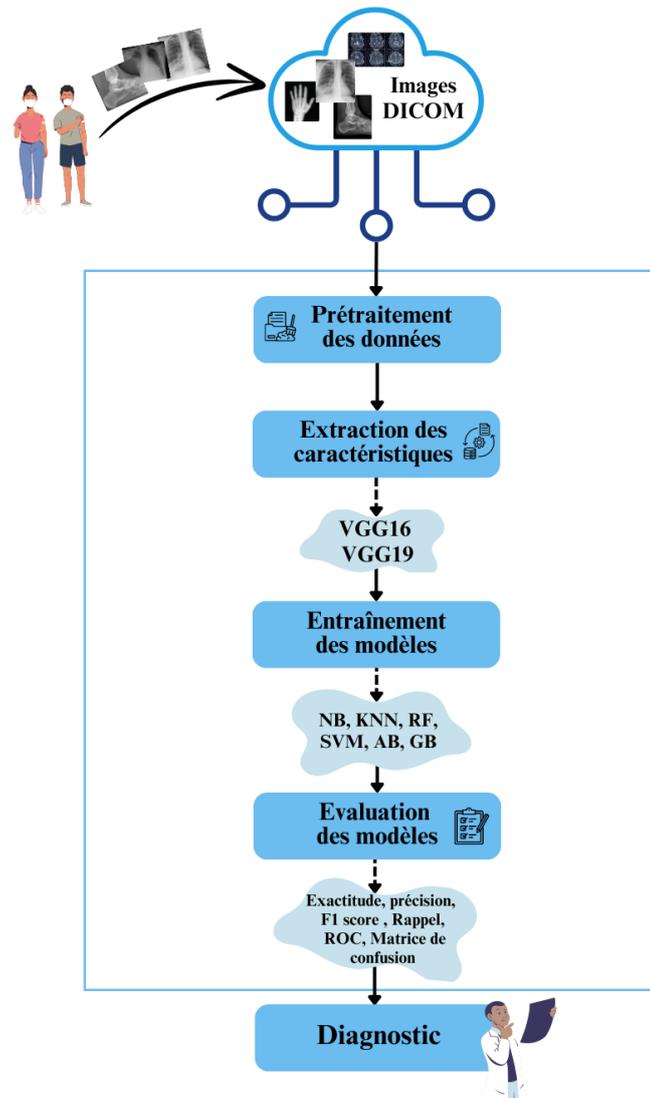


FIGURE 3.1 – Schéma de proposition

Dans ce qui suit, nous détaillons la méthodologie appliquée à notre cas d'étude pour la classification des images médicales entre pneumonie et normal, où nous avons adopté une approche hybride combinant Deep Learning et Machine Learning. Les étapes sont les suivantes :

3.2 Configuration de l'environnement de travail

Cette étape initiale comprend l'installation et le chargement des bibliothèques et des dépendances nécessaires pour notre projet. Il est essentiel de préparer l'environnement avant de commencer toute manipulation de données.

- **Chargement des bibliothèques** : Importer toutes les bibliothèques nécessaires comme `numpy`, `pandas`, `matplotlib`, `seaborn`, `scikit-learn`, `tensorflow`, et autres. Ceux-ci fournissent des fonctionnalités pour le traitement des données, la visualisation, et la construction de modèles de Machine Learning.

3.3 Collecte de données

Pour la réussite de toute étude en *Machine Learning*, une étape cruciale est la collecte des données. Dans notre cas, les données ont été collectées d'abord à partir du service imagerie médicale du CHU de Béjaïa. Vu le nombre d'images réduit, nous avons fait recours aux différentes sources en ligne, principalement sur *Kaggle*. Les jeux de données utilisés sont les suivants :

- **Chest X-Ray Images (Pneumonia)**¹ : Ce jeu de données contient des radiographies thoraciques annotées comme étant des cas de pneumonie ou normales.
- **Autres sources** : Divers autres ensembles de données de radiographies thoraciques, dans différents sites web² ont été explorés et utilisés pour compléter notre jeu de données.

Les images collectées ont été organisées dans un dossier structuré en deux sous-dossiers :

- **PNEUMONIA** : contenant 2866 images de radiographies diagnostiquées comme étant des cas de pneumonie.
- **NORMAL** : contenant 2866 images de radiographies diagnostiquées comme étant des cas normales.

3.4 Chargement des données

Les images de radiographies thoraciques ont un format PNG, elles sont stockées dans deux dossiers distincts sur un serveur local, un pour les images classées comme 'PNEUMONIA' et un autre pour celles classées comme 'NORMAL'. Par la suite, nous importons et vérifions que toutes les images ont été chargées correctement et ont des dimensions uniformes après un éventuel redimensionnement.

1. <https://www.kaggle.com/code/yunannnn/chest-x-ray-images-pneumonia-classification>

2. <https://paperswithcode.com/dataset/chestx-ray8> , <https://data.mendeley.com/datasets/jctsfj2sfn/1>

3.5 Prétraitement des données

Le prétraitement des données comprend plusieurs étapes importantes pour assurer la qualité et l'homogénéité des données avant de les utiliser pour l'entraînement des modèles de Machine Learning. Les étapes de notre prétraitement sont :

3.5.1 Redimensionnement des images

Pour garantir que toutes les images ont la même taille et forme, toutes les images ont été redimensionnées à une taille uniforme de 128x128 pixels. Ce processus a été soigneusement calibré pour s'assurer qu'il n'entraîne pas de perte significative d'informations.

3.5.2 Conversion en niveaux de gris

Les images ont été converties en niveaux de gris pour simplifier le traitement, ce qui est souvent nécessaire pour les tâches de classification d'images médicales.

3.5.3 Normalisation

Les pixels des images sont normalisés pour avoir des valeurs entre 0 et 1. Cette étape aide à améliorer les performances de l'apprentissage en traitant des données à échelle uniforme.

3.5.4 Création des étiquettes

Des étiquettes sont attribuées à chaque image, où '0' pourrait représenter 'NORMAL' et '1' représenter 'PNEUMONIA'. Ces étiquettes sont utilisées pour superviser l'apprentissage.

3.5.5 Exploration des données

L'exploration des données est une étape importante pour comprendre la distribution et les caractéristiques des données. Nous avons vérifié la distribution des étiquettes dans le jeu de données pour nous assurer qu'il y a un équilibre entre les classes 'PNEUMONIA' et 'NORMAL'.

3.6 Visualisation des données

La visualisation des données permet de mieux comprendre la répartition des classes et de vérifier la qualité des images. La figure 3.2 montre un aperçu de la visualisation d'un échantillon de radios thoraciques pour chaque classe ('PNEUMONIA' et 'NORMAL').

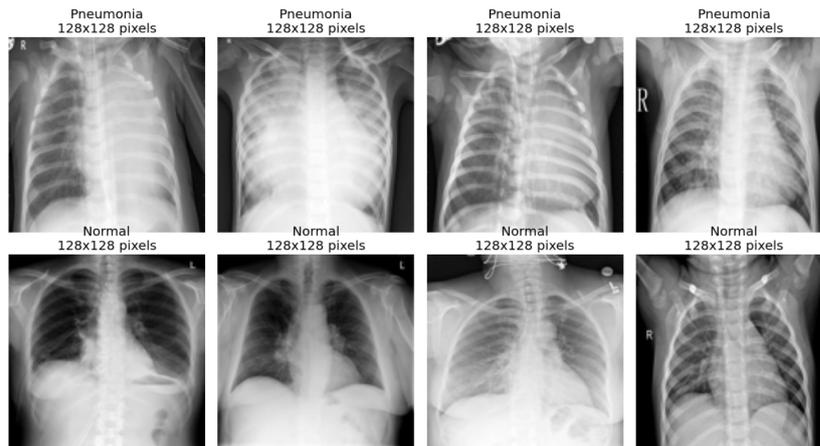


FIGURE 3.2 – Échantillon d'images pour chaque classe ('PNEUMONIA' et 'NORMAL')

Nous avons également utilisé un diagramme circulaire pour visualiser la répartition des étiquettes ('PNEUMONIA' et 'NORMAL'). La figure 3.3 illustre ce diagramme :

Nombre de PNEUMONIA et NORMAL

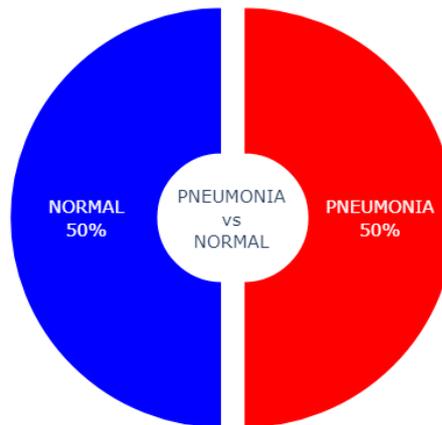


FIGURE 3.3 – Répartition des étiquettes ('PNEUMONIA' et 'NORMAL')

Cette préparation et exploration des données garantissent que les images sont prêtes pour la modélisation, en assurant une compréhension approfondie de leur distribution.

3.7 Processus d'extraction des caractéristiques

Après les visualisations initiales, nous avons procédé à l'extraction des caractéristiques en utilisant des techniques de Deep Learning, notamment les modèles VGG16 et VGG19 pré-entraînés sur ImageNet. Cette étape est cruciale pour transformer les images en ensembles de données structurés que les modèles de Machine Learning peuvent utiliser pour l'entraînement et la prédiction.

3.7.1 Chargement des modèles VGG16 et VGG19

Le modèle VGG16 (resp. VGG19) a été chargé sans les couches de classification finales pour exploiter uniquement ses couches convolutionnelles dans l'extraction des caractéristiques des images. Ces couches finales ne sont pas nécessaires pour cette tâche spécifique, car notre intérêt se porte uniquement sur les caractéristiques des images, et non sur leur classification directe. Cette approche aide également à éviter le surapprentissage, particulièrement important compte tenu de la taille réduite de notre ensemble de données.

3.7.2 Prétraitement des images

Les images ont été prétraitées pour correspondre aux attentes du modèle VGG16 (resp. VGG19). Ce prétraitement inclut le redimensionnement des images à 224x224 pixels et la mise à l'échelle des valeurs de pixels pour correspondre au format attendu par le modèle.

3.7.3 Définition de la fonction d'extraction des caractéristiques

Nous avons défini une classe `ImageFeatureExtractor` pour gérer le processus d'extraction des caractéristiques. Cette classe utilise le modèle VGG16 (resp. VGG19) pour extraire les caractéristiques des images prétraitées.

3.7.4 Chargement et prétraitement des images

Les images de radiographies ont été chargées et prétraitées. Nous avons également assigné des étiquettes aux images : 1 pour les images de pneumonie et 0 pour les images normales.

3.7.5 Combinaison des images et des étiquettes

Les images de pneumonie et normales ont été chargées séparément, puis combinées en un seul ensemble de données.

3.7.6 Extraction des caractéristiques pour toutes les images

Les caractéristiques ont été extraites pour toutes les images en utilisant le modèle VGG16 (resp. VGG19). Ces dernières sont ensuite réorganisées et exportées pour être utilisées dans les modèles de Machine Learning.

3.8 Processus d'entraînement des modèles de ML

Après l'extraction des caractéristiques, nous les avons utilisé pour entraîner plusieurs modèles de Machine Learning. Les étapes suivantes décrivent le processus d'entraînement des modèles, incluant la préparation des données, la sélection des modèles, l'entraînement et l'évaluation des performances.

3.8.1 Préparation des données

Cette étape comprenant le chargement, la normalisation des caractéristiques, et la division des données en ensembles d'entraînement et de test, est standardisée et appliquée uniformément à tous les modèles utilisés dans notre analyse.

- **Chargement des caractéristiques et des étiquettes** : Les caractéristiques extraites et les étiquettes ont été chargées à partir des fichiers sauvegardés.
- **Standardisation des caractéristiques** : La standardisation des données est fréquemment utilisée pour améliorer la stabilité et l'efficacité des algorithmes de Machine Learning. Les caractéristiques extraites, telles que les bordures, les textures et l'intensité des pixels, qui se présentent sous forme d'un ensemble de valeurs numériques, sont standardisées pour garantir une contribution équitable de chaque caractéristique. À cette fin, un scaler standard a été utilisé pour transformer les caractéristiques de sorte qu'elles aient une moyenne de 0 et un écart-type de 1.
- **Division des données** : Les données ont été divisées en ensembles d'entraînement (80%) et de test (20%). Cette approche est utilisée pour vérifier si le modèle est capable de fonctionner correctement avec de nouvelles données, distinctes de celles utilisées lors de son apprentissage.

3.8.2 Sélection des modèles

A. Naïve Bayes

Le modèle Naïve Bayes a été choisi pour sa simplicité et son efficacité dans les problèmes de classification. Il est particulièrement performant lorsqu'il est appliqué à des ensembles de données de grande taille et peut gérer des classes non linéairement séparables grâce à son approche probabiliste.

B. K-Nearest Neighbors

Le modèle KNN a été choisi pour sa simplicité et son efficacité dans les problèmes de classification. Il est particulièrement performant pour les données à distribution inconnue, et il n'est pas nécessaire de faire des hypothèses sur la distribution des données.

C. Random Forest

Le modèle Random Forest a été choisi pour ses excellentes capacités de classification et sa robustesse face aux données bruyantes et aux outliers. Il est particulièrement efficace pour les problèmes de classification grâce à son mécanisme de combinaison de plusieurs arbres de décision pour améliorer la précision et éviter le surapprentissage.

D. Support Vector Machine

Le modèle SVM a été choisi pour sa capacité à gérer efficacement les problèmes de classification binaire et à fonctionner dans des espaces de haute dimension. Il est particulièrement performant pour trouver des frontières de décision optimales entre les classes, même lorsque les données ne sont pas linéairement séparables, grâce à l'utilisation de

différentes fonctions de noyau.

E. AdaBoost

Le modèle AdaBoost a été sélectionné pour sa capacité à améliorer la précision des modèles de base faibles en combinant plusieurs estimations. AdaBoost est particulièrement efficace pour réduire le biais et la variance dans les ensembles de données.

F. Gradient Boosting Classifier

Le modèle Gradient Boosting a été sélectionné pour sa capacité à construire un modèle puissant en combinant plusieurs modèles faibles de manière additive pour corriger les erreurs des modèles précédents. Cette approche est particulièrement efficace pour améliorer la précision et la robustesse des prédictions.

3.8.3 La recherche des hyperparamètres

Dans cette étape, une recherche en grille avec validation croisée (**GridSearchCV**) a été réalisée pour optimiser les performances des différents modèles. Les intervalles des paramètres sélectionnés sont identiques pour les modèles **VGG16** et **VGG19**. Ils ont été choisis sur la base de l'expertise dans le domaine et des résultats préliminaires, et non de manière arbitraire, afin d'assurer une exploration efficace de l'espace des hyperparamètres.

Modèle	Grille de recherche des hyperparamètres
Naïve Bayes	var_smoothing : np.logspace(0, -9, num=10)
KNN	n_neighbors : [5, 7, 9] weights : ['uniform', 'distance'] metric : ['euclidean', 'manhattan']
Random Forest	n_estimators : [50, 100, 200] max_features : ['sqrt', 'log2'] max_depth : [None, 10, 20, 30] criterion : ['gini', 'entropy']
SVM	C : [0.1, 1, 10] Gamma : ['scale', 0.001] Kernel : ['linear', 'rbf', 'poly']
AdaBoost	n_estimators : [50, 100] learning_rate : [0.01, 0.1]
Gradient Boosting	n_estimators : [50, 100] learning_rate : [0.01, 0.1] max_depth : [3, 4] subsample : [1.0]

TABLE 3.1 – Grille de recherche des hyperparamètres pour différents modèles

3.8.4 Résultat de recherche des hyperparamètres

Les paramètres optimaux pour chaque modèle avec les caractéristiques extraites par VGG16 et VGG19 sont présentés dans le tableau 3.2 ci-dessous.

Modèle	VGG16	VGG19
Naïve Bayes	<code>var_smoothing : 0.01</code>	<code>var_smoothing : 1.0</code>
KNN	<code>metric : manhattan</code> <code>n_neighbors : 7</code> <code>weights : uniform</code>	<code>metric : manhattan</code> <code>n_neighbors : 9</code> <code>weights : uniform</code>
Random Forest	<code>criterion : entropy</code> <code>max_depth : 30</code> <code>max_features : sqrt</code> <code>n_estimators : 200</code>	<code>criterion : gini</code> <code>max_depth : None</code> <code>max_features : sqrt</code> <code>n_estimators : 100</code>
SVM	<code>C : 10</code> <code>gamma : scale</code> <code>kernel : rbf</code>	<code>C : 10</code> <code>gamma : scale</code> <code>kernel : rbf</code>
AdaBoost	<code>learning_rate : 0.1</code> <code>n_estimators : 100</code>	<code>learning_rate : 0.1</code> <code>n_estimators : 100</code>
Gradient Boosting	<code>learning_rate : 0.1</code> <code>max_depth : 3</code> <code>n_estimators : 100</code> <code>subsample : 1.0</code>	<code>learning_rate : 0.1</code> <code>max_depth : 4</code> <code>n_estimators : 100</code> <code>subsample : 1.0</code>

TABLE 3.2 – Paramètres optimaux de chaque algorithme ML avec VGG16 et VGG19

3.8.5 Interprétation des paramètres optimaux

Dans cette section, nous interpréterons les paramètres optimaux via **GridsearchCV** et le choix de ces valeurs.

3.8.5.1 Naïve Bayes

VGG16

- **Var Smoothing (0.01)** : Une petite valeur ajoutée à la variance de la distribution lors des calculs de probabilité pour éviter le problème de probabilité nulle, adaptée aux caractéristiques où les extrêmes sont moins fréquents.

VGG19

- **Var Smoothing (1.0)** : Un lissage beaucoup plus élevé, nécessaire pour stabiliser les estimations de probabilité dans un ensemble de données avec une variance potentiellement plus élevée ou plus d'outliers.

3.8.5.2 KNN

VGG16

- **Metric ('manhattan')** : La distance de Manhattan peut être particulièrement efficace pour les données à dimensions élevées comme celles produites par VGG16, où les distances euclidiennes peuvent devenir extrêmes.
- **N Neighbors (7)** : Un nombre modéré de voisins qui équilibre bien entre le bruit et la sensibilité locale des prédictions.
- **Weights ('uniform')** : Chaque voisin influe également sur la prédiction finale, une approche qui suppose que la densité locale est uniforme.

VGG19

- **Metric ('manhattan')** : Utilisée de la même manière pour gérer les données à haute dimension.
- **N Neighbors (9)** : Une légère augmentation du nombre de voisins peut aider à améliorer la stabilité des prédictions dans un ensemble de caractéristiques peut-être plus complexe ou dispersé.
- **Weights ('uniform')** : Comme pour VGG16, l'approche uniforme est maintenue.

3.8.5.3 Random Forest

VGG16

- **Critère ('entropy')** : Favorise des divisions qui maximisent le gain d'information, adapté pour des caractéristiques complexes où chaque division peut apporter une grande amélioration de pureté.
- **Max Depth (30)** : Permet des arbres relativement profonds, ce qui peut capturer des interactions complexes entre les caractéristiques, mais avec un risque contrôlé de surapprentissage en ne descendant pas indéfiniment.
- **Max Features ('sqrt')** : Utilise la racine carrée du nombre total de caractéristiques pour la sélection à chaque division, aidant à réduire la corrélation entre les arbres et à augmenter la diversité des modèles.
- **N Estimators (200)** : Un nombre plus élevé d'arbres pour une meilleure généralisation et robustesse du modèle.

VGG19

- **Critère ('gini')** : Vise à minimiser l'impureté de Gini, une méthode généralement plus rapide que l'entropie et suffisante pour des caractéristiques peut-être légèrement moins complexes de VGG19.
- **Max Depth (None)** : Permet aux arbres de croître autant que nécessaire, ce qui peut capter des nuances fines mais augmente le risque de surapprentissage.
- **Max Features ('sqrt')** : Identique à VGG16, favorise la diversité des arbres.
- **N Estimators (100)** : Moins d'arbres peuvent indiquer soit une efficacité accrue des arbres individuels, soit un compromis pour réduire le coût computationnel.

3.8.5.4 SVM

Pour les architectures **VGG16** et **VGG19**, les paramètres restent constants, indiquant une stabilité dans la configuration du modèle SVM face aux caractéristiques extraites par les deux architectures VGG.

- **C (10)** : Un paramètre de régularisation élevé qui minimise l'erreur de classification, suggérant que chaque erreur de classification est coûteuse.
- **Gamma ('scale')** : Adapte la portée de l'influence d'un seul point de données, ce qui est efficace pour gérer les variations dans les caractéristiques de haute dimension.
- **Kernel ('rbf')** : Le choix d'un noyau radial permet de traiter efficacement les non-linéarités dans l'espace des caractéristiques.

3.8.5.5 AdaBoost et Gradient Boosting

Pour AdaBoost et Gradient Boosting, les paramètres optimisés sont restés constants entre les caractéristiques extraites par **VGG16** et **VGG19**, ce qui montre une adaptation robuste de ces algorithmes face à différentes qualités de données.

- **Learning Rate (0.1)** : Permet un ajustement modéré de l'erreur des prédicteurs précédents, aidant à construire un modèle robuste progressivement.
- **N Estimators (100)** : Ce nombre d'estimateurs fournit un bon équilibre entre la performance et le temps de calcul, efficace pour les caractéristiques des deux architectures.

Et pour les paramètres spécifiques de **Gradient Boosting** on a :

VGG16

- **Max Depth (3)** : Configuration conçue pour un apprentissage progressif sans suradapter, avec des arbres relativement peu profonds pour une généralisation adéquate.

VGG19

- **Max Depth (4)** : Une profondeur légèrement plus élevée pour mieux exploiter les informations peut-être plus détaillées des caractéristiques de VGG19.

3.8.6 Création et entraînement des modèles

Dans cette étape, la procédure de création et d'entraînement des modèles est la même pour VGG16 et VGG19. Nous allons décrire les étapes cruciales pour chaque algorithme.

- **Naïve Bayes** : Un modèle Naïve Bayes est créé avec un paramètre de lissage pour améliorer sa performance. L'entraînement est réalisé sur l'ensemble d'entraînement, permettant au modèle d'apprendre la distribution des données.
- **K-Nearest Neighbors (KNN)** : Le modèle KNN est configuré avec des hyperparamètres qui définissent comment les voisins sont calculés et utilisés dans la prédiction. Le modèle est ensuite entraîné pour identifier les patterns ou les proximités dans l'ensemble d'entraînement.

- **Random Forest** : Un modèle de forêt aléatoire est créé et configuré avec des hyperparamètres appropriés. Ce modèle est ensuite entraîné sur l'ensemble d'entraînement en utilisant la méthode `fit`, adaptant ainsi le modèle aux données pour une meilleure prédiction.
- **Support Vector Machine (SVM)** : Le modèle SVM est configuré avec les hyperparamètres optimisés. Ensuite, le modèle est entraîné sur l'ensemble de données d'entraînement pour apprendre les frontières de décision optimales entre les classes.
- **AdaBoost** : Le modèle AdaBoost est initialisé avec des paramètres qui contrôlent la dynamique de renforcement adaptatif. L'entraînement de ce modèle vise à améliorer progressivement sa capacité à prédire correctement en ajustant les poids des classificateurs faibles.
- **Gradient Boosting** : Un modèle de boosting par gradient est créé pour combiner de manière optimale plusieurs modèles faibles en un modèle robuste. Ce modèle est soigneusement entraîné sur l'ensemble d'entraînement, ajustant les paramètres pour minimiser l'erreur de prédiction.

3.8.7 Évaluation et validation des modèles

Après l'entraînement, les modèles sont évalués et validés sur l'ensemble de test. Les métriques de performance telles que l'exactitude (accuracy), la précision, le rappel, et le score F1 sont calculées pour évaluer chaque modèle. Les matrices de confusion sont souvent utilisées pour visualiser les performances de classification de chaque modèle, ainsi leur courbe ROC est tracée, et l'aire sous la courbe (AUC) est calculée pour évaluer et comparer la performance des différents modèles de manière visuelle et quantitative (Voir les résultats dans **Chapitre 4**).

3.8.8 Sauvegarde des modèles

Les modèles entraînés sont sauvegardés pour une utilisation future, permettant ainsi de charger le modèle et de faire des prédictions sans avoir à réentraîner.

3.8.9 Prédiction sur de nouvelles données

Des fonctions sont définies pour prétraiter de nouvelles images, extraire leurs caractéristiques avec VGG16 et VGG19, et utiliser les modèles entraînés pour prédire si les images montrent des signes de pneumonie ou pas.

Conclusion

Dans ce chapitre, nous avons proposé une approche approfondie pour automatiser le diagnostic des radiographies thoraciques en utilisant une méthode hybride. On a utilisé les modèles VGG16 et VGG19 pour extraire les caractéristiques, puis des différents algorithmes de Machine Learning pour classifier. L'efficacité de cette méthode a été démontrée par une augmentation de la précision et de la rapidité des diagnostics de pneumonie.

4

Résultats et évaluation

Sommaire

Introduction	43
4.1 Environnement de développement	44
4.2 Évaluation des modèles de ML avec VGG16	45
4.3 Évaluation des modèles de ML avec VGG19	50
4.4 Comparaison entre les deux architecture VGG16 et VGG19	53
4.5 Synthèse	54
Conclusion	55

Introduction

Dans ce chapitre, nous décrivons le développement réalisé en Python, en utilisant l'environnement Jupyter Notebook et les bibliothèques clés, contribuant à la réalisation de notre évaluation. Nous nous concentrons sur les résultats et les performances des modèles de Machine Learning appliqués à la classification de radiographies thoraciques, en utilisant des caractéristiques extraites à partir des architectures VGG16 et VGG19. Après avoir décrit en détail notre méthodologie, de la collecte et du prétraitement des données à l'entraînement des modèles, nous analysons les métriques de performance telles que la précision, le rappel, le F1-score, ainsi que l'interprétation des courbes ROC et des matrices de confusion. L'objectif est de comparer les deux architectures VGG16 et VGG19 et de déterminer laquelle offre les meilleures caractéristiques pour différents algorithmes de Machine Learning dans la tâche de classification des radiographies thoraciques.

4.1 Environnement de développement

Dans cette section, nous avons détaillé les outils, le langage de programmation, la plateforme de développement et les bibliothèques que nous avons exploités pour concevoir et mettre en œuvre notre projet.

4.1.1 Langage de Programmation

Pour notre projet, nous avons utilisé un ensemble de technologies et de bibliothèques Python performantes afin de concevoir et mettre en place des modèles d'apprentissage automatique.

Python est un langage interprété, dynamique et polyvalent, qui se démarque par sa clarté et sa facilité d'utilisation. Différents paradigmes de programmation sont compatibles avec Python, tels que la programmation procédurale et orientée objet. Sa gestion automatique de la mémoire et son système de typage dynamique en font un choix parfait pour le prototypage rapide et d'autres applications telles que le développement web, les sciences de données et l'apprentissage automatique. La popularité de Python réside également dans sa large bibliothèque standard qui simplifie l'intégration et le traitement de données, ainsi que dans son écosystème riche en frameworks et bibliothèques de tiers [10].

4.1.2 Plateforme de Développement

Jupyter Notebook [42] a été l'outil privilégié pour l'implémentation de notre projet. Cette application web open source, compatible avec Python, offre un environnement de développement intégré (IDE) puissant permettant de combiner code, visualisations et documentation au sein d'un même espace interactif. Ses capacités intégrées d'exécution de code, de génération de sorties riches en format texte et graphique, ainsi que son support étendu des bibliothèques Python dédiées à la science des données, en font un choix judicieux pour le prototypage et la mise en œuvre efficace de modèles d'apprentissage automatique. L'utilisation de Jupyter Notebook a grandement facilité l'implémentation, le débogage et la visualisation intuitive des résultats de notre modèle.

4.1.3 Bibliothèques utilisées

Les bibliothèques suivantes ont été utilisées dans notre projet :

4.1.3.1 Manipulation de données

- **NumPy (numpy)** : Une bibliothèque Python essentielle pour la manipulation efficace de grands tableaux numériques multidimensionnels, indispensable aux calculs complexes en Deep Learning et Machine Learning [42].
- **Pandas (pandas)** : Offre des structures de données puissantes et flexibles, facilitant la préparation et l'exploration des données avant leur utilisation dans les modèles de Machine Learning [42].

4.1.3.2 Visualisation de données

- **Matplotlib** (`matplotlib.pyplot`) : Permet la visualisation des résultats et performances des modèles à travers une grande variété de graphiques statiques, animés et interactifs [42].
- **Seaborn** (`seaborn`) : Fournit des interfaces pour dessiner des graphiques statistiques attrayants.
- **Plotly** (`plotly`) : Une bibliothèque Python dédiée à la visualisation de données interactives et avancées. Elle permet la création de graphiques dynamiques attrayants, incluant des visualisations 3D, des cartographies et des tableaux de bord interactifs [42].

4.1.3.3 Traitement d'images

- **OpenCV** (Open Source Computer Vision Library) (`cv2`) : est une bibliothèque logicielle libre spécialisée dans le traitement d'images et la vision par ordinateur. Elle offre une vaste gamme de fonctions pour l'analyse d'images, la détection et la reconnaissance d'objets, le traitement vidéo, etc [43].
- **Pillow** (Python Imaging Library) (`PIL.Image`) : Dédiée au traitement d'images. Elle offre des fonctionnalités complètes pour ouvrir, manipuler et enregistrer différents formats d'images, comme le redimensionnement, la rotation et l'application de filtres [42].
- **Scikit-Image** (`skimage`) : Propose des outils pour la segmentation, la transformation et l'analyse d'images.

4.1.3.4 Prétraitement et modélisation

- **Scikit-Learn** (`sklearn`) : Offre des outils puissants pour le prétraitement des données, la sélection de modèles, la validation croisée et les métriques.
- **TensorFlow Keras** (`tensorflow.keras`) : TensorFlow est une plateforme complète conçue par Google pour la création, l'entraînement et la mise en œuvre de modèles de Machine Learning à grande échelle. Keras, une API de premier ordre basée sur TensorFlow, simplifie la conception de réseaux de neurones en rendant le code plus accessible et plus facile à superviser [42].

4.1.3.5 Réseaux de neurones pré-entraînés

- **VGG16 et VGG19** (`tensorflow.keras.applications.vgg16` et `tensorflow.keras.applications.vgg19`) : Modèles de réseaux de neurones profonds pré-entraînés sur ImageNet.

4.2 Évaluation des modèles de ML avec VGG16

Dans cette section, nous discutons des résultats des modèles de classification utilisés pour la pneumonie à partir de radiographies thoraciques. Afin d'évaluer les performances des modèles sur des données de test, nous utilisons la méthode `.predict(X_test)` pour prédire les

étiquettes des données de ce dernier. Les performances sont évaluées en utilisant la matrice de confusion et la courbe ROC, ainsi qu'en analysant l'exactitude, la précision, le rappel, le F1-score et l'AUC de chaque modèle.

4.2.1 Matrices de confusion

Nous débutons par l'utilisation des matrices de confusion, qui nous permettent de visualiser les erreurs de classification et à mesurer la capacité de chaque modèle à différencier correctement les classes "PNEUMONIA" et "NORMAL".

Pour l'interprétation de la matrice, prenons le modèle **Random Forest** comme exemple :

- **Vrais négatifs (VN = 535)** : Cela signifie que le modèle Random Forest a correctement identifié 535 images normales comme étant normales.
- **Faux positifs (FP = 14)** : Le modèle a incorrectement classé 14 images normales comme étant des cas de pneumonie.
- **Faux négatifs (FN = 56)** : Le modèle a incorrectement classé 56 images de pneumonie comme étant normales.
- **Vrais positifs (VP = 542)** : Le modèle a correctement identifié 542 images de pneumonie comme étant des cas de pneumonie.

Le modèle Random Forest a démontré une bonne capacité à identifier les cas de pneumonie et les cas normales, mais les 56 faux négatifs et 14 faux positifs soulignent des lacunes potentiellement critiques dans sa précision diagnostique.



FIGURE 4.1 – Matrices de confusion des différents algorithmes

4.2.1.1 Interprétation globale des matrices de confusion

1. Vrais Positifs (TP)

- **KNN** a le nombre le plus élevé de vrais positifs (555), indiquant une excellente capacité à identifier correctement les cas de PNEUMONIA.
- **SVM** et **GradientBoost** suivent avec des scores de 553 et 551 respectivement, montrant également une forte capacité de détection positive (PNEUMONIA).
- **Random Forest** et **AdaBoost** ont des performances légèrement inférieures mais compétitives avec 542 et 543 vrais positifs.

2. Vrais Négatifs (TN)

- **SVM** montre la meilleure performance en vrais négatifs avec 548, soulignant sa capacité à rejeter efficacement les non-cas (NORMAL).
- Les autres modèles, **Random Forest** et **GradientBoost** affichent également de bons résultats avec respectivement 535 et 537 vrais négatifs.
- Tandis que pour **KNN** a correctement identifié 508 cas comme étant normaux, ce qui montre une bonne performance mais inférieure à celle de SVM.

3. Faux Positifs (FP)

- **SVM** se distingue à nouveau avec seulement **1 faux positif**, indiquant une excellente spécificité.
- **GradientBoost** et **Random Forest** maintiennent des niveaux bas de faux positifs avec respectivement 12 et 14.
- **AdaBoost** présente un nombre plus élevé de 20 faux positifs.
- **KNN** a identifié 41 faux positifs, indiquant que le modèle a une bonne spécificité mais inférieure à celle de SVM encore une fois.

4. Faux Négatifs (FN)

- **Random Forest** et **AdaBoost** représentent respectivement (56) et (55) de faux négatifs, indiquant que les modèles sont généralement bons pour identifier les cas positifs mais peuvent encore manquer certains.
- **SVM** (45) et **Gradient Boost** (47), ce qui montre une haute efficacité des deux modèles dans la détection des cas positifs.
- **Naïve Bayes** représente (65) qui est le plus élevé parmi les modèles examinés, indiquant que ce dernier a tendance à manquer plus de cas positifs que les autres modèles.
- **KNN**, a le nombre le plus bas de faux négatifs (43), suggérant que le modèle est assez compétent pour identifier les vrais cas positifs, bien que son nombre de faux positifs nécessite une attention.

4.2.2 Analyse des courbes ROC

Les courbes ROC sont utilisées pour évaluer la performance des modèles de classification. Une courbe ROC plus proche du coin supérieur gauche indique une meilleure performance.

La figure 4.2 montre les courbes ROC des différents modèles de classification basés sur les caractéristiques extraites avec l'architecture VGG16 :

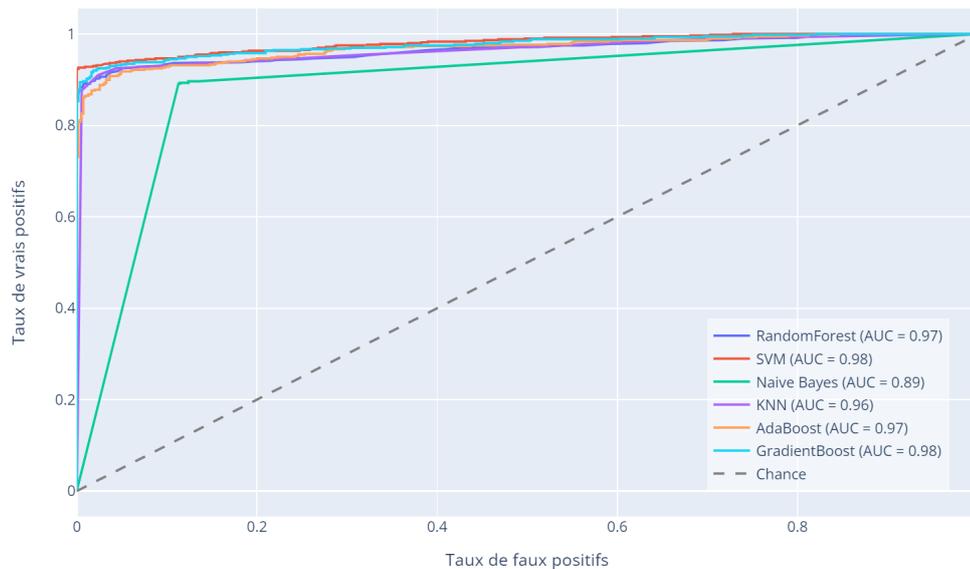


FIGURE 4.2 – Courbes ROC pour différents modèles (VGG16)

En analysant ces courbes, nous constatons :

- **SVM (AUC = 0.98)** : Le modèle **SVM** a la plus grande **AUC** parmi tous les modèles testés, ce qui indique qu'il a la meilleure capacité à distinguer entre les classes **PNEUMONIA** et **NORMAL**. La courbe **ROC** du **SVM** est la plus proche du coin supérieur gauche, ce qui signifie qu'il a le taux de vrais positifs (**sensibilité**) le plus élevé pour un faible taux de faux positifs (**FPR**).
- **Gradient Boost (AUC = 0.98)** : Le modèle **Gradient Boost** montre une performance équivalente à celle du **SVM** avec une **AUC** de **0.98**, démontrant une excellente efficacité dans la classification des classes.
- **Random Forest et AdaBoost (AUC = 0.97)** : Les deux modèles présentent des performances remarquablement similaires avec une **AUC** de **0.97**, ce qui signifie qu'ils ont une très bonne capacité de classification. Leurs courbes **ROC**, très proches du coin supérieur gauche, indiquent une sensibilité élevée et un faible taux de faux positifs, démontrant leur efficacité à discriminer avec précision entre les classes **PNEUMONIA** et **NORMAL**.
- **KNN (AUC = 0.96)** : L'**AUC** du modèle est excellente et indique une forte capacité de distinction entre les classes **PNEUMONIA** et **NORMAL**. Cette performance est très proche de celle des meilleurs modèles, suggérant que **KNN** est presque aussi compétent que les modèles les plus performants dans cette tâche spécifique.
- **Naïve Bayes (AUC = 0.89)** : Le modèle a l'**AUC** la plus basse parmi les modèles analysés, mais avec **0.89**, il maintient une performance acceptable en termes de capacité de classification.

Finalement, le **SVM** et le **Gradient Boost** se distinguent comme les modèles les plus performants selon cette courbe, suivis de près par **AdaBoost** et **Random Forest**. Les autres modèles, bien qu'ayant des **AUC** légèrement inférieures, montrent également des performances respectables.

4.2.3 Analyse des autres métriques de performance

Pour compléter l'évaluation des modèles, nous avons également illustré le graphique 4.3 comparant leurs performances en termes de **précision**, de **rappel** et de **F1-score**.

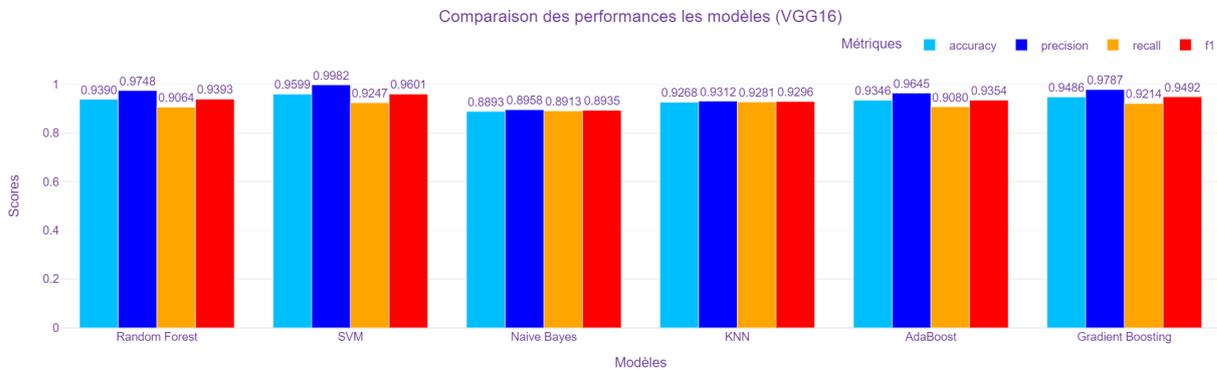


FIGURE 4.3 – Comparaison des performances pour différents modèles (VGG16)

Voici une analyse globale des résultats :

- **Random Forest** : Le modèle montre une haute précision (**0.9748**), un bon rappel (**0.9064**), et un score F1 solide (**0.9393**). Ces résultats indiquent que le modèle est fiable pour identifier correctement les cas de pneumonie avec un faible taux de faux positifs.
- **SVM** : Détient une précision exceptionnelle de **0.9982**, indiquant que presque toutes les prédictions positives du modèle sont correctes. Un rappel de **0.9247**, et un score F1 de **0.9601** illustrent une excellente capacité à classifier précisément les images médicales.
- **Naive Bayes** : Présente des valeurs relativement plus faibles comparées aux autres modèles avec une précision de **0.8958**, un rappel de **0.8913**, et un score F1 de **0.8935**. Malgré cela, il maintient une performance acceptable, n'empêche qu'il pourrait ne pas être le choix optimal dans certains cas.
- **KNN** : Avec une précision de **93.12%**, un rappel de **92.81%**, et un score F1 de **92.96%**, KNN montre des performances équilibrées et robustes, indiquant une bonne capacité à classifier correctement les cas de pneumonie.
- **AdaBoost** : Avec une précision de **0.9645**, un rappel de **0.9080**, et un score F1 de **0.9354**. Ces indicateurs montrent que AdaBoost est performant pour classer les images de manière précise et fiable.
- **Gradient Boosting** : Affiche une précision de **0.9787**, un rappel de **0.9214**, et un score F1 de **0.9492**, reflétant une forte capacité de classer correctement la majorité des instances positives.

Ces résultats indiquent que, bien que tous les modèles performant relativement bien, SVM et Gradient Boosting se démarquent en termes de précision et de score F1, ce qui les rend particulièrement efficaces pour la classification des images de pneumonie.

4.3 Évaluation des modèles de ML avec VGG19

Cette section suit la même démarche que celle adoptée pour l'évaluation des modèles avec VGG16. Nous appliquons les mêmes métriques de mesure des performances pour comparer et analyser les résultats obtenus avec l'architecture VGG19.

4.3.1 Matrices de confusion

Pour l'interprétation globale de la matrice de confusion, (voir la figure 4.4) nous avons :



FIGURE 4.4 – Matrices de confusion des différents algorithmes

1. Vrais Positifs (TP) :

- Le modèle **SVM** montre le nombre le plus élevé de vrais positifs, atteignant **552**, suivi de près par **GradientBoost** avec **549**. Cela suggère que le **SVM** est le plus efficace pour identifier correctement les cas positifs.
- **Random Forest (544)**, **KNN (543)**, **Naïve Bayes (541)** et **AdaBoost (541)** présentent également de bonnes performances, bien qu'un peu inférieures, mais toujours compétitives.

2. Vrais Négatifs (TN) :

- **SVM** est le modèle avec le plus haut nombre de vrais négatifs (**543**), indiquant une très bonne capacité à reconnaître les cas négatifs. **Random Forest** montre également une bonne performance avec (**536**).
- **KNN (537)** et **GradientBoost (535)** montrent également de bonnes performances en termes de vrais négatifs, tandis que **AdaBoost (530)** et **Naïve Bayes (519)** montrent des performances légèrement inférieures.

3. Faux Positifs (FP) :

- **SVM** se démarque avec le plus bas nombre de faux positifs (6), indiquant une grande spécificité. **KNN (12)** et **GradientBoost (14)** ont également de faibles faux positifs.
- **AdaBoost (19)** et **Random Forest (13)** montrent des résultats légèrement supérieurs, tandis que **Naïve Bayes (30)** a le plus grand nombre, ce qui le rend moins efficace pour répondre aux objectifs dans certains contextes.

4. Faux Négatifs (FN) :

- **SVM (46)** présente le nombre le plus bas de faux négatifs, indiquant une bonne sensibilité. **GradientBoost (49)** et **Random Forest (54)** sont également performants dans cette catégorie.
- **Naïve Bayes (57)**, **AdaBoost (57)** et **KNN (55)** ont les valeurs les plus élevées, ce qui peut être un inconvénient dans une situation où il est très important de ne pas manquer de cas positifs.

En résumé, **SVM** se démarque comme étant le modèle le plus équilibré, avec les meilleurs scores en vrais positifs (**TP**) et les plus bas en faux négatifs (**FN**) et faux positifs (**FP**), ce qui est important pour minimiser les erreurs de classification. Bien que **Random Forest** et **GradientBoost** offrent également de bonnes performances globales, ils enregistrent quelques faux négatifs (**FN**) supplémentaires comparé à **SVM**. Les autres modèles, quoique performants dans certains aspects, présentent certaines faiblesses, comme les faux positifs (**FP**) plus élevés pour **Naïve Bayes** et les faux négatifs (**FN**) pour **KNN** et **AdaBoost**, cela les rend moins efficace pour certains contextes spécifiques.

4.3.2 Analyse des courbes ROC

La figure 4.5 montre les courbes ROC des différents modèles de classification basés sur les caractéristiques extraites avec l'architecture VGG19 :

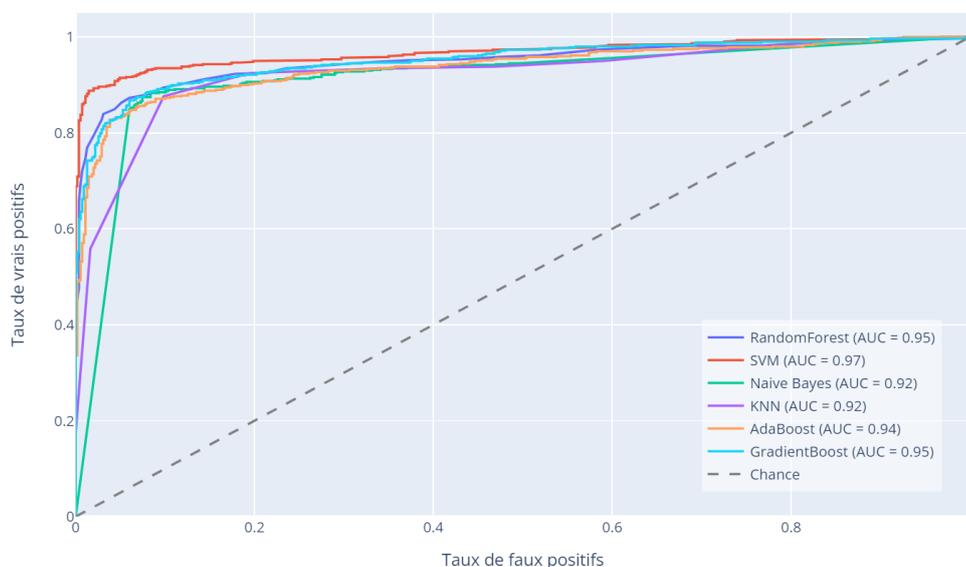


FIGURE 4.5 – Courbes ROC pour différents modèles (VGG19)

En analysant ces courbes, nous pouvons faire les observations suivantes :

- **SVM (AUC = 0.97)** : Le modèle SVM a la plus grande AUC parmi tous les modèles testés, ce qui indique qu'il a la meilleure capacité à distinguer entre les classes PNEUMONIA et NORMAL. La courbe ROC du SVM est la plus proche du coin supérieur gauche, ce qui signifie qu'il a le taux de vrais positifs (**sensibilité**) le plus élevé pour un faible taux de faux positifs (**FPR**).
- **Random Forest et Gradient Boost (AUC = 0.95)** : Les modèles **Random Forest** et **Gradient Boost** ont des performances similaires avec une AUC de **0.95**. Leurs courbes ROC montrent également une bonne performance en termes de distinction entre les deux classes.
- **AdaBoost (AUC = 0.94)** : Le modèle AdaBoost suit de près avec une AUC de **0.94**. Sa courbe ROC montre également une bonne performance en termes de classification.
- **Naïve Bayes et KNN (AUC = 0.92)** : Les modèles **Naïve Bayes** et **KNN** ont des performances similaires avec une AUC de **0.92**. Leurs courbes ROC sont légèrement inférieures à celles du SVM, **Random Forest**, **Gradient Boost**, et **AdaBoost**, mais montrent néanmoins une bonne capacité de classification.

Finalement, Le SVM se distingue comme le modèle le plus performant selon cette courbe, suivi par Random Forest, Gradient Boost, et AdaBoost. Les autres modèles, bien qu'ayant des AUC légèrement inférieures, montrent également des performances respectables.

4.3.3 Analyse des autres métriques de performance

Pour compléter l'évaluation des modèles du VGG19, nous avons également évalué leurs métriques. La figure 4.6 montre cette comparaison.

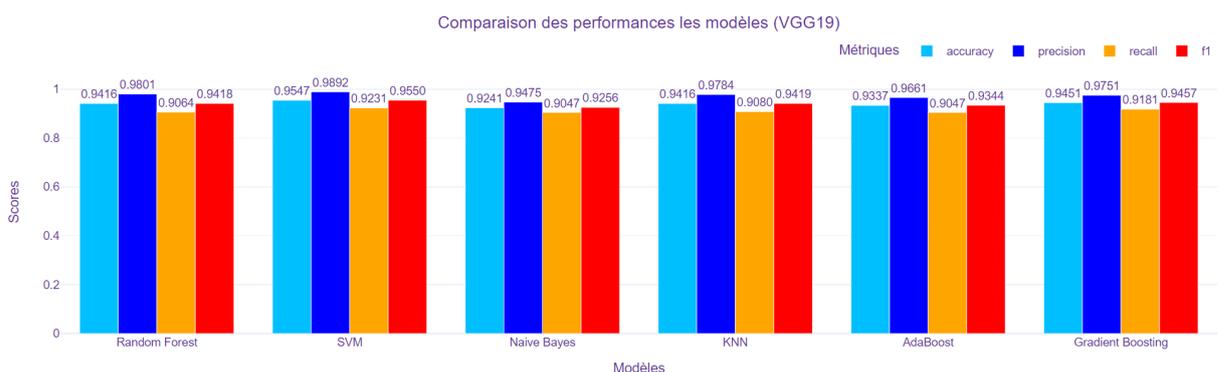


FIGURE 4.6 – Comparaison des performances pour différents modèles (VGG19)

- **SVM** : Le modèle SVM se distingue par la meilleure précision (**0.9892**) et le meilleur F1-score (**0.9550**), ce qui indique une excellente capacité de classification, avec un bon équilibre entre la précision et le rappel.
- **Random Forest** : Le modèle Random Forest affiche également de bonnes performances avec une précision de **0.9801** et un F1-score de **0.9418**, montrant une capacité robuste à identifier correctement les images de pneumonie et les images normales.

- **Gradient Boost** : Ce modèle présente une précision de **0.9751** et un F1-score de **0.9457**, des résultats similaires à ceux de Random Forest, démontrant également une bonne performance de classification.
- **AdaBoost** : Le modèle AdaBoost affiche une précision de **0.9661** et un F1-score de **0.9344**, montrant une capacité de classification solide, bien que légèrement inférieure à celle de SVM, Random Forest et Gradient Boost.
- **KNN** : Le modèle KNN a une précision de **0.9784** et un F1-score de **0.9419**. Bien qu'il soit légèrement en retrait par rapport à SVM, il montre néanmoins une bonne performance globale.
- **Naïve Bayes** : Avec une précision de **0.9475** et un F1-score de **0.9256**, le modèle Naïve Bayes est le moins performant parmi les modèles comparés, mais il reste efficace pour la classification des images de pneumonie.

En résumé, les résultats montrent que les modèles **SVM, Random Forest et Gradient Boost** sont particulièrement efficaces pour la classification des images de pneumonie extraites à partir de VGG19. **AdaBoost et KNN** offrent également des performances solides, tandis que **Naïve Bayes**, bien que performant, est légèrement en retrait par rapport aux autres modèles.

4.4 Comparaison entre les deux architecture VGG16 et VGG19

Dans cette section nous procédons à comparer les deux architectures VGG16 et VGG19 en se basant sur la précision.

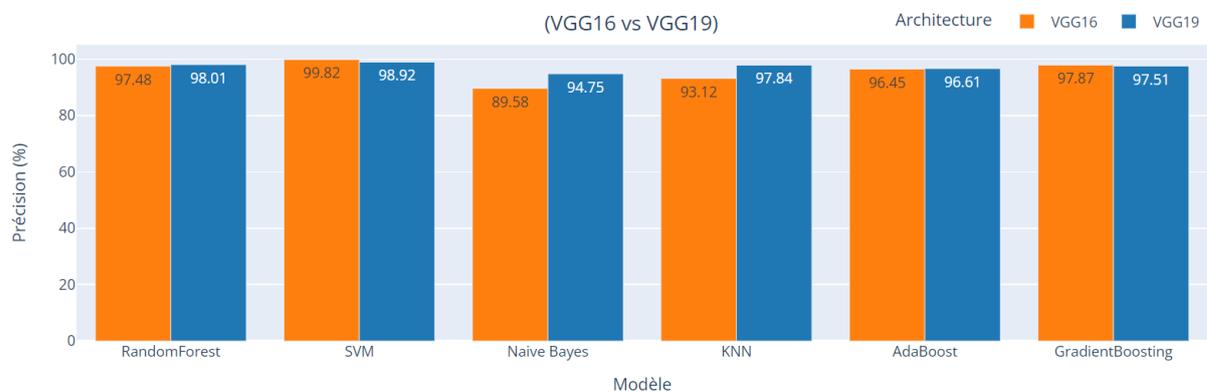


FIGURE 4.7 – Comparaison de la précision des architectures VGG19 et VGG16

La figure 4.7 compare la précision (en %) des modèles de Machine Learning utilisant les caractéristiques extraites par les architectures VGG16 et VGG19. Voici une analyse des résultats :

Pour le modèle Random Forest, la précision est légèrement meilleure avec VGG19 (98.01%) par rapport à VGG16 (97.48%), indiquant une légère supériorité des caractéristiques extraites par VGG19.

Le modèle SVM montre une précision plus élevée avec VGG16 (99.82%) par rapport à VGG19 (98.92%), bien que les deux performances restent très élevées, ce qui indique que les caractéristiques extraites par VGG16 sont légèrement plus efficaces pour ce modèle.

Le modèle Naïve Bayes bénéficie d'une amélioration significative avec VGG19, atteignant une précision de 94.75 % contre 89.58% avec VGG16. Cela suggère que les caractéristiques extraites par VGG19 sont plus discriminantes pour ce modèle.

La précision du modèle KNN avec VGG16 est de 93.12%, illustrant une bonne performance. Toutefois, avec VGG19, la précision augmente significativement pour atteindre 97.84%. Cet accroissement souligne l'avantage des caractéristiques extraites par VGG19, qui semblent mieux convenir à ce modèle pour améliorer sa capacité de classification.

Les performances du modèle AdaBoost sont très similaires pour les deux architectures, avec une précision de 96.61% pour VGG19 et 96.45% pour VGG16, montrant une légère amélioration avec VGG19. Enfin, pour le modèle Gradient Boosting, les performances sont presque identiques entre les deux architectures, avec une précision de 97.87% pour VGG16 et 97.51% pour VGG19.

Voici la figure 4.8 qui illustre les différences entre toutes les métriques des différents modèles des deux architectures :

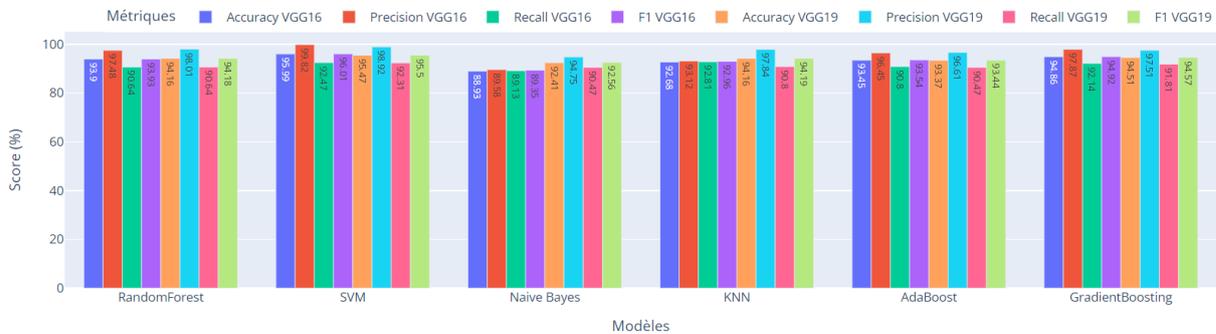


FIGURE 4.8 – Comparaison des métriques des architectures VGG19 et VGG16

4.5 Synthèse

Les résultats de la comparaison des précisions des modèles de Machine Learning utilisant les caractéristiques extraites par les architectures VGG16 et VGG19 montrent que VGG19 offre des avantages notables pour certains modèles. En particulier, VGG19 améliore significativement la précision des modèles Naïve Bayes et KNN, ce qui indique que les caractéristiques extraites par cette architecture sont plus efficaces pour ces algorithmes. Pour d'autres modèles comme RandomForest, SVM, AdaBoost et Gradient Boosting, les différences de précision entre VGG16 et VGG19 sont minimes, suggérant que les deux architectures fournissent des caractéristiques de haute qualité. En conclusion, bien que VGG19 puisse offrir des performances légèrement supérieures pour certains modèles, VGG16 reste une alternative robuste et efficace, et le choix entre les deux architectures peut être guidé par le type de modèle de Machine Learning utilisé.

Conclusion

Ce chapitre a permis d'évaluer et de comparer les performances des modèles de Machine Learning basés sur les caractéristiques extraites par les architectures VGG16 et VGG19 pour la classification des images de pneumonie. Ces dernières fournissent des résultats de haute qualité, permettant une classification efficace. L'analyse des matrices de confusion et des courbes ROC a révélé que les modèles SVM, Random Forest et Gradient Boosting offrent des performances exceptionnelles en termes de précision, rappel et F1-score, avec SVM se distinguant particulièrement par sa capacité à minimiser les faux positifs et les faux négatifs. VGG19 a montré des avantages notables pour certains modèles, notamment Naïve Bayes et KNN, en améliorant leur précision par rapport à VGG16. Pour d'autres modèles comme Random Forest, SVM, AdaBoost et Gradient Boosting, les différences de performances entre VGG16 et VGG19 sont minimales, suggérant que les deux architectures sont tout aussi efficaces. En conclusion, bien que VGG19 puisse offrir des performances légèrement supérieures pour certains modèles, VGG16 reste une alternative robuste et performante, et le choix entre les deux architectures peut être guidé par le type de modèle de Machine Learning utilisé.

Conclusion générale et travaux futures

Ce mémoire a exploré l'application de l'intelligence artificielle, en particulier des réseaux de neurones convolutifs (CNN), VGG16 et VGG19, combinées à divers algorithmes de Machine Learning, pour l'analyse des images médicales, avec un accent particulier sur le diagnostic de la pneumonie à partir des radiographies thoraciques. En suivant une méthodologie comprenant la collecte et le prétraitement des données, le développement et l'entraînement des modèles, ainsi que l'évaluation des performances, nous avons démontré le potentiel des technologies de Machine Learning à améliorer le diagnostic médical.

Les résultats obtenus montrent que les paramètres de chaque classificateur jouent un rôle essentiel dans l'amélioration de la précision des résultats. Il est donc primordial de chercher les paramètres les plus appropriés afin de concevoir un classificateur efficace. Le SVM, parmi les divers algorithmes testés, se révèle être le meilleur modèle lorsqu'il est combiné avec les deux architectures (VGG16 et VGG19) pour classer les données. Par ailleurs, nos modèles ont atteint une précision élevée dans la classification des images radiographiques de la pneumonie, soulignant ainsi l'efficacité et la fiabilité des approches de Deep Learning dans cette application.

L'intégration de modèles de Machine Learning dans les processus de diagnostic peut non seulement améliorer la précision et la rapidité des diagnostics, mais aussi alléger la charge de travail des radiologues. Cela est particulièrement pertinent dans les contextes où les ressources humaines et matérielles sont limitées, comme dans de nombreux établissements de santé.

Malgré les résultats prometteurs, notre étude présente certaines limitations. Le succès des modèles de Machine Learning dépend fortement de la qualité et de la diversité des données utilisées pour l'entraînement. De plus, Les défis associés au prétraitement des images radiographiques comprennent la nécessité de garantir la qualité et la cohérence des données avant leur analyse par les modèles. De plus, l'interprétation des modèles de Deep Learning pose un défi en raison de leur complexité et de la difficulté à expliquer de manière claire et transparente les décisions prises par ces algorithmes. Autrement dit, l'interprétation des résultats par les modèles nécessite une validation clinique rigoureuse pour garantir leur applicabilité dans des situations réelles. La généralisation des modèles à des données variées et nouvelles est également une préoccupation, nécessitant des tests sur des ensembles de données diversifiés et l'utilisation de techniques de régularisation. Enfin, la confidentialité des données médicales posent des défis supplémentaires, nécessitant des protocoles de sécurité rigoureux pour protéger les informations des patients tout en permettant l'entraînement des modèles. En abordant ces limitations et défis, nous pouvons améliorer l'impact et l'applicabilité des technologies de Machine Learning dans le diagnostic médical, contribuant ainsi à une meilleure prise en charge des patients.

Perspectives et travaux futurs

Pour améliorer la précision des modèles de détection de pneumonie à partir d'images radiographiques, il est recommandé de :

- Amélioration du prétraitement des images en optimisant la qualité des images en utilisant des méthodes de renforcement du contraste, et d'augmentation de données (rotation, zoom, décalage) pour accroître la robustesse du modèle face à des variations d'images.
- L'exploration d'architectures plus récentes comme ResNet ou DenseNet pourrait apporter des améliorations supplémentaires en termes de performance et d'efficacité, offrant de nouvelles perspectives pour la détection de la pneumonie.
- L'intégration d'une segmentation préalable des images pulmonaires pourrait significativement réduire le bruit et concentrer l'analyse sur les zones d'intérêt, améliorant ainsi la précision diagnostique.
- Une recherche plus approfondie des hyperparamètres pour chaque combinaison de VGG et d'algorithme de Machine Learning pourrait affiner les résultats et améliorer la robustesse des modèles.

En tant que contribution aux science des données et à l'aide à la décision, cette recherche démontre le potentiel de ces technologies pour transformer la pratique clinique. Les modèles développés pourraient servir d'outils d'aide à la décision pour les radiologues, accélérant le diagnostic et améliorant potentiellement la qualité des soins.

En somme, ce mémoire a démontré le potentiel du Machine Learning pour améliorer le diagnostic de maladies comme la pneumonie. Les améliorations proposées ouvrent la voie à des innovations significatives dans le domaine de la santé, contribuant à une prise en charge plus efficace et plus précise des patients. Les travaux futurs devront se concentrer sur la validation clinique et l'amélioration continue des modèles pour maximiser leur impact et leur adoption dans le milieu médical.

Bibliographie

- [1] Site web du centre hospitalo-universitaire de béjaïa. <https://www.chubejaia.dz/OrganigrammeCHU>, 2024.
- [2] S.M.J. Marouruana, O.P. Ali, T.A. Ida, S. Habibou, K. Bassirou, Nina-Astrid O., B.Y. Abel, and D. Ousseini. Evaluation of medical prescribers' theoretical knowledge on medical imaging in the northern region of burkina faso. *Open Journal of Radiology*, 14 :33–41, 2024.
- [3] Catherine Simand and Hagop Demirdjian. La radiographie ii. qu'est-ce qu'un rayon x? comment en produire? quel mécanisme permet d'obtenir une radiographie? *CultureSciences-Chimie*, 2007.
- [4] X-ray (radiography), chest. <https://www.radiologyinfo.org/en/info/chestrad>. Reviewed Mar-15-2010.
- [5] Saad Khan Afridi. Computed tomography (ct) scanning : Principles and applications. *Department of Radiology, Yale University*, May 2023.
- [6] Caruyer Emmanuel. *Imagerie par résonance magnétique de diffusion. Le corps en images*. HAL Id : inserm-03798395, 2022.
- [7] et al. C. H. Haddar. Strategy using a new antigenic test for rapid diagnosis of streptococcus pneumoniae infection in respiratory samples from children consulting at hospital. *BMC Microbiology*, 2019.
- [8] Roumaïssa Doumani and Radja Kertous. Modèle deep learning pour différencier la pneumonie covid-19 des autres pneumonies. Master's thesis, UAMO de Bouira, Faculté des Sciences et des Sciences Appliquées, Département d'Informatique, October 2021.
- [9] C. Feldman D. G. Wootton. The diagnosis of pneumonia requires a chest radiograph (x-ray)-yes, no or sometimes? *Pneumonia*, 2014.
- [10] Mark Lutz. *Learning Python*. O'Reilly Media, 5th edition, 2013.
- [11] Siafa Aya. Un schéma de pondération et de sélection des termes pertinents basé sur la distribution des fréquences des documents et des termes entre catégories. Master, Université de 8 Mai 1945 – Guelma, 2022.
- [12] H. A. K. I. M. El Massari. *Proposition d'un modèle de prédiction basé sur Machine Learning et le web sémantique*. PhD thesis, UNIVERSITE SULTAN MOULAY SLIMANE, 2023.
- [13] M. K. Moudachirou. *Classification et forêts aléatoires : Application à l'aide à la décision chirurgicale du genou par arthroplastie*. Doctoral dissertation, Télé-université, 2017.

- [14] N. M. Benhamdine M. A. Hamour. Prédiction du churn rate par le machine learning dans le secteur des m&a application au sein de kpmg. Master's thesis, École Nationale Polytechnique, 2020.
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [17] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning : with Applications in R*. Springer, 2013.
- [18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.
- [19] Bradley J. Erickson and Felipe Kitamura. Magician's corner : 9. performance metrics for machine learning models. *Radiology : Artificial Intelligence*, 3(3), 2021.
- [20] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 2006.
- [21] C. J. Lin and Y. C. Li. Lung nodule classification using taguchi-based convolutional neural networks for computer tomography images. *Electronics*, 9(7), 2020.
- [22] Ahmet Çinar and Muhammed Yildirim. Detection of tumors on brain mri images using the hybrid convolutional neural network architecture. *Medical Hypotheses*, 139, 2020.
- [23] S. Chen, J. Zhang, X. Wei, and Q. Zhang. Alzheimer's disease classification using structural mri based on convolutional neural networks. In *Proceedings of the 2020 2nd International Conference on Big-Data Service and Intelligent Computation*, pages 7–13, December 2020.
- [24] Aicha Fakroun and Souhila Maache. Détection du covid-19 à l'aide des images médicales x-ray. Master's thesis, Université Kasdi Merbah Ouargla, Faculté des nouvelles technologies de l'information et de la communication, Département d'informatique et technologie de l'information, 2022. Mémoire de Master en Informatique, Option : Administration et sécurité des réseaux.
- [25] M. A. Nagnan. Développement d'outils web de détection d'annotations manuscrites dans les imprimés anciens. Master's thesis, Université de Paris, April 2021.
- [26] Raphael Kassel. <https://datascientest.com/quest-ce-que-le-modele-vgg>, 2021.
- [27] V. Sudha and T. R. Ganeshbabu. A convolutional neural network classifier vgg-19 architecture for lesion detection and grading in diabetic retinopathy based on deep learning. *Computers, Materials & Continua*, 66(1) :828–842, 2021.
- [28] N. Dey, Y. D. Zhang, V. Rajinikanth, R. Pugalenth, and N. S. M. Raja. Customized vgg19 architecture for pneumonia detection in chest x-rays. *Pattern Recognition Letters*, 143 :67–74, 2021.
- [29] Zhi-Ping Jiang, Ying-Ying Liu, Zhi-En Shao, and Kai-Wei Huang. An improved vgg16 model for pneumonia image classification. *Applied Sciences*, 11 :11185, 2021.
- [30] S. S. Guia, A. Laouid, M. Kara, and M. Hammoudeh. Tuberculosis detection using chest x-ray image classification by deep learning. *Journal Name*, 2023.

- [31] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Convolutional block attention module. *arXiv preprint arXiv :1807.06521*, 2018.
- [32] Ange Tato and Roger Nkambou. Improving adam optimizer. *Workshop track - ICLR 2018*, 2018.
- [33] A. Mabrouk, R.P. Díaz Redondo, A. Dahou, M. Abd Elaziz, and M. Kayed. Pneumonia detection on chest x-ray images using ensemble of deep convolutional neural networks. *Applied Sciences*, 12 :6448, 2022.
- [34] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets : Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv :1704.04861*, 2017.
- [35] Forrest Iandola, Matthew Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet : Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv :1404.1869*, 2014.
- [36] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16×16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929*, 2020.
- [37] R. Kundu, R. Das, Z. W. Geem, G.-T. Han, and R. Sarkar. Pneumonia detection in chest x-ray images using an ensemble of deep learning models. *PLoS ONE*, 16(9) :e0256630, 2021.
- [38] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8 : Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *Department of Radiology and Imaging Sciences, Clinical Center, National Institutes of Health*, 2017.
- [39] A. R. Aronson and F.-M. Lang. An overview of metamap : historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3) :229–236, May 2010.
- [40] Clément Dalloux. Détection de l’incertitude et de la négation : un état de l’art. In *RECITAL 2017 - 18ème Rencontre des Etudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, pages 1–14, Orléans, France, Jun 2017.
- [41] E. Pérez and S. Ventura. An ensemble-based convolutional neural network model powered by a genetic algorithm for melanoma diagnosis. *Neural Computing and Applications*, 34, 2022.
- [42] J. VanderPlas. *Python Data Science Handbook*. O’Reilly Media, 2016.
- [43] Gary Bradski and Adrian Kaehler. *Learning OpenCV : Computer Vision with the OpenCV Library*. O’Reilly Media, Inc., 2008.
- [44] Khadidjatou Iman BAMBÀ. Les fondamentaux du cloud computing. Mémoire de master en informatique, option : Administration et sécurité des réseaux, Université Kasdi Merbah Ouargla, Faculté des nouvelles technologies de l’information et de la communication, Département d’informatique et technologie de l’information, 2022.
- [45] Peter Mell and Timothy Grance. The nist definition of cloud computing. Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD, USA, sep 2011.

- [46] H. Nadir, S. Khedija, and O. Nacéra. *Etude et comparaison des failles de sécurité d'OpenStack et OpenNebula*. Doctoral dissertation, Université Mouloud Mammeri, 2019.
- [47] Hamza Sahli. *Modélisation des Systèmes Élastiques Cloud : vers la Vérification Formelle de leur Comportement*. Thèse de doctorat, Université Constantine 2 - Abdelhamid Mehri, 2017.
- [48] Hadjer Fellah. *Modélisation des Systèmes Élastiques Cloud : vers la Vérification Formelle de leur Comportement*. Thèse de magister, Université Larbi Ben M'Hidi-Oum El Bouaghi, 2014.
- [49] R. E. G. U. I. E. G. Hicham. *Cloud computing : services informatiques dynamiques basés sur le web - concepts et notions de base*, 2020. Cours.
- [50] M. Zenati and L. Rezzoug. *Conception et réalisation d'un mini-could pour les étudiant de l'université de bejaia*. Master, Université Abderrahmane Mira Bejaia, 2021.

Annexe A

A.1 Arbres de décision

L'arbre de décision est une méthode de classification et de régression très utilisée en Machine Learning. Il construit un modèle sous forme d'une structure arborescente, où chaque nœud interne représente une "question" sur une caractéristique (par exemple, si une variable dépasse un certain seuil), chaque branche représente le résultat de cette question, et chaque feuille représente une classe ou une valeur cible [17].

A.2 Mesures de pureté

Indice de Gini

L'indice de Gini est un critère pour mesurer à quel point un nœud est pur. Si toutes les instances d'un nœud appartiennent à la même classe, alors l'indice de Gini est 0 (pur). Si les instances sont réparties uniformément entre les différentes classes, l'indice de Gini atteint son maximum (imparfait). Le critère de Gini est calculé comme suit :

$$Gini(T) = 1 - \sum_{i=1}^c p_i^2 \quad (4.1)$$

où p_i est la proportion d'instances de la classe i dans le nœud T , et c est le nombre total de classes [14].

Entropie

L'entropie est une autre mesure d'impureté utilisée dans les arbres de décision et elle est basée sur le concept d'entropie en théorie de l'information. L'entropie atteint son minimum (0) lorsque toutes les instances dans un nœud appartiennent à une seule classe, et son maximum lorsque les instances sont uniformément distribuées entre les classes. Elle est calculée comme suit :

$$Entropie(T) = - \sum_{i=1}^c p_i \log_2(p_i) \quad (4.2)$$

où p_i est encore une fois la proportion de la classe i dans le nœud T [14].

Annexe B

B.1 Cloud Computing

Le Cloud signifie « nuage » et Computing « informatique », le Cloud Computing est donc l'informatique en nuage pour une traduction littérale anglais français [44].

Différentes définitions ont été proposées pour définir le cloud, ce qui a engendré une certaine confusion quant à la compréhension de ce qu'est le cloud. Pour cet effet, conformément au National Institute of Standards and Technology (NIST) :

Définition. « *L'informatique en nuage est un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peut être rapidement provisionné et libéré avec un minimum d'effort de gestion ou d'interaction avec le fournisseur de services. Ce modèle de nuage se compose de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement* » [45].

En d'autres termes, le Cloud Computing est un modèle informatique qui permet un accès à la demande et en libre-service à des ressources informatiques configurables et partagées sur un réseau de communication.

B.1.1 Caractéristiques du Cloud Computing

Les principales caractéristiques du cloud computing sont [46] :

- **Service à la demande**

La disponibilité des ressources informatiques en fonction des besoins du client, automatiquement, sans nécessiter l'interaction humaine avec chaque fournisseur de services.

- **Large accès au réseau**

Ces fonctionnalités sont disponibles sur le réseau et sont accessibles via des mécanismes standard prenant en charge leur utilisation par des plates-formes de clients légers ou lourds disparates (telles que les téléphones mobiles, les tablettes et les ordinateurs portables, etc.).

- **Ressources partagées**

Les ressources, qu'elles soient physiques ou virtuelles, sont allouées et réallouées dynamiquement en fonction de la demande et partagées entre les différents utilisateurs du service.

- **Elasticité**

L'élasticité est fournie dans le cloud selon une stratégie réactive ou proactive, à l'aide de trois méthodes : **La mise à l'échelle horizontale** implique l'ajout ou la suppression d'instances virtuelles pour ajuster la capacité des services cloud. **La mise à l'échelle verticale**, consiste à augmenter ou diminuer les ressources, comme la RAM ou le CPU. Enfin, la méthode de **migration** consiste à déplacer ou redéployer une instance d'un hôte vers un autre hôte [47].

- **Service mesuré**

L'utilisation des ressources peut être surveillée, contrôlée et signalée en exploitant une capacité de mesure à un niveau d'abstraction approprié au type de service, ce qui assure la transparence tant pour le fournisseur que pour le consommateur du service utilisé.

- **Paiement à l'usage (pay-as-you-use)**

Le coût est proportionnel à l'usage, donc l'utilisateur paye pour exactement ce qu'il utilise [48].

B.1.2 Types de cloud :

Selon la description du Cloud Computing fournie par le NIST, il y a quatre types du Cloud : le Cloud privé, le Cloud communautaire, le Cloud public et le Cloud hybride [49]. Grâce à ces modèles, il est possible de spécifier de quelle manière l'utilisateur final peut accéder aux fournisseurs de services Cloud.

1. **Cloud privé** : Un cloud privé représente un ensemble de services et de ressources dédiés à un seul client, tel qu'une entreprise. Cela peut être défini comme un réseau informatique propriétaire ou un centre de données qui fournit des services hébergés à un nombre limité d'utilisateurs. Ce modèle, également appelé cloud interne, signifie que le fournisseur et l'utilisateur appartiennent à la même unité organisationnelle, telle qu'une entreprise, un gouvernement, etc. Dans ce cadre, le contrôle des données demeure entre les mains des utilisateurs ou de leur organisation. L'avantage de ce cloud est lié à la sécurité et à la protection des données.
2. **Cloud publics** : Les cloud publics, également appelés cloud externes, regroupent des services cloud dans lesquels les fournisseurs et les utilisateurs ne font pas partie de la même unité organisationnelle. Dans ce modèle, les prestataires rendent leur Cloud accessible au public via un portail Web en libre-service. Les entreprises peuvent externaliser leurs ressources informatiques pour réduire les coûts et les efforts liés à la création d'une infrastructure interne ou pour fournir des services uniques aux utilisateurs externes. C'est un modèle de Cloud Computing courant, avec des services proposés par des prestataires externes.
3. **Le Cloud communautaire** : est un modèle de cloud computing partagé par plusieurs organisations au sein d'une communauté ayant des intérêts communs, tels que des problèmes de réglementation ou de sécurité des données. L'infrastructure peut être gérée par l'organisation elle-même ou par un tiers et est utilisée de manière décentralisée par les membres de la communauté.
4. **Le Cloud hybride** : est une infrastructure composée de deux ou plusieurs environnements cloud distincts, qu'ils soient privés, communautaires, ou publics. Cette approche permet

d'optimiser les coûts en tirant parti de l'externalisation propre aux Clouds publics tout en conservant un niveau de contrôle souhaité sur les données sensibles.

B.1.3 Architecture de Cloud Computing

L'architecture cloud représente la façon dont différents composants technologiques s'assemblent pour former un environnement cloud, où les ressources sont consolidées grâce à la virtualisation et partagées à travers un réseau. L'architecture du Cloud Computing est divisée en deux parties : **Frontend et Backend** [50].

- **Frontend** : Représente l'infrastructure client, contenant des applications et des interfaces qui sont nécessaires pour accéder à la plateforme Cloud, autrement dit, il fournit une GUI (Interface Utilisateur Graphique), pour interagir avec le Cloud.
- **Backend** : Fait référence au Cloud Computing lui-même, qui comprend un énorme stockage, des applications virtuelles, des Machines virtuelles, des mécanismes de sécurités, des modèles de déploiement, etc.

La figure 4.9 représente l'architecture de cloud computing simplifiée, montrant la séparation entre le front end (interface client), l'internet, et le back end (serveurs cloud) qui gère l'application, le service, la sécurité, et l'infrastructure de stockage.

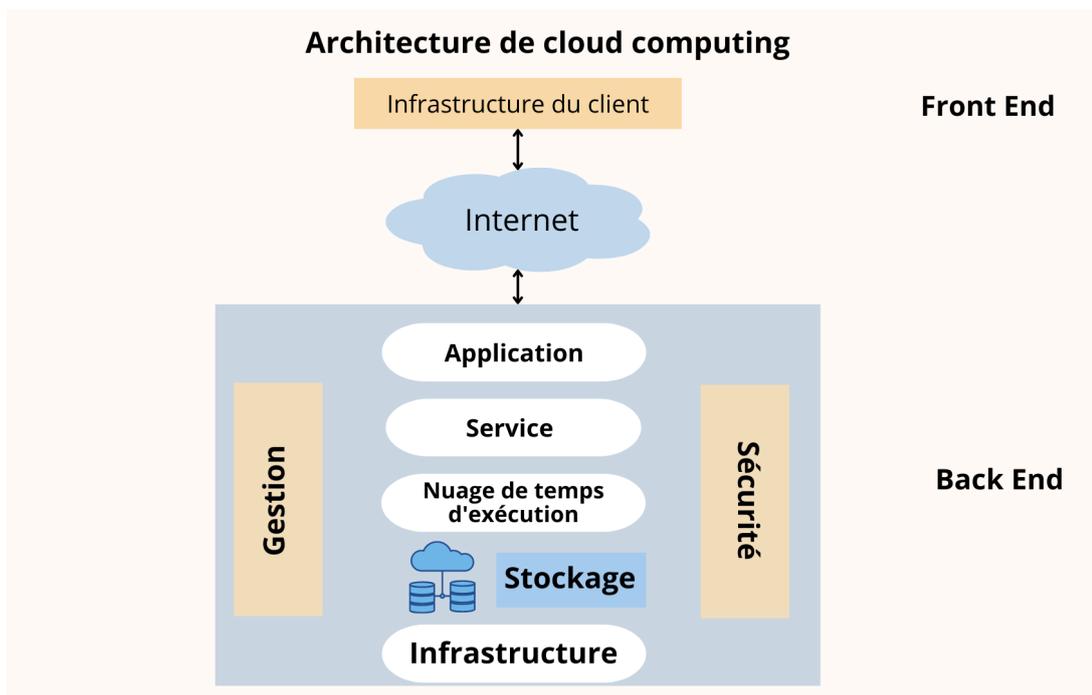


FIGURE 4.9 – Architecture du cloud computing

Quant à ses composants, on a :

- **Applications** : Se réfère à un logiciel ou une plate-forme à laquelle le client accède.
- **Service** : Gère le type de service auquel l'utilisateur accède et se réfère aux trois principaux types de services basés sur le cloud SaaS (Software as a Service), PaaS (Platform as a Service) et IaaS (Infrastructure as a Service).

-
- **Nuage de temps d'exécution** : C'est l'environnement d'exécution pour les applications et les services.
 - **Stockage** : C'est l'un des composants les plus importants, il fournit une énorme capacité de stockage pour stocker et gérer les données.
 - **L'infrastructure** : Fait référence aux composants matériels et logiciels du cloud tels que les serveurs, le stockage, les périphériques réseau, les logiciels de virtualisation nécessaires pour le modèle Cloud Computing.
 - **Gestion** : Est utilisé pour la gestion des composants backend tels que l'application, le service, le runtime Cloud, le stockage, l'infrastructure et d'autres mécanismes de sécurité, et établit une coordination entre eux.
 - **Sécurité** : Il met en œuvre un mécanisme de sécurité dans le Backend.

D'autant plus, on a le composant **Internet** qui fait référence à la connexion internet agit comme un pont et établit l'interaction et la communication entre le Frontend et le Backend.

Résumé

Ce mémoire explore l'application du Machine Learning (ML) dans le diagnostic de la pneumonie à partir d'un échantillon d'images étiquetées, en prenant comme cas d'étude le Centre Hospitalier Universitaire (CHU) de Béjaïa. L'objectif est de mettre en oeuvre une méthode automatique pour classifier les images à l'une des classes établies au préalable. La contribution principale de ce travail réside dans l'évaluation d'une approche hybride qui combine des réseaux de neurones convolutifs (CNN), particulièrement les architectures VGG16 et VGG19, avec divers algorithmes d'apprentissage automatique tels que SVM et KNN. Cette méthode est utilisée pour extraire les caractéristiques et classer correctement les images pulmonaires. Les résultats montrent que l'utilisation du Machine Learning améliore significativement la précision et la rapidité du diagnostic de la pneumonie, démontrant ainsi le potentiel de ces technologies pour optimiser les pratiques médicales traditionnelles. Ainsi, l'efficacité des méthodes hybrides dans l'amélioration des performances diagnostiques, tout en ouvrant la voie à des études futures pour optimiser davantage ces techniques.

Mots clés : Machine Learning, SVM, KNN, CNN, VGG16, VGG19, Classification, Pneumonie.

Abstract

This thesis explores the application of Machine Learning (ML) in diagnosing pneumonia using a labeled set of images, focusing on the University Hospital Center (CHU) of Béjaïa as a case study. The goal is to implement an automatic method to classify the images into pre-defined categories. The main contribution of this work lies in evaluating a hybrid approach that combines Convolutional Neural Networks (CNNs), specifically the VGG16 and VGG19 architectures, with various machine learning algorithms such as SVM and KNN. This method is used to extract features and accurately classify pulmonary images. The results demonstrate that the use of Machine Learning significantly enhances the accuracy and speed of pneumonia diagnostics, thus showcasing the potential of these technologies to optimize traditional medical practices. The effectiveness of hybrid methods in improving diagnostic performances paves the way for future studies to further optimize these techniques.

Key words : Machine Learning, SVM, KNN, CNN, VGG16, VGG19, Classification, Pneumonia.