الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique Et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère De L'enseignement Supérieur Et De La Recherche Scientifique

Université Constantine 1 Frères Mentouri
Faculté des Sciences de la Nature et de la Vie

جامعة قسنطينة 1 الإخوة منتوري
كلية علوم الطبيعة والحياة

Département : Biologie Appliquée

قسم : البيولوجيا التطبيقية

Mémoire présenté en vue de l'obtention du Diplôme de Master

Domaine : Sciences de la Nature et de la Vie

Filière : Biotechnologies

Spécialité : Bioinformatique

N° d'ordre :
N° de série :

Intitulé :

Analyse de données bioinformatiques et prédiction par approches basées sur l'IA dans le diagnostic du cancer pulmonaire non à petites cellules.

Présenté par : SAKRAOUI Chourouk & MAYOUF Roua

Le : 25/06/2025

Jury d'évaluation :

Président : AMINE KHODJA Ihsene Rokia (MCB)

Encadrant : BENSAADA Mostafa (MCA)

Examinateur : Mohamed Skander DAAS (MCA)

Année Universitaire : 2024 / 2025

REMERCIEMENTS

Nous exprimons avant tout notre gratitude à **ALLAH,** Le Tout-Puissant, pour nous avoir guidés tout au long de ce travail, et pour nous avoir accordé la force, la patience et la persévérance nécessaires à l'accomplissement de ce mémoire.

Nous tenons ensuite à exprimer notre profonde reconnaissance à **notre encadrant, BENSAADA Mostafa**, pour leur encadrement, leurs conseils avisés, leur disponibilité constante et leur soutien tout au long de la réalisation de ce travail.

Nous adressons nos sincères remerciements à **Madame Pr. Beddar**, cheffe du Service Central d'Anatomie Pathologique et de Cytologie du Centre Hospitalo-Universitaire (CHU) de Constantine, ainsi qu'à son équipe, pour leur accueil chaleureux, leur disponibilité et leur précieuse collaboration. Grâce à leur soutien, nous avons pu accéder à des données réelles essentielles à la validation de notre travail de recherche

Nous remercions également l'ensemble des **enseignants et membres du jury** de la spécialité Bioinformatique pour la qualité de l'enseignement dispensé et pour leurs retours constructifs.

Enfin, nous exprimons notre profonde reconnaissance à nos **familles et proches**, pour leur soutien moral inestimable, leur patience et leur encouragement durant toutes ces années d'études.

DÉDICACE

Je dédie ce mémoire à :

Ma chère famille, à **mon père**, **ma mère** et **ma sœur**,
Pour leur amour inconditionnel, leurs encouragements et leur soutien inestimable tout au long de mon parcours.

Mon binôme **Roua**,
Pour avoir été une partenaire de confiance, toujours présente, persévérante et engagée.
Merci pour ton sérieux, ta bonne humeur, ton esprit d'équipe et ton soutien tout au long de ce travail.

Mes amies proches, **Chayma** et **Assala**,
Pour leur soutien moral, leur précieuse amitié et les moments de joie partagés. Votre présence a été un vrai réconfort.

À toutes les personnes qui m'ont soutenu(e), de près ou de loin,
Par un mot, un geste, ou simplement par leur présence bienveillante,
Je vous adresse toute ma gratitude.

SAKRAOUI Chourouk

DÉDICACE

Je dédie ce modeste travail à ceux qui m'ont tous donné sans rien me demander, mes très chers parents.

A la lune de ma vie, mon très cher père, qui a toujours fait preuve d'amour et de courage, pour m'assurer une bonne éducation. Qu'il puisse jouir de ce qui a été le fruit de ses sacrifices, je n'ai rien de plus cher à lui dédier que ce modeste mémoire en témoignage de ma gratitude.

A l'étoile de ma vie, ma très chère mère, qui m'a toujours été d'un grand soutien moral, par ses encouragements, ses conseils. Elle a toujours demeuré dévouée à mon bien être, ce témoignage est guise de reconnaissance de ma part.

A ma chère sœur Imane et mon cher frère Issam

A toute ma famille sans exception

A mes amies

A mon binôme Chourouk, avec qui j'ai partagé de précieux souvenirs. Elle a toujours été à mes côtés depuis la licence, traversant avec moi les bons comme les mauvais moments. Elle restera à jamais gravée dans ma mémoire.

A tous les gens qui ont participé à l'élaboration de cette mémoire de près ou de loin.

Je tiens enfin à exprimer toute ma gratitude à toute la promotion 2024/2025 **Bioinformatique.**

MAYOUF Roua

RÉSUMÉ

Le cancer du poumon non à petites cellules (CPNPC) constitue un enjeu crucial en oncologie, particulièrement pour ce qui est d'établir un diagnostic rapide et exact. Le travail de recherche que nous exposons dans ce mémoire aborde la création et l'implémentation de DiagnoLung, une plateforme interactive fondée sur l'intelligence artificielle, plus précisément sur des modèles multimodaux dans l'aide à la détection et au diagnostic du CPNPC. DiagnoLung combine et analyse des données cliniques ainsi que des images médicales, comme les images histopathologiques de biopsies de tumeurs solides et les tomodensitométries thoraciques (CT Scan), dans le but de soutenir les oncologues dans la détection précoce et le diagnostic du CPNPC. En intégrant diverses sources d'informations médicales, et l'intelligence artificielle nous participons à l'amélioration du diagnostic par une optimisation de la décision clinique dans le contexte du cancer du poumon et plus largement les autres cancers solides.

Mots clés : Cancer du poumon, CPNPC, Intelligence artificielle, Modèles multimodaux, Données cliniques, Imagerie médicale.

ABSTRACT

Non-small cell lung cancer (NSCLC) represents a critical challenge in oncology, particularly in establishing a rapid and accurate detection and diagnosis. The research work presented in this thesis addresses the development and implementation of DiagnoLung, an interactive artificial intelligence-based platform, specifically leveraging multimodal models to assist in the **early** detection and diagnosis of NSCLC. DiagnoLung integrates and analyzes clinical data along with medical imaging, such as histopathological images from solid tumor biopsies and chest CT scans, to support oncologists in the detection and diagnosis of NSCLC. By combining diverse sources of medical information and artificial intelligence, we contribute to improving diagnostic accuracy by optimizing clinical decision-making in the context of lung cancer and, more broadly, other solid cancers.

Keywords **:** Lung cancer, NSCLC, Artificial intelligence, Multimodal models, Clinical data, Medical imaging.

الملخص

سرطان الرئة ذو الخلايا غير الصغيرة (CPNPC) يمثل أحد التحديات الكبرى في علم الأورام، لا سيما عند السعي إلى تحقيق كشف وتشخيص دقيق وسريع. تتناول هذه الأطروحة تطوير وتنفيذ"DiagnoLung" ، وهي منصة تفاعلية قائمة على الذكاء الاصطناعي، تعتمد على نماذج متعددة الوسائط تهدف إلى دعم الكشف المبكر وتشخيص سرطان الرئة ذو الخلايا غير الصغيرة.

تُوظّف DiagnoLung دمجًا متكاملًا بين البيانات السريرية والصور الطبية، مثل صور الخزعات النسيجية للأنسجة الورمية الصلبة وصور التصوير المقطعي المحوسب (CT Scan) للصدر، بهدف تعزيز دعم القرار السريري لأطباء الأورام. ومن خلال هذا التكامل بين مصادر متعددة للمعلومات الطبية وتقنيات الذكاء الاصطناعي، تسعى المنصة إلى تحسين دقة وفاعلية الكشف والتشخيص، ليس فقط في حالات سرطان الرئة، بل أيضًا في سرطانات صلبة أخرى.

.

الكلمات المفتاحية : سرطان الرئة، سرطان الرئة ذو الخلايا غير الصغيرة، الذكاء الاصطناعي، نماذج متعددة الوسائط، البيانات السريرية، التصوير الطبي.

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATION

3D : Trois Dimensions
AIDA : Artificial Intelligence Drug Advisor
AI-DSS : Artificial Intelligence Decision Support System
AUC : Area Under the Curve
API : Application Programming Interface
CART : Classification and Regression Trees
CA-ResNet50 : Variante de ResNet50
CBNPC : Cancer Bronchique Non à Petites Cellules
CNN : Convolutional Neural Network
CONV : Convolutional Layer
CPNPC : Cancer du Poumon Non à Petites Cellules
CPPC : Cancer du Poumon à Petites Cellules
CSS : Cascading Style Sheets
CT Scan : Computed Tomography Scan
df : DataFrame
DL : Deep Learning
DenseNet121 : Densely Connected Network 121
DT2 : Diabète de Type 2
ECG : Électrocardiogramme
ELM : Extreme Learning Machine
FC : Fully Connected
FN : Faux Négatifs
FP : Faux Positifs
GB : Giga-octet
GlobalAveragePooling2D : Couches de réduction spatiale moyennée
GPS : Global Positioning System
H&E : Hématoxyline et Éosine
HIELCC-EDL : Hybrid Intelligent Ensemble Learning for Cancer Classification using Evolutionary Deep Learning
HTML : HyperText Markup Language
IA : Intelligence Artificielle
IRM : Imagerie par Résonance Magnétique
JPEG : Joint Photographic Experts Group
JS : JavaScript
LiDAR : Light Detection and Ranging
LSTM : Long Short-Term Memory
MB : Méga-octet
ML : Machine Learning
MML : Multimodal
NC : Normalisation des Couleurs
np : numpy
OMS : Organisation Mondiale de la Santé
pd : pandas
plt : matplotlib.pyplot
POOL : Pooling Layer
ReLU : Rectified Linear Unit
ResNet50 : Residual Network 50
RF : Random Forest
RGB : Red-Green-Blue

RNA : Réseau Neuronal Artificiel
ROC : Receiver Operating Characteristic
sklearn : scikit-learn
SPI : Système de Prescription Intelligente
SVM : Support Vector Machine
TAL : Traitement Automatique du Langage
tf : tensorflow
TNM : Tumor, Node, Metastasis
UI : User Interface
VS Code : Visual Studio Code
VN : Vrais Négatifs
VP : Vrais Positifs

TABLE DES MATIERES

Introduction Générale

Introduction Générale

Le cancer des poumons est un problème de santé publique à l'échelle mondiale. Selon les estimations les plus récentes de l'Organisation Mondiale de la Santé (OMS, 2023), il s'agit de la principale cause mondiale de décès liés au cancer, représentant environ 1,8 million de décès par an, en soit plus que le total combiné des cancers de la prostate, du côlon et du sein [1].

Parmi les divers types histologiques du cancer du poumon, le cancer du poumon non à petites cellules (CPNPC) qui est de loin le plus fréquent, représentant 80 à 85 % des cas diagnostiqués. Malgré les avancées thérapeutiques récentes, son pronostic reste sombre, avec un taux de survie à 5 ans inférieur à 20 % aux stades avancés, principalement en raison d'un diagnostic souvent tardif [2].

Actuellement, le diagnostic de CPNPC repose principalement sur l'examen histopathologique des biopsies, méthode qui présente plusieurs limites majeures. L'interprétation des lames par les anatomo-pathologistes ne peut être que subjective et des taux de discordance inter-observateurs de 30 % ont été constatés pour certains sous-types histologiques [3]. Ensuite, la démarche diagnostique classique ne tient compte des données cliniques que partiellement (âge, tabagisme, marqueurs moléculaires) ; alors qu'elles renferment des informations pronostiques primordiales. Enfin, dans certains contextes, le délai entre le prélèvement et le rendu du diagnostic définitif peut atteindre plusieurs semaines, retardant d'autant la mise en route du traitement. En raison de ces limites les apports de l'intelligence artificielle (IA), et plus précisément les approches multimodales, ouvrent des perspectives qui réduisent les inconvénients de l'analyse histopathologique. Sans omettre de signaler que la démarche basée sur les techniques IA pose elle aussi la question de : « Comment développer un système de diagnostic assisté par IA qui, combine de manière optimale l'analyse des images CT-scan, histopathologiques et des données cliniques, pourrait-il améliorer à la fois la précision, la rapidité et la personnalisation du diagnostic du CPNPC ?

Pour répondre à cette question nous ambitionnons dans notre travail de développer et valider un modèle d'aide au diagnostic innovant combinant de manière synergique :

- L'analyse des données cliniques des patients, à l'aide de techniques d'apprentissage automatique.
- L'analyse automatisée des images de tomodensitométrie (CT-scan), au moyen de modèles profonds pré-entraînés par apprentissage par transfert, pour la détection d'anomalies pulmonaires.

- La fusion des données cliniques et des images CT-scan, par des architectures multimodales, afin d'améliorer la précision diagnostique.
- L'analyse automatique des images histopathologiques, au moyen de réseaux de neurones convolutifs profonds pré-entraînés par apprentissage par transfert, pour détecter le cancer pulmonaire non à petites cellules (CPNPC) et en identifier les sous-types.
- Le déploiement de l'ensemble du modèle sur une plateforme web DiagnoLung, facilitant le travail des professionnels de santé.

Recherche Bibliographique

Chapitre 1 : Généralités sur le cancer du poumon

Chapitre 1 : Généralités sur le cancer du poumon

1 Poumons

Les poumons organes vitaux sont situés dans le thorax, de chaque côté du cœur. Ils servent à respirer et les échangent gazeux indispensables à la vie.

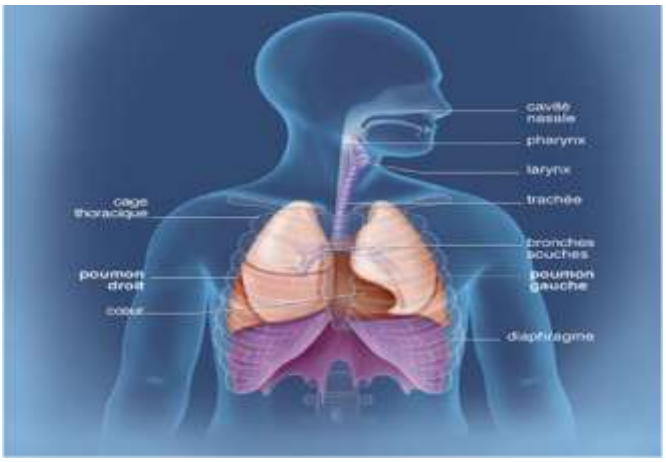


Figure 1: Poumons dans le corps humain [4].

Les poumons sont divisés en plusieurs lobes, eux-mêmes divisés en plusieurs segments. Le poumon gauche comprend deux lobes et le poumon droit en compte trois.

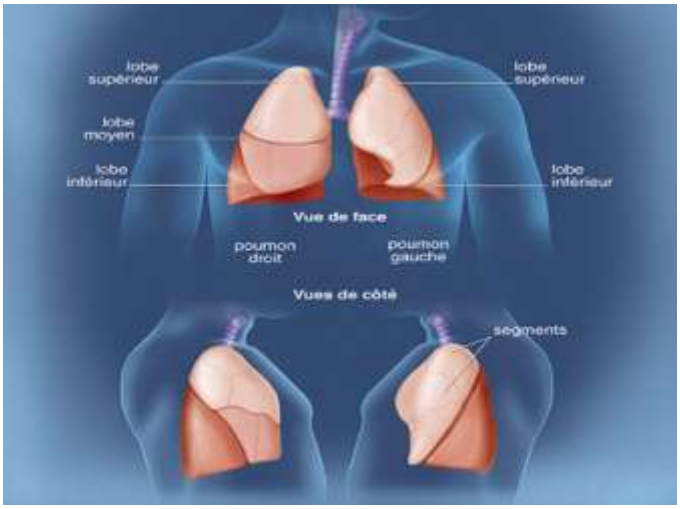


Figure 2 : Structure des poumons [4].

L'air inspiré par le nez et par la bouche apporte à toutes les cellules du corps l'oxygène nécessaire à leur fonctionnement. Il circule dans la trachée qui se divise à droite et à gauche en deux bronches souches. Ces bronches souches se ramifient dans les poumons en bronches, puis en bronchioles. Elles se terminent par des alvéoles pulmonaires, petites cavités où ont lieu les échanges gazeux entre l'air respiré et le sang.

L'oxygène contenu dans l'air inspiré traverse la paroi des alvéoles pour passer dans le sang. Le sang distribue ensuite l'oxygène à toutes les cellules de l'organisme.

Dans le même temps, en sens inverse, le gaz carbonique rejeté par toutes les cellules du corps est ramené par le sang jusqu'aux poumons. Il traverse la paroi des alvéoles et passe par les bronches. A l'expiration, l'air est évacué par la trachée, puis le nez ou la bouche.

Les poumons sont protégés par la cage thoracique qui est délimitée notamment par les côtes. Ils sont enveloppés par la plèvre. Entre les deux poumons, se situe la région du médiastin qui s'étend du sternum à la colonne vertébrale. Le médiastin contient le cœur, de gros vaisseaux sanguins, la trachée et l'œsophage. Il comprend également les ganglions lymphatiques médiastinaux. Ces ganglions font partie du système lymphatique, dont le rôle est d'évacuer les déchets émis par l'organisme grâce à un liquide, la lymphe. Les ganglions médiastinaux peuvent être atteints par les cellules cancéreuses [4].

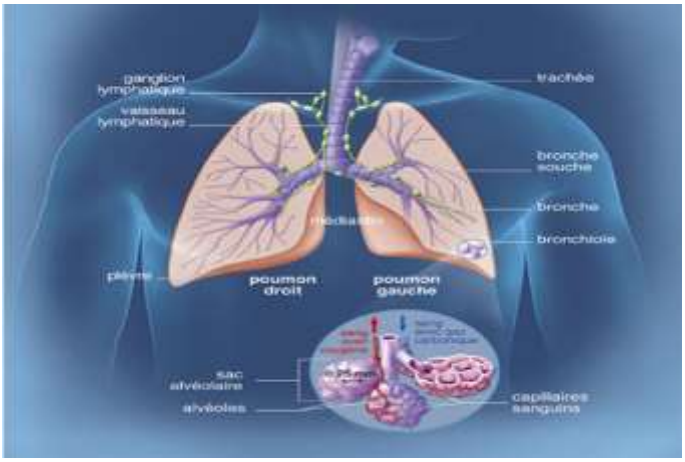


Figure 3 : À l'intérieur des poumons [4].

2 Cancer du poumon

Un cancer du poumon, appelé aussi cancer bronchique ou cancer bronchopulmonaire, touche les cellules bronchiques ou, plus rarement, des cellules qui tapissent les alvéoles pulmonaires. Il se développe à partir d'une cellule initialement normale qui se transforme et se multiplie de façon anarchique, jusqu'à former une masse appelée tumeur maligne.

3 Symptômes

Le cancer du poumon peut causer plusieurs symptômes susceptibles d'indiquer un problème dans les poumons. Les symptômes les plus courants sont les suivants :

- Une toux qui ne disparaît pas ;
- Des douleurs thoraciques ;
- Un essoufflement ;
- Expectoration de sang (hémoptysie) ;
- Fatigue intense ;
- Perte de poids sans cause connue ;
- Infections pulmonaires qui reviennent sans cesse.

Les premiers symptômes peuvent être légers ou considérés comme des problèmes respiratoires courants, ce qui retarde le diagnostic [5].

4 Prévention

Le fait d'éviter de fumer du tabac est le meilleur moyen de prévenir le cancer du poumon. Les autres facteurs de risque à éviter sont les suivants :

- La fumée secondaire ;
- La pollution de l'air ;
- Les dangers en milieu professionnel comme les produits chimiques et l'amiante.

Un traitement précoce peut empêcher le cancer du poumon de s'aggraver et de se propager à d'autres parties de l'organisme. La prévention du cancer du poumon comprend des mesures de prévention primaire et secondaire [5].

4.1 Prévention primaire

Elle vise à prévenir l'apparition initiale d'une maladie par la réduction des risques et la promotion d'un comportement sain. En santé publique, ces mesures préventives comprennent le sevrage tabagique, la promotion d'environnements sans tabac, la mise en œuvre de politiques de lutte antitabac, la lutte contre les risques professionnels et la réduction des niveaux de pollution de l'air [5].

4.2 Prévention secondaire

Elle suppose l'application des méthodes de dépistage qui visent à détecter la maladie dès les premiers stades, avant que les symptômes ne deviennent apparents, et ils peuvent être indiqués pour les personnes à haut risque. Dans cette population, la détection précoce peut augmenter

considérablement les chances de succès du traitement et améliorer les résultats. La principale méthode de dépistage du cancer du poumon est la tomodensitométrie à faible dose [5].

5 Diagnostic

Les méthodes de diagnostic du cancer du poumon comprennent l'examen clinique, l'imagerie (comme les radiographies pulmonaires, la tomodensitométrie et l'imagerie par résonance magnétique), l'examen de l'intérieur du poumon à l'aide d'une bronchoscopie, le prélèvement d'un échantillon de tissu (biopsie) pour l'examen histopathologique et la définition du sous-type spécifique (carcinome non à petites cellules ou carcinome à petites cellules), et les tests moléculaires pour identifier des mutations génétiques spécifiques ou des biomarqueurs [6].

6 Types de cancer du poumon

Le cancer du poumon est un groupe hétérogène de cancers, généralement réparti en cancers du poumon à petites cellules (CPPC), qui représentent environ 15 % de tous les cas de cancer du poumon, et en cancers non à petites cellules (CPNPC) [7].

Dans le cadre de ce mémoire, nous nous concentrerons exclusivement sur le CPNPC, car il constitue le sujet principal de notre étude.

6.1 Cancer du poumon non à petites cellules

Les cancers bronchiques non à petites cellules (CBNPC) représentent près de 85 % des cancers du poumon. Les formes les plus fréquentes de cancers bronchiques non à petites cellules (CBNPC) sont :

- **L'adénocarcinome bronchique :** prend souvent naissance en périphérie des poumons
- **Le carcinome épidermoïde :** se développe habituellement dans les grosses bronches situées dans la partie centrale du poumon.
- **Le carcinome à grandes cellules :** peut siéger dans toutes les parties du poumon [8].

6.1.1 Différents stades du CPNPC

La classification dite TNM (Tumor, Node, Metastasis) comme système de stratification est utilisée dans la majorité des cancers afin de classer, par stade, du plus précoce au plus évolué, les cancers. C'est une classification clinique établie selon:

- T: la taille de la tumeur ;
- N: l'atteinte ganglionnaire ;
- M: l'atteinte métastatique**.**

Les tumeurs sont par la suite classées selon 5 stades allant de 0 à 4. On retrouve souvent le stade inscrit en chiffre romain. En règle générale, plus le stade est élevé, plus le cancer est étendu. Dans le cadre du CBNPC, la dernière révision de la classification, 8ème édition, a été publiée en 2016.

Dans le langage courant, lorsque les médecins évoquent le stade d'un cancer, ils peuvent également employer les termes de :

- Local : le cancer se situe uniquement dans le poumon ;
- Régional : le cancer a envahi les ganglions lymphatiques ou d'autres régions thoraciques du côté de la tumeur initiale ;
- Distant : le cancer a envahi d'autres parties du corps en dehors du thorax.

1) **Stade 0** : Des cellules cancéreuses sont présentes dans le revêtement de la voie respiratoire ou des sacs alvéolaires du poumon uniquement.

2) **Stade IA :** La tumeur est située dans le poumon, sa taille mesure jusqu'à 3 cm. Il n'y a pas d'atteinte ganglionnaire ni métastatique à distance.

- Stade IA1 : taille de la tumeur < 1 cm ;
- Stade IA2 : taille de la tumeur comprise entre 1 et 2 cm ;
- Stade IA3 : taille de la tumeur comprise entre 2 et 3 cm.

3) **Stade IB** : La taille de la tumeur pulmonaire est comprise entre 3 et 4 cm (T2A) ; Il n'y a pas d'atteinte ganglionnaire ni métastatique à distance.

4) **Stade IIA** : La tumeur est classée 2A si :

- la taille de la tumeur pulmonaire est comprise entre 4 cm et 5 cm (T2b) ;
- ou elle a envahi la principale voie respiratoire (hors région où la trachée se divise en bronches souches) ou la plèvre viscérale ;
- ou il a engendré l'affaissement d'un poumon, bloqué une bronche ou provoqué l'inflammation des tissus pulmonaires d'une partie ou de la totalité du poumon.

5) **Stade IIB** : La tumeur est classée IIB si :

- La tumeur mesure 5 cm ou moins (T2), mais a envahi les ganglions lymphatiques à proximité des bronches (N1) ;

ou si elle comporte l'un de ces critères :

- La tumeur mesure entre 5 et 7 cm (T3) sans envahissement ganglionnaire lymphatique ou à distance ;
- Les cellules cancéreuses se sont propagées à la membrane externe qui tapisse les poumons, soit à la plèvre pariétale, la paroi thoracique, le nerf principal qui se rend jusqu'au diaphragme,

soit le nerf phrénique, ou à la membrane externe qui recouvre le cœur (feuillet pariétal du péricarde) ; présence de 2 tumeurs dans un même lobe du poumon.

6) Stade IIIA : La tumeur est classée IIIA si :

- La tumeur mesurant jusqu'à 5 cm de grand axe, mais a envahi les ganglions lymphatiques proches de la trachée du côté de la tumeur ou ceux présents près de la région où la trachée se divise en bronche souche gauche et en bronche souche droite, ou bien à tous ces ganglions ;

Ou la tumeur mesure plus de 5 cm et comporte l'un de ces critères :

- La tumeur a envahi les ganglions lymphatiques situés près des bronches ;
- La tumeur s'est propagée à l'une de ces parties du corps : le diaphragme, le médiastin, le cœur ou les gros vaisseaux sanguins près du cœur, la trachée, un nerf principal qui se rend au larynx, l'œsophage, une vertèbre ou la région où la trachée se divise en bronches souches ;
- La présence d'une autre tumeur dans le même poumon.

7) Stade IIIB

- La tumeur mesure jusqu'à 5 cm ou moins et a envahi les ganglions lymphatiques du côté opposé de la trachée ou du poumon ou aux ganglions lymphatiques situés dans la partie inférieure du cou ;
- La tumeur mesure plus de 5 cm avec au moins une autre tumeur dans le même poumon (T3). Le cancer a envahi les ganglions lymphatiques (soit ceux situés à côté de la trachée du même côté du corps que la tumeur, soit ceux situés sous la région où la trachée se divise en bronches souches, soit l'ensemble de tous ces ganglions, N1 ou N2).

8) Stade IIIC

- La taille de la tumeur est supérieure à 5 cm (T3) ; ou il y a plus d'une tumeur dans un lobe différent du même poumon(T4) ;
- Le cancer a par ailleurs envahi les ganglions lymphatiques du côté opposé de la trachée, ou du poumon, ou ceux situés dans la partie inférieure du cou (N3).

9) Stade IV

- Présence de métastases à distance révélant l'extension du cancer à d'autres parties du corps. Le terme de cancer du poumon non à petites cellules métastatiques est aussi employé.

10) Stade IVA

- Le cancer a envahi l'autre poumon (M1a) ;

- Ou le cancer a envahi la plèvre ou le péricarde (M1a) ; on observe un épanchement pleural (M1a) ; le cancer s'est propagé avec présence d'une nouvelle tumeur qui se développe en dehors du thorax (M1b).

11) Stade IVB

- Le cancer s'est propagé avec présence d'au moins 2 autres tumeurs qui se développent en dehors du thorax (M1c) [9].

6.1.2 Traitement du CPNPC

Les CPNPC de stade I et II sont généralement traités dans un but curatif, par intervention chirurgicale ou, si elle n'est pas possible, par radiothérapie stéréotaxique ablative. Le risque de récidive commence à augmenter à partir du moment où la taille de la tumeur devient supérieure à 4 cm ou s'il y a atteinte aux ganglions lymphatiques. Dans un tel cas, on peut administrer 4 cycles de chimiothérapie adjuvante en doublets de platine, qui comportent la cisplatine combinée à un autre médicament, comme la vinorelbine ou le pémétrexed.

Le quart des patients se présentent avec un cancer de stade III (soit avec des tumeurs plus volumineuses ou avec une atteinte aux ganglions lymphatiques médiastinaux). Si ces tumeurs ne sont pas résécables, une radiation thoracique est indiquée avec une chimiothérapie simultanée pour la synergie ; elles seront suivies d'une immunothérapie adjuvante pendant 1 an. Une récente étude a démontré une meilleure survie sans incident lorsque les stades II et III d'un CPNPC résécable étaient traités avec 3 cycles de chimiothérapie néoadjuvante et l'immunothérapie avant l'intervention chirurgicale, une pratique qui deviendra probablement courante à l'avenir.

Chapitre 2 : L’intelligence artificielle et Multimodalité

Chapitre 2 : L’intelligence artificielle et Multimodalité

1 Définition et importance de l’IA en médecine

L’intelligence artificielle est une discipline scientifique qui vise à concevoir des algorithmes capables de simuler l’intelligence humaine, notamment la perception, le raisonnement, l’apprentissage, la planification et la prédiction [10]. Elle correspond au développement de systèmes informatiques capables d’exécuter des tâches nécessitant habituellement l’intelligence humaine, comme la reconnaissance de formes, l’apprentissage à partir de données et la prise de décision [11].

Ce domaine multidisciplinaire intègre l’informatique, l’analyse de données, les statistiques, l’ingénierie, la linguistique, les neurosciences, la philosophie et la psychologie. L’IA est un paradigme interdisciplinaire, fruit des transformations issues des sciences cognitives, des neurosciences, de l’informatique et de la robotique [12].

Sur le plan opérationnel, l’IA s’appuie principalement sur les apprentissages automatique (machine learning) et profond (deep learning), utilisés pour l’analyse de données, la prédiction, la classification, le traitement du langage naturel, les systèmes de recommandation et la recherche d’informations [13].

2 Approches d’intelligence artificielle

Afin de comprendre les avantages que l’intelligence artificielle apporte en termes de diagnostic médical, il est important d’expliquer les deux techniques dont elles sont constituées et qui sont l’apprentissage automatique (ML) et l’apprentissage profond (DL). Bien qu’elles soient différentes, nous expliquerai leurs principes ci-dessous :

2.1 Apprentissage automatique

L'apprentissage automatique est un sous-ensemble de l'IA qui consiste à créer des modèles informatiques capables d'apprendre et de formuler des prédictions ou des décisions indépendantes à partir des données fournies. Ces modèles améliorent continuellement leur précision grâce aux données apprises [14]. Pour cela, il existe trois grandes méthodes d’apprentissage : L’apprentissage supervisé, l’apprentissage non supervisé et l’apprentissage par renforcement.

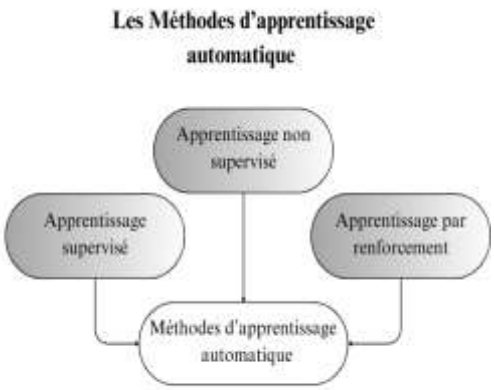


Figure 4: Méthodes d’apprentissage automatique.

La méthode la plus utilisée dans ce travail est l'apprentissage supervisé, particulièrement la classification, qui constitue l’un des axes majeurs de notre approche.

2.1.1 Apprentissage supervisé :

L'apprentissage supervisé consiste à entraîner le modèle en lui fournissant des données d'entrée (features) accompagnées de leurs sorties correspondantes (labels). À travers ce processus d'entraînement, l'ordinateur apprend à établir des relations et des chemins entre les données d'entrée et les sorties, ce qui lui permet par la suite de prédire les sorties pour de nouvelles données [15]. L'apprentissage supervisé peut être classé en régression, où la variable de résultat est continue, et classification, où la variable de résultat prend des valeurs catégorielles discrètes [16].

Dans ce travail, nous nous intéresserons particulièrement à la classification, car elle est au cœur des objectifs de diagnostic et de sous-typage du cancer.

- **Classification :**

Les algorithmes de classification sont utilisés pour regrouper des données en prédisant une étiquette catégorielle ou une variable de sortie en fonction des données d'entrée. La classification est utilisée lorsque les variables de sortie sont catégorielles, c'est-à-dire qu'il existe au moins deux classes. Les modèles de classification prennent les données en entrée du classificateur et les classent dans une classe spécifique, généralement entre deux classes [17].

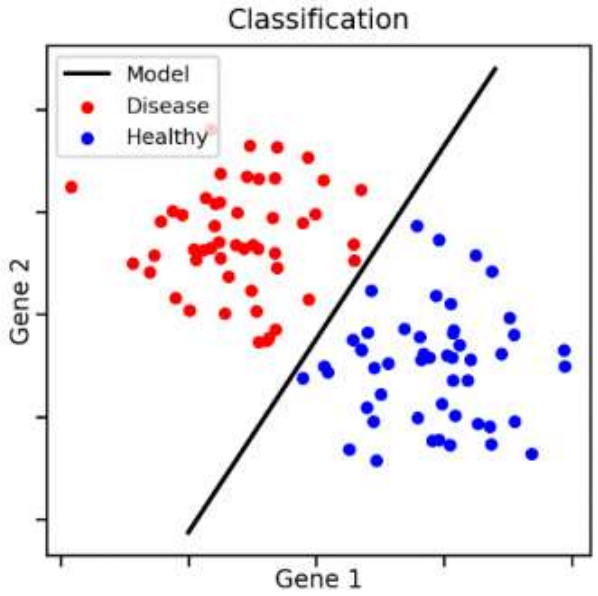


Figure 5: Illustration d'un modèle de classification binaire appliqué à des données biologiques [18]**.**

Parmi les approches de classification supervisée, adoptée dans notre étude s'est particulièrement appuyée sur l'algorithme Random Forest, choisi pour ses performances reconnues dans le traitement des données cliniques.

2.1.2 Random Forest (RF)

Random Forest est un algorithme d'apprentissage automatique supervisé puissant, utilisable pour les problèmes de régression et de classification [19]. Il est définie comme un ensemble d'arbres de décisions, ou une forêt d'arbres de décisions, combinant plusieurs modèles d'arbres en un seul algorithme d'ensemble [19].

- **Principe de Random Forest**

Le principe clé du Random Forest est de placer entre les arbres une certaine diversité en introduisant deux mécanismes : le bagging et la sélection au hasard des attributs à chaque division.

Premièrement, le bagging (bootstrap aggregating) consiste à entraîner chaque arbre de décision sur un sous-ensemble aléatoire des exemples de l'ensemble d'entraînement. En d'autres termes, chaque arbre de décision de la forêt aléatoire est entraîné sur un sous-ensemble d'exemples différents [20].

Deuxièmement, la sélection aléatoire des attributs (l'échantillonnage des attributs) consiste à ce qu'au lieu de rechercher la meilleure condition pour toutes les fonctionnalités disponibles, seul un

sous-ensemble aléatoire de fonctionnalités sera testé à chaque nœud, cet ensemble étant échantillonné de manière aléatoire à chaque nœud de l’arbre de décision [20].

Enfin, lors de la prédiction, l'ensemble des arbres vote pour la classe majoritaire (en classification) ou calcule la moyenne (en régression), consolidant ainsi la décision.

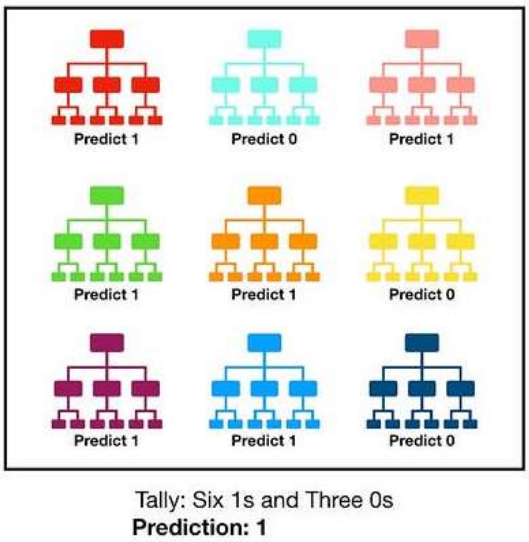


Figure 6 : Combinaison des prédictions pour produire le résultat final [21].

2.2 Apprentissage profond

L'apprentissage profond est une branche des méthodes d'apprentissage automatique qui implique l'apprentissage de plusieurs niveaux de représentation et d'abstraction, et a la capacité de traiter les données dans leur format brut, ainsi que de découvrir les représentations nécessaires à la détection ou à la classification de manière automatisée [22]. Les modèles d'apprentissage profond sont des réseaux de neurones artificiels qui contiennent plusieurs couches cachées de neurones. En général, ils ont une grande précision, mais sont plus coûteux en calcul que les autres méthodes d'apprentissage automatique [23]. L'entraînement supervisé des modèles d'apprentissage profond nécessite de vastes ensembles de données étiquetées [24].

Le modèle appelé « Deep learning » utilise des réseaux de neurones artificiels pour apprendre à partir des données. Les réseaux de neurones sont constitués de couches de nœuds interconnectés, et chaque nœud est responsable de l'apprentissage d'une caractéristique spécifique des données. Un réseau neuronal est un système d'apprentissage informatique qui utilise un réseau de fonctions pour comprendre et traduire une entrée de données d'une forme en la sortie souhaitée, généralement sous une autre forme. Les réseaux neuronaux, dans ce contexte, font référence à un ensemble de neurones qui pourraient être artificiels. Plusieurs couches de nœuds liés constituent un réseau neuronal [51].

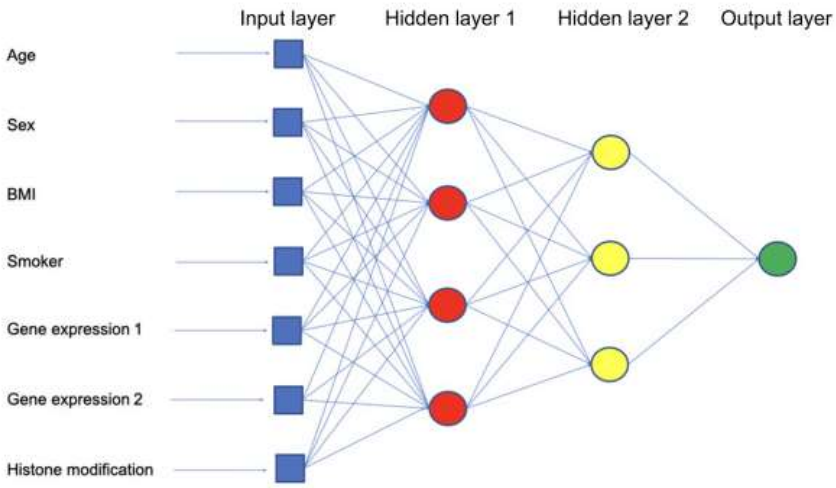


Figure 7: Réseaux de neurones artificiels [25].

À mesure que le réseau apprend, les pondérations sur les connexions entre les nœuds sont ajustées afin que le réseau puisse mieux classifier les données. Ce processus, appelé "entraînement", peut être réalisé à l'aide de diverses techniques, telles que l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [52].

2.3 Apprentissage par transfert

L'apprentissage par transfert est une stratégie d'apprentissage automatique où les connaissances acquises lors de l'entraînement d'un modèle sur une tâche générale sont transférées pour être réutilisées dans une seconde tâche spécifique [33]. L'emploi d'un modèle pré-entraîné permet de bénéficier des connaissances collectés sur d'importants volumes de données, rendant ainsi plus facile l'adaptation du modèle à de nouvelles missions. L'apprentissage par transfert permet de réutiliser le modèle pré-entraîné. Il exploite les connaissances acquises lors de la tâche précédente [34]. Au lieu de recommencer à partir de zéro, cette technique permet au modèle d'utiliser les connaissances obtenues durant l'entraînement initial. L’apprentissage par transfert peut réduire considérablement les coûts de calcul et les besoins en données ; les modèles utilisés pour mener l'optimisation sont pré-entraînés et prêts à être utilisés pour tout ensemble de données d'intérêt à un coût de calcul mineur [35].

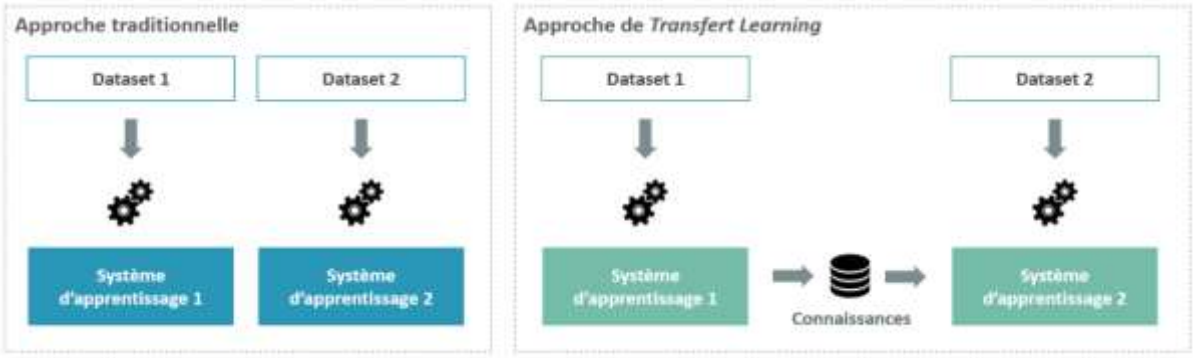


Figure 8: Approche traditionnelle vs. Approche de Transfert Learning [36].

Parmi les architectures les plus populaires utilisées dans le cadre du transfert learning, on trouve ResNet et DenseNet121. Ces deux réseaux convolutifs profonds ont montré une grande efficacité dans diverses tâches de classification d'images grâce à leurs structures innovantes et leur capacité à capturer des caractéristiques complexes dans les données d'entrée.

- **ResNet**

Les réseaux résiduels, communément appelés ResNet, représentent une architecture révolutionnaire de réseau neuronal convolutif CNN développée par Kaiming He et ses collègues de Microsoft Research. Présenté dans leur article de 2015,"Deep Residual Learning for Image Recognition", ResNet s'est attaqué à un défi majeur de l'apprentissage profond DL: le problème de la dégradation. Ce problème survient lorsque l'ajout de couches supplémentaires à un réseau très profond entraîne une erreur d'apprentissage plus élevée, contrairement à l'attente selon laquelle les modèles plus profonds devraient être plus performants. L'innovation de ResNet a permis de former avec succès des réseaux nettement plus profonds que ce qui était possible auparavant, faisant ainsi progresser de manière significative l'état de l'art dans diverses tâches de vision par ordinateur [37].

ResNet50 est couramment employé en tant que modèle pré-entraîné dans le contexte de l'apprentissage par transfert, notamment sur d'importants jeux de données tels qu'ImageNet. Sa structure permet de réutiliser efficacement les caractéristiques apprises pour des tâches spécifiques, notamment en classification d'images médicales [38].

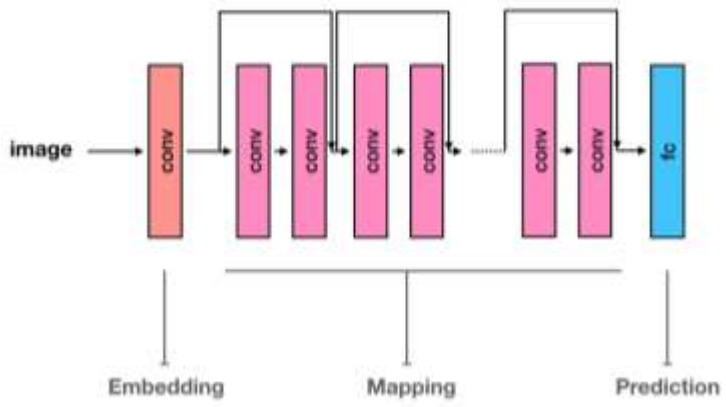


Figure 9: Vue schématique de l'architecture ResNet [39].

- **DenseNet**

DenseNet est un extracteur de caractéristiques d'images en informatique qui utilise des connexions denses dans toutes les couches convolutives. Il concatène chaque couche avec toutes les couches précédentes de la dimension canal et les utilise comme entrée pour la couche suivante. Cette connexion dense permet d'améliorer les performances du modèle grâce à la réutilisation des caractéristiques [40]. DenseNet peut être considéré comme une extension de l'architecture ResNet50 , où chaque couche reçoit des entrées supplémentaires de toutes les couches précédentes plutôt qu'une seule connexion de saut. Elle transfère ses sorties à toutes les couches convolutives suivantes pour concaténation. Ainsi, chaque couche est censée obtenir une (connaissance collective) de toutes les couches convolutives précédentes. DenseNet est utilisé dans quelques études et a montré de bonnes performances par rapport aux autres réseaux pré-entraînés [41]. Dans DenseNet, chaque couche de convolution reçoit en entrée la sortie (c'est-à-dire les cartes de caractéristiques) de toutes les couches précédentes et transmet sa propre sortie (c'est-à-dire les cartes de caractéristiques) à toutes les couches suivantes. Ainsi, chaque couche acquiert la connaissance collective de toutes les couches précédentes. Le modèle CNN obtenu devient plus fin et plus compact en raison de la diminution du nombre de cartes de caractéristiques.

Le modèle DensNet existe en plusieurs versions, telles que DenseNet-121, DeneNet-169 et DenseNet-201 [42]. L'architecture DenseNet est composée de plusieurs blocs denses, séparés par des couches de transition. Voici les composants clés :

  - **Bloc Dense**
- **Couches Convolutives :** Chaque bloc dense contient plusieurs couches convolutives ;

- **Connexions Denses :** Chaque couche reçoit les caractéristiques de toutes les couches précédentes du bloc. Cela signifie que la i-ème couche reçoit les caractéristiques des couches 0, 1, …, i-1 ;
- **Concaténation :** Les caractéristiques sont concaténées ensemble, plutôt que sommées, préservant ainsi les informations de toutes les couches précédentes ;
- **Croissance du Canal :** Un paramètre clé est le "taux de croissance" (k), qui contrôle le nombre de canaux (caractéristiques) ajoutés par chaque couche convolutive ;

▪ **Couche de Transition**

Entre les blocs denses, il y a des couches de transition qui contrôlent la taille et la profondeur des caractéristiques. Elles contiennent généralement une convolution suivie d'une couche de pooling pour réduire la dimensionnalité.

▪ **Couche de Classification**

Après les blocs denses et les couches de transition, il y a généralement une couche de pooling global suivie d'une couche dense pour la classification [43].

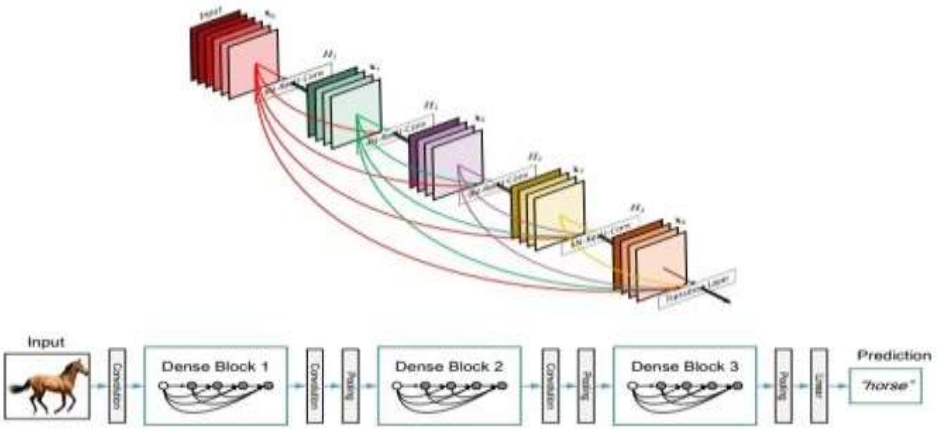


Figure 10: Différents blocs et couches dans DenseNet [44].

3 Applications de l'IA dans le domaine médical et de la santé

L'intelligence artificielle joue un rôle de plus en plus central en médecine, notamment dans la recherche fondamentale et clinique, la pratique hospitalière, les examens médicaux et les soins aux patients. Son intégration améliore la précision des diagnostics, affine les pronostics et de développer des approches médicales plus personnalisées et ciblées. De plus, Elle contribue aux avancées technologiques dans l'imagerie médicale, l'analyse des données biologiques et les outils d'assistance robotisée pour les interventions chirurgicales [45]. Un des premiers domaines

d’application de l’IA en médecine est la structuration et l’exploitation des données de santé. Grâce à des algorithmes avancés, elle est capable de collecter, d’analyser et d’organiser de vastes quantités de données cliniques et génomiques, facilitant ainsi leur interprétation par les professionnels de santé [46]. L'évaluation précise de données moléculaires complexes afin de faciliter le diagnostic et la prise en charge rapides des maladies génomiques nécessitera des méthodes d'intelligence artificielle [47].

Un autre champ d’application majeur est l’aide au diagnostic. Les modèles d’apprentissage automatique permettent d’analyser plus rapidement et plus efficacement des échantillons biologiques, de détecter des cellules cancéreuses et d’identifier des anomalies sur des électrocardiogrammes ou des images médicales. L’IA améliore considérablement la rapidité et la précision des interprétations radiologiques, permettant ainsi des diagnostics et une planification thérapeutique plus précis dans la prise en charge du cancer [48]. Exemple, les outils prédictifs basés sur l'IA utilisant des données ECG (électrocardiogrammes) peu coûteuses, accessibles et potentiellement applicables à distance peuvent devenir utiles pour le dépistage et le diagnostic en cardio-oncologie [49].

L’IA ne se limite pas seulement à la détection des maladies, elle intervient aussi dans la prise en charge thérapeutique. En croisant des données issues de cas cliniques passés et en analysant les caractéristiques propres à chaque patient, elle peut aider à personnaliser les traitements. Les modèles d'IA peuvent prédire comment un patient atteint de cancer pourrait réagir à un schéma de chimiothérapie particulier en fonction de son profil génétique et des caractéristiques génétiques de sa tumeur [50]. Un modèle radiopathomique intégré basé sur l'intelligence artificielle pour prédire la réponse pathologique complète chez les patients atteints d'un cancer du rectum localement avancé, à l'aide d'IRM préthérapeutique et de lames de biopsie colorées à l'hématoxyline-éosine (H&E) [51].

Enfin, l’IA peut être utilisée pour modéliser et simuler des phénomènes biologiques complexes afin d’améliorer la compréhension des maladies et d’optimiser les stratégies thérapeutiques. Des modèles informatiques permettent de simuler les processus mécaniques, chimiques et physiologiques d’un organe, d’un tissu ou d’une cellule, facilitant ainsi l’identification des mécanismes sous-jacents à certaines pathologies et le développement de nouvelles approches thérapeutiques [45]. L'association de la biologie computationnelle intégrative et de l'IA offre le potentiel d'améliorer la compréhension et le traitement des maladies en identifiant des biomarqueurs et en construisant des modèles explicatifs caractérisant chaque patient [52].

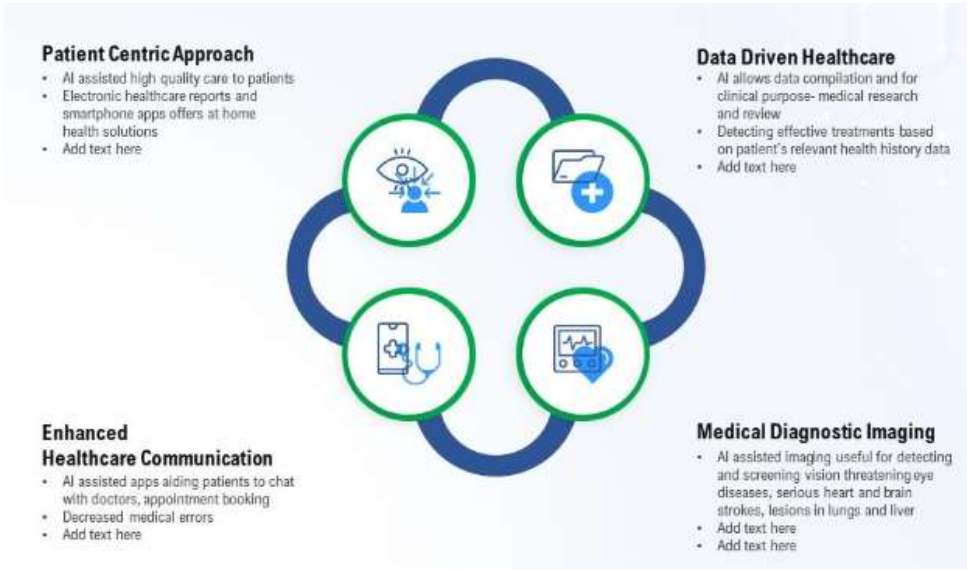


Figure 11: Tendances actuelles de l'IA dans le secteur de la santé [53].

L'IA s'impose comme un outil essentiel dans l'amélioration des résultats, traitements ou gestion de pathologies dans plusieurs cas :

- **Analyse d'imagerie médicale et aide au diagnostic médical**

Les diagnostics sont plus rapides, plus fiables comme en pathologie, grâce à l'analyse précise des images de biopsies, L'étude de l'imagerie médicale via l'IA est aussi devenue un soutien précieux pour les professionnels de la santé. La précision diagnostique des pathologistes est passée de 97,1 % à 100 % grâce au support de l'IA. De plus, l'utilisation de l'IA a considérablement amélioré l'efficacité des pathologistes, réduisant leur temps d'examen d'une moyenne de 16,5 % pour les 3 pathologistes et a conduit à une réduction de 33 % de l'utilisation de l'immunohistochimie [54]. Pour détecter les tumeurs, fractures ou infections à partir des images médicales on peut utiliser des réseaux neuronaux profonds (un processus d'apprentissage automatique). Les modèles d'IA détectent désormais de manière fiable les pathologies rachidiennes clés, atteignant des performances de niveau expert dans des tâches telles que l'identification des fractures, des sténoses, des infections et des tumeurs [55]. L'utilisation d'outils d'IA permet d'examiner des volumes massifs de données pour détecter des anomalies invisibles à l'œil humain, comme une micro lésion. L'IA peut détecter des micro-caractéristiques au-delà de l'œil humain et apporter des solutions dans les cas de diagnostic critiques [56]. Les performances diagnostiques variaient selon les classes de lésions carieuses (lésions non cavitaires, translucidité/microcavité grisâtre, cavitation, dent détruite), avec des précisions allant de 88,9 % à 98,1 %, des sensibilités allant de 68,8 % à 98,5 % et des spécificités allant de 86,1 % à 99,4 % [57].

- **Chatbots médicaux et télémédecine**

Des chatbots utilisent des algorithmes d’analyse du langage naturel pour interpréter les réponses des patients dans les échanges écrits Ils sont orientés vers des soins appropriés en fonction de la gravité de leurs symptômes. Une application d’intelligence artificielle (IA) conversationnelle basée sur la voix peut aider les patients atteints de diabète de type 2 à titrer l’insuline basale à domicile pour obtenir un contrôle glycémique rapide [58].

Des algorithmes permettent de réaliser des consultations à distance (téléconsultations) ou de suivre l’évolution des patients de manière continue. Les technologies portables intègrent des capteurs et peuvent mesurer l'activité physique, la fréquence et le rythme cardiaques, ainsi que la glycémie et les électrolytes. Pour les personnes à risque, les dispositifs portables ou autres dispositifs peuvent être utiles pour la détection précoce de la fibrillation auriculaire ou des états infra cliniques de maladies cardiovasculaires, la prise en charge des maladies cardiovasculaires telles que l'hypertension et l'insuffisance cardiaque, et la modification du mode de vie [59].

- **IA au service de la prescription et de l’optimisation des traitements**

En analysant un dossier médical électronique, l’intelligence artificielle peut suggérer des traitements optimisés en tenant compte des antécédents du patient, de ses allergies, ou des interactions médicamenteuses possibles. Les systèmes de prescription intelligents (SPI) représentent une avancée prometteuse dans le domaine de la santé, offrant le potentiel d'optimiser la sélection, la posologie et le suivi des médicaments, en fonction des besoins de chaque patient [60]. Le dosage des traitements peut être ajusté via IA en fonction des besoins spécifiques du patient. Le développement d’Un conseiller IA en dosage et en combinaison de médicaments (AIDA) pour la gestion de la glycémie, en utilisant les dossiers médicaux électroniques de 107 854 patients atteints de DT2 inscrits au registre du diabète SingHealth [61]. La dose d'insuline guidée par un système automatisé d'aide à la décision basé sur l'intelligence artificielle (AI-DSS) sont aussi efficaces et sûrs que ceux guidés par les médecins pour contrôler la glycémie [62].

Un système électronique en boucle fermée pour pomper automatiquement l'insuline nécessaire dans le corps du patient en synchronisation avec les lectures du capteur de glucose [63].

- **Gestion des données de santé, parcours des patients et dossiers médicaux**

L’IA structure les données de santé pour les rendre exploitables. Utilisent l’IA pour organiser les Big Data de santé, facilitant la communication entre praticiens, établissements de santé et patients. L'intelligence artificielle dans le domaine de la santé décrit des techniques de calcul algorithmiques qui gèrent et analysent de vastes ensembles de données afin de réaliser des inférences et des prédictions [64].

4 Modèles IA utilisés pour la prédiction du CPNPC

4.1 Définition et objectifs des modèles multimodaux

Les méthodes multimodales nécessitent la combinaison et l'examen de différentes sortes de données, telles que les images médicales, les biosignaux, les dossiers médicaux et d'autres sources pertinentes, dans le but d'acquérir une meilleure compréhension de la condition des patients. Chaque modalité expose un aspect spécifique de la physiologie et de la pathologie, et la combinaison ainsi que l'interprétation efficace des données multimodales offrent simultanément des opportunités et des défis singuliers [65]. Différentes formes de données sont mises en application dans le contexte clinique. Cela comprend les données d'imagerie (comme les rayons X), les données textuelles (telles que les comptes rendus de radiologie), les données chronologiques (par exemple, les signes vitaux) et les données transversales ou de panel sous forme de tableaux (telles que les métadonnées). Conscients de l'importance cruciale du choix des données dans l'apprentissage automatique multimodal, nous proposons une synthèse des principaux types de données employés dans les recherches de modélisation [66].

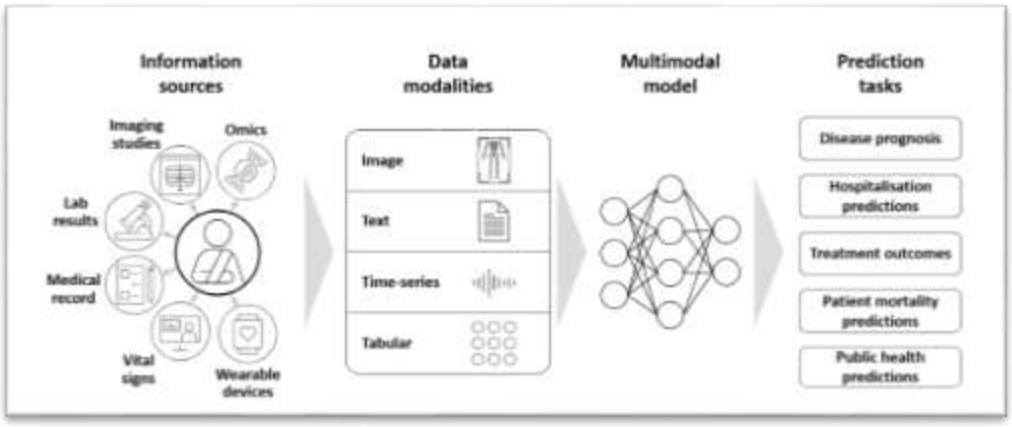


Figure 12 : Modalités des données et tâches de prédiction [66].

4.2 Techniques de fusion des données multimodales

La Fusion de Données Multimodales fait référence au processus d'intégration d'informations provenant de diverses sources ou modalités afin d'obtenir une compréhension plus approfondie ou d'améliorer la performance d'un système. Les modalités peuvent englober divers types de données tels que le texte, les images, l'audio et les lectures de capteurs. Le processus de fusion vise à exploiter les points forts de chaque modalité tout en compensant leurs faiblesses individuelles [67]. L'intégration en MML (L'apprentissage multimodal) peut s'effectuer à divers niveaux, y compris au stade précoce (niveau des attributs), intermédiaire (niveau du modèle) ou avancé

(niveau de la prise de décision). Chaque phase de fusion comporte ses propres atouts et points faibles, et la sélection de la phase appropriée est déterminée par les spécificités des données et de l'opération à réaliser [68].

4.2.1 Fusion précoce

La fusion anticipée fait référence à la combinaison des caractéristiques dérivées de diverses modalités de données en un seul vecteur avant le processus d'apprentissage du modèle. Les vecteurs caractéristiques issus de diverses modalités sont fusionnés en un seul vecteur, qui sert d'entrée au modèle d'apprentissage automatique. Cette méthode peut être mise en œuvre lorsque les modalités contiennent des informations additionnelles et peuvent être aisément synchronisées, comme dans le cas de la fusion de caractéristiques visuelles et sonores pour une application d'analyse vidéo. L'enjeu majeur de la fusion précoce est d'assurer que les caractéristiques dérivées des diverses modalités sont compatibles et apportent des renseignements complémentaires [68].

4.2.2 Fusion intermédiaire

La fusion intermédiaire implique l'entraînement de modèles séparés pour chaque type de donnée, suivi de la fusion des résultats obtenus à des fins d'inférence ou de prédiction. Cette stratégie est appropriée lorsque les modalités de données sont autonomes et ne peuvent pas être aisément fusionnées au niveau des attributs à travers la moyenne, la moyenne pondérée ou toute autre technique. L'enjeu majeur de la fusion intermédiaire consiste à sélectionner une technique adéquate pour agréger les résultats de divers modèles [68].

4.2.3 Fusion tardive

Dans le cadre de la fusion tardive, les résultats propres à chaque modalité spécifique sont utilisés pour rendre une décision de manière indépendante. Toutes les décisions sont ensuite fusionnées pour parvenir à une décision définitive. Cette méthode est appropriée quand les modalités offrent des renseignements supplémentaires, sans pour autant être forcément indépendantes entre elles. Le principal enjeu de la fusion tardive est de sélectionner une méthode adéquate pour l'assemblage des prédictions individuelles. On peut y recourir par le biais du vote majoritaire, du vote pondéré ou d'autres modèles d'apprentissage automatique [68].

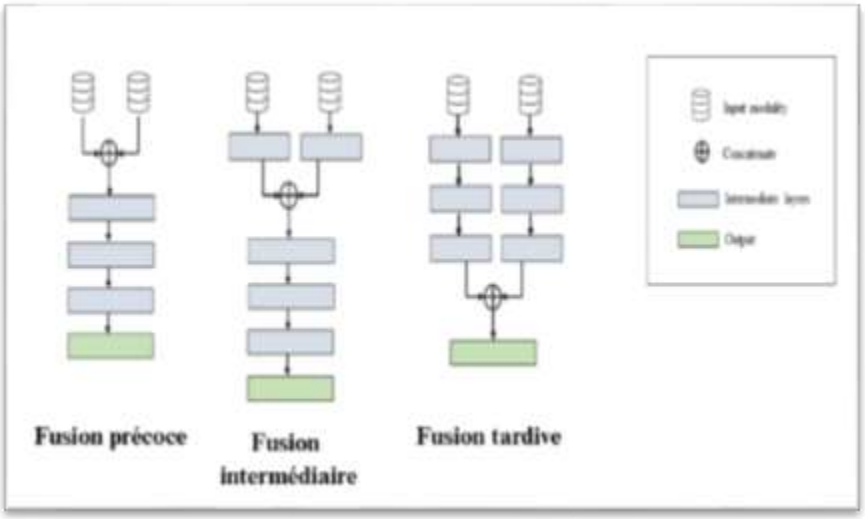


Figure 13 : architectures de fusion de données [69]

4.3 Applications Multimodal de Données Fusion

➢ **Soins de santé :**

- Utilisations : Intègre l'imagerie médicale, l'électronique de santé, les enregistrements et les capteurs de données pour améliorer le diagnostic, le traitement, la planification et la surveillance des patients.
- Exemples : Association d'IRM avec l'historique du patient et ses signes vitaux pour améliorer la détection et la gestion de la maladie.

➢ **Autonome Systèmes :**

- Utilisations : Exploitation des données issues de caméras, LiDAR, radar et GPS pour activer les véhicules autonomes, leur permettre de naviguer, détecter les obstacles et prendre des décisions.
- Exemples : Les voitures autonomes utilisent des données visuelles, spatiales et capteurs pour prendre des décisions en temps réel lors de la fabrication.

➢ **Sociale Médias Analyse :**

- Utilisations : Analyse de texte, d'images et de vidéos pour comprendre le comportement, les émotions et les tendances des utilisateurs.
- Exemples : Combinaison de texte et d'images utilisées pour évaluer le sentiment public sur les plateformes de médias sociaux.

➢ **Sécurité et Surveillance**

- Utilisations : Amélioration de la surveillance et des mesures de sécurité grâce à l'utilisation de moissonneuses-batteuses vidéo, d'enregistrements audio et de capteurs de données.

- Exemples : Identification et réponse aux menaces de sécurité à travers des alertes audio lors de l'intégration des soins du visage.

➢ **Environnement Surveillance**

- Utilisations : Combine des capteurs de données, des images satellites et des enregistrements environnementaux pour surveiller et gérer les conditions environnementales.
- Exemples : Association des données de qualité de l'air provenant de capteurs avec les images satellite pour une résolution des niveaux et leurs origines [70].

Matériel et Méthodes

Matériel et Méthodes

Dans le cadre de ce travail de recherche, nous avons développé une approche fondée sur l'intelligence artificielle afin d'assister les médecins dans le diagnostic précoce et le sous-typage du cancer pulmonaire. Notre démarche s'articule autour de trois axes principaux.

La première phase consiste en la mise en place d'un modèle multimodal destiné au diagnostic précoce du cancer pulmonaire, combinant deux types de données : les données cliniques et les images thoraciques (CT Scan). Ce modèle intègre deux approches complémentaires : un classifieur Random Forest entraîné sur les données cliniques, et un modèle de transfert d'apprentissage basé sur ResNet50, appliqué aux images médicales.

Le deuxième axe concerne plus spécifiquement le diagnostic du Cancer Pulmonaire Non à Petite Cellule (CPNPC). Dans ce contexte, nous avons conçu un modèle ensembliste combinant les prédictions de deux architectures convolutionnelles performantes : ResNet50 et DenseNet121. Cette approche vise à améliorer la fiabilité des résultats grâce à la fusion des prédictions issues des deux modèles.

Enfin, la troisième étape a été dédiée à la conception de DiagnoLung, une plateforme interactive destinée à faciliter l'utilisation clinique des modèles développés. Cette interface permet aux praticiens de charger facilement les données d'un patient (données cliniques et scans) et d'obtenir une prédiction rapide et interprétable concernant la probabilité d'un cancer pulmonaire ainsi que son éventuel sous-type.

1 Modèle multimodal de diagnostic précoce

1.1 Modélisation clinique basée sur Random Forest

1.1.1 Prétraitement des données

Les données cliniques utilisées dans le cadre de ce projet proviennent d’un dataset public disponible sur la plateforme Kaggle, sous le nom : Lung Cancer Dataset [71]. Ce jeu de données regroupe des informations cliniques liées à des patients atteints ou non de cancer pulmonaire. Le fichier principal, au format CSV, contient les données de 309 patients, chacun décrit par 15 variables cliniques et une variable cible (LUNG_CANCER) indiquant la présence ou l’absence de cancer.

L’étape de prétraitement des données consiste à importer les bibliothèques, charger et nettoyer les données cliniques, encoder et normaliser les variables, équilibrer les classes, puis sauvegarder le dataset final.

- **Description des données brutes**

Tableau 1: Description de dataset des données cliniques.

SEXE	M (homme), F (femme)
AGE	Âge du patient
TABAGISME	OUI=2, NON=1.
DOIGTS JAUNES	OUI=2, NON=1.
ANXIETE	OUI=2, NON=1.
PRESSION SOCIALE	OUI=2, NON=1.
MALADIE CHRONIQUE	OUI=2, NON=1.
FATIGUE	OUI=2, NON=1.
ALLERGIE	OUI=2, NON=1.
RESPIRATION SIFFLANTE	OUI=2, NON=1.
ALCOOL	OUI=2, NON=1.
TOUX	OUI=2, NON=1.
ESSOUFFLEMENT	OUI=2, NON=1.
DIFFICULTE A AVALER	OUI=2, NON=1.
DOULEUR THORACIQUE	OUI=2, NON=1.
CANCER DU POUMON	OUI, NON.

- **Importation des bibliothèques nécessaires :**

  Les bibliothèques essentielles à la manipulation des données, à la visualisation des résultats et à la construction du modèle ont été importées.

```
[1]: #1. Importation des bibliothèques nécessaires

[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler, LabelEncoder
```

Figure 14: Script python pour importer les bibliothèques nécessaires.

- **Chargement du fichier survey lung cancer.csv :**

  Le fichier contenant les données cliniques a été chargé et stocké dans un DataFrame à l'aide de la bibliothèque Pandas.

```
[3]: #2. Chargement des données

[4]: df = pd.read_csv("survey lung cancer.csv")
     df
```

[4]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY
0	M	69	1	2	2
1	M	74	2	1	1
2	F	59	1	1	1
3	M	63	2	2	2
4	F	63	1	2	1
...	...	...	...	...	...
304	F	56	1	1	1
305	M	70	2	1	1

Figure 15 : Script python pour le chargement du fichier.

- **Vérification et gestion des valeurs manquantes :**

  Une vérification a été faite pour détecter les valeurs manquantes.

```
[5]: #3. Vérification et gestion des valeurs manquantes

[6]: missing_values = df.isnull().sum()
     print("Valeurs manquantes par colonne :\n", missing_values)
```

```
Valeurs manquantes par colonne :
 GENDER                    0
AGE                        0
SMOKING                    0
YELLOW_FINGERS             0
ANXIETY                    0
PEER_PRESSURE              0
CHRONIC DISEASE            0
FATIGUE                    0
ALLERGY                    0
WHEEZING                   0
ALCOHOL CONSUMING          0
COUGHING                   0
SHORTNESS OF BREATH        0
SWALLOWING DIFFICULTY      0
CHEST PAIN                 0
LUNG_CANCER                0
dtype: int64
```

Figure 16: Script python pour la vérification et gestion des valeurs manquantes.

- **Vérification et gestion des doublons :**

Les doublons ont été identifiés et supprimés afin d'éviter toute source de biais dans les analyses.

```
[7]: #4. Vérification et gestion des doublons

[8]: #Vérifier les doublons
duplicates = df.duplicated().sum()
print(f"Nombre de doublons : {duplicates}")

Nombre de doublons : 33

[9]: # Supprimer les doublons
df = df.drop_duplicates()
df
```

[9]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS
0	M	69	1	2
1	M	74	2	1
2	F	59	1	1
3	M	63	2	2
4	F	63	1	2
...	...	...	...	...

Figure 17: Scripts python pour la vérification et gestion des doublons.

- **Encodage des variables catégoriques :**

Les colonnes contenant des valeurs textuelles (GENDER, LUNG_CANCER) ont été encodées en valeurs numériques pour permettre leur utilisation dans les algorithmes de classification.

```
[5]: # Vérifier les colonnes catégoriques
categorical_cols = df.select_dtypes(include=["object"]).columns
print("Colonnes catégoriques :", categorical_cols)

Colonnes catégoriques : Index(['GENDER', 'LUNG_CANCER'], dtype='object')

[7]: # Encodage des variables catégoriques avec LabelEncoder
encoder = LabelEncoder()
for col in categorical_cols:
    df[col] = encoder.fit_transform(df[col])
print(df[categorical_cols])

     GENDER  LUNG_CANCER
0         1            1
1         1            1
2         0            0
3         1            0
4         0            0
..      ...          ...
279       0            1
280       0            0
281       1            0
282       1            0
283       1            1

[276 rows x 2 columns]
```

Figure 18: Encodage des variables catégoriques avec LabelEncoder.

- **Équilibrage des classes :**

Un suréchantillonnage de la classe minoritaire (patients sains) a été effectué afin d'équilibrer les données entre les deux classes.

```
[13]: #6. Compter le nombre de personnes malades et saines

[14]: cancer_counts = df['LUNG_CANCER'].value_counts()

      # Afficher le résultat
      print("Nombre de personnes malades et saines :")
      print(cancer_counts)

Nombre de personnes malades et saines :
LUNG_CANCER
1    238
0     38
Name: count, dtype: int64

[15]: #7. équilibrer les données 50% de personnes maladeset 50% de personnes saines

[16]: # Séparer les malades et les sains
      df_malades = df[df['LUNG_CANCER'] == 1]  # 238 patients malades
      df_sains = df[df['LUNG_CANCER'] == 0]    # 38 patients sains

      # Augmenter les patients sains (oversampling) jusqu'à en avoir 238
      df_sains_augmente = df_sains.sample(n=len(df_malades), replace=True, random_state=42)

      # Combiner pour former un dataset équilibré
      df_equilibre = pd.concat([df_malades, df_sains_augmente], ignore_index=True)

      # Mélanger les lignes pour ne pas avoir d'ordre
      df_equilibre = df_equilibre.sample(frac=1, random_state=42).reset_index(drop=True)

      # Vérifier l'équilibre
      print("✅ Nombre de patients après oversampling :")
      print(df_equilibre['LUNG_CANCER'].value_counts())

✅ Nombre de patients après oversampling :
LUNG_CANCER
0    238
1    238
Name: count, dtype: int64
```

Figure 19: Scripts python pour l'équilibrage des classes (Oversampling).

- **Transformation des valeurs binaires (2 → 1, 1 → 0) :**

Les valeurs binaires ont été harmonisées en remplaçant les 2 par des 1 et les 1 par des 0, afin d'assurer une cohérence dans la représentation des réponses.

```
[10]: # Liste des colonnes à modifier (toutes sauf GENDER, AGE et LUNG_CANCER)
colonnes_a_modifier = [
    'SMOKING', 'YELLOW_FINGERS', 'ANXIETY', 'PEER_PRESSURE', 'CHRONIC DISEASE',
    'FATIGUE ', 'ALLERGY ', 'WHEEZING', 'ALCOHOL CONSUMING', 'COUGHING',
    'SHORTNESS OF BREATH', 'SWALLOWING DIFFICULTY', 'CHEST PAIN'
]

# Remplacer 2 par 1 et 1 par 0 dans les colonnes spécifiées
df_equilibre[colonnes_a_modifier] = df_equilibre[colonnes_a_modifier].replace({2: 1, 1: 0})
df_equilibre
```

[10]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE	FATIGUE	A
0	0	59	1	0	0	0	1	1	
1	0	64	1	1	0	1	0	1	
2	0	74	0	1	1	1	1	1	
3	0	55	1	0	0	1	1	1	
4	1	60	0	1	1	1	0	1	
...	...	...	...	...	...	...	...	...	
471	1	51	1	0	0	0	0	1	
472	0	63	0	0	0	0	1	1	
473	1	69	0	0	0	1	0	1	
474	0	55	1	0	1	0	0	1	
475	1	57	1	0	1	0	1	1	

476 rows × 16 columns

Figure 20: Recodage des variables binaires dans le dataset équilibré.

- **Normalisation de l'âge avec MinMaxScaler :**

La variable AGE a été normalisée à l'aide de la méthode MinMaxScaler, afin de restreindre les valeurs dans une plage comprise entre 0 et 1.

```
[97]: #9. Normalisation des données

[98]: from sklearn.preprocessing import MinMaxScaler

# Étape 1: Sauvegarder le DataFrame final après nettoyage et encodage
df_final = df_equilibre.copy()

# Étape 2: Identifier les colonnes numériques continues à normaliser
columns_to_normalize = ["AGE"]

# Étape 3: Appliquer la normalisation
scaler = MinMaxScaler()
df_final[columns_to_normalize] = scaler.fit_transform(df_final[columns_to_normalize])

# Étape 4: Vérifier les résultats
df_final
```

[98]:

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	CHRONIC DISEASE
0	0	0.575758	1	0	0	0	1
1	0	0.651515	1	1	0	1	0
2	0	0.803030	0	1	1	1	1
3	0	0.515152	1	0	0	1	1
4	1	0.590909	0	1	1	1	0
...	...	...	...	...	...	...	...

Figure 21: Normalisation des variables numériques avec MinMaxScaler.

- **Sauvegarde du DataFrame final :**

  Le DataFrame nettoyé et préparé a été sauvegardé dans un fichier CSV en vue d'une utilisation ultérieure.

```
[ ]: df_final.to_csv('df_final_données_cliniques.csv', index=False)
```

Figure 22: Sauvegarde du dataset final prétraité au format CSV.

1.1.2 Random Forest – Données cliniques

Un algorithme d'apprentissage supervisé de classification, le Random Forest Classifier, a été utilisé pour prédire la présence ou non d'un cancer pulmonaire à partir des données cliniques. Les performances du modèle ont été évaluées à l'aide des métriques classiques de classification.

- **Importation des bibliothèques pour la modélisation :**

  Les bibliothèques nécessaires à la création, l'évaluation et la sauvegarde du modèle ont été importées (scikit-learn et joblib).

```
[1]: #1. Importation des bibliothèques pour la modélisation

[2]: from sklearn.model_selection import train_test_split
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
     import joblib
```

Figure 23: Importation des bibliothèques pour la modélisation.

- **Séparation des variables explicatives (features) et de la variable cible :**

Les variables explicatives ont été extraites en supprimant la colonne LUNG_CANCER du DataFrame final, tandis que la variable cible a été isolée dans un vecteur distinct.

```
[3]: #2. Séparer les features et la cible

[4]: X = df_final.drop(columns=["LUNG_CANCER"])
     y = df_final["LUNG_CANCER"]
```

Figure 24: Séparation des variables explicatives et de la variable cible.

- **Division des données en ensembles d'entraînement, de validation et de test :**

Une première séparation a été effectuée pour réserver 20 % des données à l'évaluation finale (ensemble de test). Ensuite, les 80 % restants ont été divisés en deux sous-ensembles :

- 70 % pour l'entraînement
- 10 % pour la validation (représentant environ 12,5 % du total initial).

```
[5]: #3. Division des données en ensembles d'entrainement, de validation et de test

[6]: # 1. Séparer temporairement 80% pour train/val et 20% pour test
     X_temp, X_test, y_temp, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

     # 2. Séparer X_temp en 70% train et 10% val (ce qui fait 87.5% * 0.125 = 10%)
     X_train, X_val, y_train, y_val = train_test_split(X_temp, y_temp, test_size=0.125, random_state=42, stratify=y_temp)

     print(f"Train: {X_train.shape}, Validation: {X_val.shape}, Test: {X_test.shape}")
```

Figure 25: Division des données en ensembles d'entraînement, de validation et de test.

- **Entraînement du modèle Random Forest :**

Un modèle de type Random Forest a été initialisé avec 100 arbres de décision et une profondeur maximale de 5. Ce modèle a ensuite été entraîné à l'aide de l'ensemble d'apprentissage.

- **Évaluation sur l'ensemble de validation et test :**

Les prédictions ont été réalisées sur l'ensemble de validation, et la performance du modèle a été mesurée à l'aide de l'accuracy, du rapport de classification (précision, rappel, F1-score) ainsi que de la matrice de confusion. Une fois le modèle validé, il a été évalué sur l'ensemble de test réservé, afin d'estimer sa performance finale sur des données totalement inédites.

- **Sauvegarde du modèle entraîné :**

Le modèle final entraîné a été sauvegardé au format .pkl à l'aide de la bibliothèque joblib, afin de pouvoir être réutilisé sans avoir à le réentraîner.

```
[13]: #7. Sauvegarde du modèle entraîné

[14]: joblib.dump(rf_model, 'random_forest_clinical_model.pkl')
      print("\n✅ Modèle Random Forest sauvegardé sous 'random_forest_clinical_model.pkl'")
```

Figure 26: Sauvegarde du modèle entraîné.

1.2 Modélisation d'images CT scan basée sur ResNet50

1.2.1 Présentation et préparation du jeu de données binaire

Les images utilisées dans le cadre de ce projet proviennent d'un jeu de données public disponible sur la plateforme Kaggle, intitulé : Chest CT-Scan Images [72].

Ce jeu de données contient des images de tomodensitométrie (CT-Scan) du thorax au format .jpg et présentent des dimensions variables, classées selon déférent types de tissus pulmonaires.

L'ensemble des données est structuré en trois dossiers principaux : train, valid et test, correspondant respectivement aux phases d'entraînement, de validation et de test. Chaque dossier contient des sous-dossiers représentant différentes classes diagnostiques.

- **Répartition des données :**

Tableau 2: Description de la répartition des images dans le datatset.

	train	valid	test
adenocarcinoma	195 images	23 images	120 images
large.cell.carcinoma	115 images	21 images	51 images
squamous.cell.carcinoma	155 images	15 images	90 images
normal	148 images	13 images	54 images

Dans le cadre de ce projet, une classification binaire des images a été mise en place afin d'opposer les cas pathologiques (cancer) aux cas sains (normal). Pour cela, les images représentant les différentes formes de cancer pulmonaire ont été regroupées en une seule et unique catégorie intitulée « cancer ». Plus précisément, cette catégorie « cancer » a été constituée par la combinaison des trois sous-types tumoraux présents dans le jeu de données initial (Adénocarcinome pulmonaire, Carcinome à grandes cellules, Carcinome épidermoïde).

La classe « normal », quant à elle, regroupe les images de tissu pulmonaire sain, telles que fournies initialement. Afin d'assurer un équilibrage des classes pour chaque phase du projet

(entraînement, validation, test), un nombre équivalent d'images a été sélectionné pour les deux catégories. Le nombre d'images de la classe « cancer » a été déterminé en fonction du nombre d'images disponibles pour la classe « normal ».

Tableau 3: Répartition des images dans le jeu de données binaire (cancer vs normal).

	Classe cancer (combinaison des 3 sous-types)	Classe normal
train	148 images	148 images
valid	13 images	13 images
test	54 images	54 images

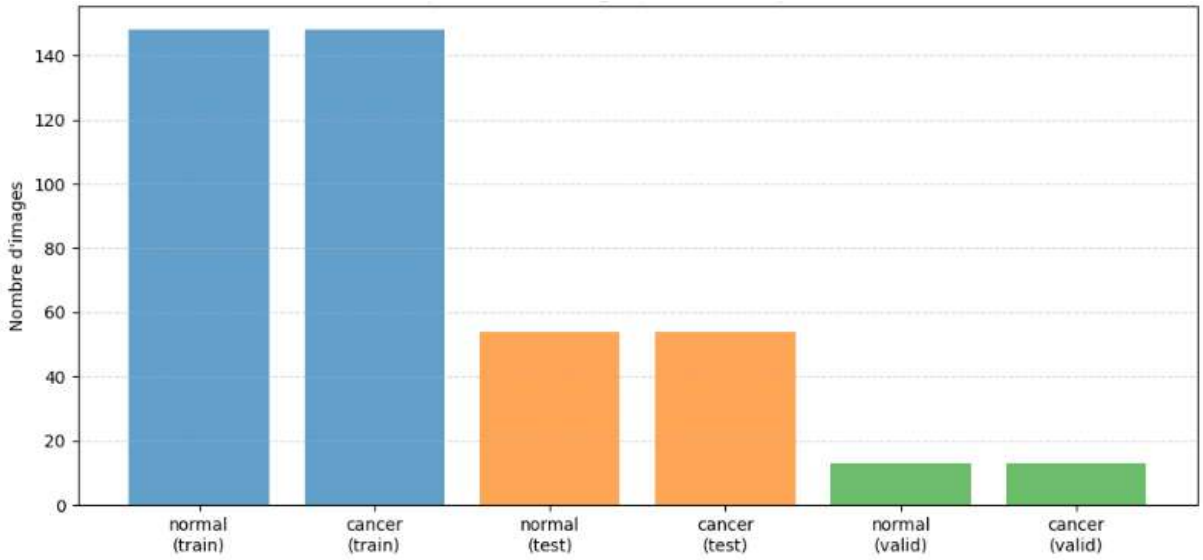


Figure 27: Répartition du nombre d'images par classe (normal vs cancer) et par phase (train, test, validation).

1.2.2 Prétraitement des images

Avant l'entraînement du modèle de classification, un prétraitement des images CT-Scan a été effectué afin d'assurer une cohérence des données en entrée du réseau de neurones et d'améliorer la robustesse du modèle.

Dans ce projet, les bibliothèques TensorFlow/Keras ont été utilisées pour charger, transformer et normaliser les images via la classe ImageDataGenerator.

- **Chargement des bibliothèques nécessaires :**

La première étape consiste à importer les bibliothèques indispensables pour le traitement des images, la construction du modèle et l'évaluation des performances.

```
[3]: import os
     import numpy as np
     import matplotlib.pyplot as plt
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     from tensorflow.keras.applications import ResNet50
     from tensorflow.keras.models import Model
     from tensorflow.keras.layers import Dense, Flatten, Dropout
     from tensorflow.keras.optimizers import Adam
     from sklearn.metrics import classification_report, confusion_matrix
```

Figure 28: Importation des bibliothèques pour la classification d'images avec ResNet50.

- **Définition des répertoires de données :**

Les données ont été organisées en trois sous-dossiers : train, test et valid, chacun contenant les images classées dans des dossiers selon leur étiquette (cancer ou normal).

```
[5]: base_dir = r"C:\Users\pc\Desktop\data_finale"
     train_dir = os.path.join(base_dir, "train")
     test_dir = os.path.join(base_dir, "test")
     valid_dir = os.path.join(base_dir, "valid")
```

Figure 29: Configuration des chemins d'accès aux ensembles d'entraînement, de validation et de teste.

- **Normalisation et augmentation des images :**

La classe ImageDataGenerator permet de normaliser les images (valeurs de pixels ramenées à l'intervalle [0,1]) et d'appliquer des transformations d'augmentation uniquement sur les données d'entraînement. Ceci permet de générer artificiellement plus d'exemples à partir du jeu d'entraînement, réduisant ainsi le risque de surapprentissage.

- rescale=1./255 : normalise les images ;
- rotation_range=15 : applique des rotations aléatoires jusqu'à 15° ;
- zoom_range=0.2 : effectue un zoom aléatoire jusqu'à 20 % ;
- horizontal_flip=True : effectue des retournements horizontaux aléatoires.

```
[7]: train_datagen = ImageDataGenerator(
         rescale=1./255,
         rotation_range=15,
         zoom_range=0.2,
         horizontal_flip=True
     )

     test_datagen = ImageDataGenerator(rescale=1./255)
```

Figure 30: Configuration des générateurs d'images pour l'augmentation des données et la normalisation.

- **Génération des flux d'images :**

Les générateurs permettent de charger les images directement depuis leur répertoire en les redimensionnant à une taille uniforme (224×224 pixels) compatible avec l'entrée de ResNet50, et en les regroupant par batchs de 32 images, c'est-à-dire en petits lots traités simultanément par le modèle à chaque itération, ce qui permet d'optimiser l'utilisation de la mémoire, d'accélérer le processus d'apprentissage et de stabiliser la mise à jour des poids du réseau.

```
[8]: train_generator = train_datagen.flow_from_directory(
         train_dir,
         target_size=(224, 224),
         batch_size=32,
         class_mode='binary'
     )

     test_generator = test_datagen.flow_from_directory(
         test_dir,
         target_size=(224, 224),
         batch_size=32,
         class_mode='binary'
     )

     validation_generator = test_datagen.flow_from_directory(
         valid_dir,
         target_size=(224, 224),
         batch_size=32,
         class_mode='binary'
     )

Found 296 images belonging to 2 classes.
Found 108 images belonging to 2 classes.
Found 26 images belonging to 2 classes.
```

Figure 31: Chargement des images en lots (batches) pour l'entraînement, validation et le teste.

1.2.3 ResNet50 – Données d'imagerie CT scan

Cette étape consiste à utiliser le modèle pré-entraîné ResNet50 comme extracteur de caractéristiques, en gelant ses couches pour ne pas réentraîner les poids. Ensuite, on ajoute des couches personnalisées (Flatten, Dense, Dropout) pour adapter le modèle à notre tâche binaire (prédiction cancer ou non). Enfin, le modèle est compilé avec l'optimiseur Adam, une fonction de perte adaptée à la classification binaire, et l'accuracy comme métrique d'évaluation.

- **Entraînement du modèle :**

Le modèle est entraîné sur les données d'entraînement pendant 10 époques, avec un suivi des performances sur l'ensemble de validation. Cela permet d'ajuster les poids du modèle tout en surveillant sa capacité à généraliser à des données non vues.

- **Sauvegarde du modèle :**

Le modèle entraîné est sauvegardé au format .h5, ce qui permet de le recharger ultérieurement sans avoir à le réentraîner. Ce format est standard pour les modèles Keras.

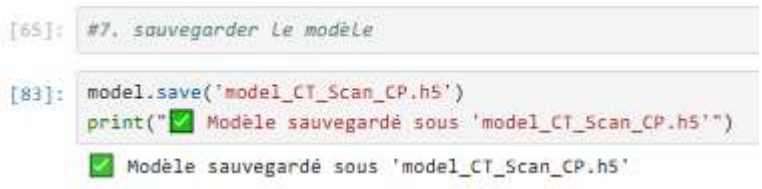


Figure 32: Sauvegarde du modèle entraîné.

1.3 Fusion multimodale

La fusion multimodale a pour objectif de combiner l'information issue de deux sources différentes les données cliniques et les images médicales afin d'améliorer la performance du diagnostic du cancer pulmonaire. L'approche adoptée ici repose sur une fusion tardive (late fusion) à l'aide d'un métamodèle entraîné à partir des prédictions issues de chaque modalité.

1.3.1 Création du dataset multimodal

Afin de constituer un dataset associant pour chaque individu ses données cliniques et une image de CT scan correspondant à sa classe (cancer ou normal), nous avons procédé comme suit :

Le dataset final des données cliniques comprenait 476 patients équilibrés (238 atteints de cancer, 238 sains). Afin de le rendre compatible avec le dataset d'images CT-Scan d'entraînement (296 images réparties équitablement entre 148 patients malades et 148 sains), un sous-échantillonnage aléatoire a été réalisé pour obtenir un jeu de 296 patients cliniques, enregistré sous le nom df_DC_multimodal.csv. Ce fichier a ensuite été combiné avec les images correspondantes dans multimodal_dataset.csv. Les 180 patients restants (90 malades et 90 sains) ont été conservés dans df_reste_patients.csv pour former un jeu de validation. De même, les images CT restantes ont été

réparties dans deux répertoires : test (54 patients : 27 normaux, 27 cancéreux) et valid (26 patients : 13 normaux, 13 cancéreux). Un échantillon de 26 patients cliniques issus de df_reste_patients.csv a été sélectionné pour correspondre à la taille du dossier valid, formant df_DC_multimodal_validation.csv, et chaque patient a été associé à une image CT de la même classe pour produire le fichier final multimodal_dataset_validation.csv.

Il est important de noter que les images contenues dans le dossier valid n'ont pas été altérées ni modifiées, afin de pouvoir être réutilisées plus tard dans le cadre de la plateforme web développée, où elles serviront à illustrer les prédictions du modèle intégré.

[1]:
```python
import pandas as pd
import random
import os

# Charger dataset clinique
df = pd.read_csv('df_DC_multimodal.csv')

# Dossiers d'images
path_cancer = 'C:\\Users\\pc\\Desktop\\data_finale\\train\\cancer'
path_normal = 'C:\\Users\\pc\\Desktop\\data_finale\\train\\normal'

# Lister les fichiers images
images_cancer = os.listdir(path_cancer)
images_normal = os.listdir(path_normal)

# Fonction d'attribution
def assign_image(row):
    if row['LUNG_CANCER'] == 1:
        return os.path.join(path_cancer, random.choice(images_cancer))
    else:
        return os.path.join(path_normal, random.choice(images_normal))

# Ajouter une colonne IMAGE_PATH
df['IMAGE_PATH'] = df.apply(assign_image, axis=1)

print(df.head())
df.to_csv('multimodal_dataset.csv', index=False)
```

Figure 33 : Création du dataset multimodal.

1.3.2 Intégration multimodale : chargement, prétraitement et fusion des modèles

Les modèles préalablement entraînés sont chargés :

- Le modèle Random Forest (random_forest_clinical_model_final.pkl) pour les données cliniques.
- Le modèle ResNet50 (model_CT_Scan_CP.h5) pour la classification d'images.

```
[4]: import joblib
     from tensorflow.keras.models import load_model

     # Le modèle Random Forest
     rf_model = joblib.load('random_forest_clinical_model_final.pkl')

     # Le modèle ResNet50
     RN_model = load_model('model_CT_Scan_CP.h5')
```

Figure 34: Chargement des modèles individuels.

Une fonction de prétraitement des images est définie pour standardiser leur taille (224×224 pixels) et normaliser leurs valeurs (échelle [0, 1]). Les données cliniques et images sont ensuite extraites dans des variables distinctes (X_clinical, X_images), tandis que la variable cible (y) est isolée.

```
[5]: import numpy as np
     from tensorflow.keras.preprocessing import image

     # Fonction de prétraitement des images
     def preprocess_image(img_path):
         img = image.load_img(img_path, target_size=(224, 224))
         img_array = image.img_to_array(img)
         return img_array / 255.0

     # Créer X_clinical et X_images
     X_clinical = df.drop(columns=['LUNG_CANCER', 'IMAGE_PATH'])
     X_images = np.array([preprocess_image(p) for p in df['IMAGE_PATH']])
     y = df['LUNG_CANCER'].values
```

Figure 35: Pipeline de préparation des données cliniques et d'imagerie.

Les noms des colonnes cliniques sont nettoyés (espaces supprimés et ajustés) pour garantir la compatibilité avec le modèle Random Forest. Une vérification finale des colonnes est effectuée.

```
[6]: # Nettoyer les noms de colonnes en supprimant les espaces au début/fin
     df.columns = df.columns.str.strip()
```

```
[7]: X_clinical = df.drop(columns=['LUNG_CANCER', 'IMAGE_PATH'])
```

```
[8]: # Correction des noms de colonnes
     X_clinical = X_clinical.rename(columns={
         'FATIGUE': 'FATIGUE ',       # Ajoute un espace à la fin
         'ALLERGY': 'ALLERGY ',       # Ajoute un espace à la fin

     })

     # Vérification finale
     print("Colonnes après correction :", X_clinical.columns.tolist())
```

```
Colonnes après correction : ['GENDER', 'AGE', 'SMOKING', 'YELLOW_FINGERS',
ING', 'ALCOHOL CONSUMING', 'COUGHING', 'SHORTNESS OF BREATH', 'SWALLOWING
```

Figure 36: Ajustement des noms des colonnes.

Les probabilités de prédiction des deux modèles sont calculées (Fusion des prédictions).

- clinical_preds : Probabilités de cancer issues du Random Forest.
- image_preds : Probabilités issues du ResNet50.

Ces résultats sont fusionnés en un tableau 2D (X_fusion) via np.vstack.

```
[9]: # Prédictions du modèle Random Forest (probabilité classe 1)
clinical_preds = rf_model.predict_proba(X_clinical)[:, 1]

# Prédictions du modèle ResNet50 (probabilité classe 1)
image_preds = RN_model.predict(X_images, batch_size=32).flatten()

10/10 [==============================] - 27s 2s/step

[10]: # Fusionner les prédictions en un seul tableau d'entrée
X_fusion = np.vstack((clinical_preds, image_preds)).T  # shape (n_samples, 2)
```

Figure 37: Fusion des prédictions des deux modèles.

- Un modèle de régression logistique est entraîné sur les prédictions fusionnées (X_fusion), avec une répartition 80/20 pour l'entraînement et le test. Les performances sont évaluées via un rapport de classification.

- Le modèle fusionné est sauvegardé au format .pkl pour une réutilisation ultérieure.

```
[12]: joblib.dump(fusion_model, "fusion_model_final.pkl")

[12]: ['fusion_model_final.pkl']
```

Figure 38: Sauvegarde du modèle de fusion.

2 Modèle ensembliste de diagnostic du CPNPC

2.1 Prétraitement des données

Les images histopathologiques utilisées dans ce projet sont issues d'un jeu de données public disponible sur la plateforme Kaggle, intitulé : Lung and Colon Cancer Histopathological Images [73]. Ce dataset contient des images microscopiques de tissus pulmonaires et colorectaux, acquises à l'aide d'un microscope optique à fort grossissement. Seules les images pulmonaires ont été retenues dans le cadre de ce travail.

Les données sont structurées en cinq dossiers correspondants chacun à une classe histologique :

- lung_aca (adénocarcinome pulmonaire)
- lung_scc (carcinome épidermoïde pulmonaire)
- lung_n (tissu pulmonaire normal)
- colon_aca (adénocarcinome colorectal)
- colon_n (tissu colorectal normal)

Seules les trois premières classes, relatives aux tissus pulmonaires, ont été utilisées ici. Chaque dossier contient 5000 images au format JPEG, de résolution 768 × 768 pixels.

- **Répartition des données sélectionnées**

Tableau 4: Description de la répartition des images pulmonaires utilisées dans le projet.

Classe	Nombre d'images
lung_aca (adénocarcinome)	5000
lung_scc (carcinome)	5000
lung_n (normal)	5000

Avant d'entraîner les modèles d'apprentissage profond, une phase de prétraitement des images est réalisée afin de préparer les données dans un format compatible avec les architectures de réseaux neuronaux convolutifs utilisées. Cette étape inclut l'importation des bibliothèques nécessaires, la définition des répertoires de données, l'inspection des fichiers disponibles, la visualisation des images, ainsi que la génération de jeux de données d'entraînement et de validation à l'aide de la classe ImageDataGenerator. Un prétraitement spécifique à chaque modèle est appliqué à l'aide de la fonction preprocess_input, propre à chaque architecture, afin de normaliser les images en conformité avec les paramètres attendus.

- **Importation des bibliothèques**

Dans cette étape initiale, les bibliothèques essentielles à la manipulation des données, à la visualisation et à la modélisation par apprentissage profond sont importées.

Parmi celles-ci :

- NumPy et Pandas pour la manipulation des données tabulaires
- Matplotlib et Seaborn pour la visualisation graphique des données
- TensorFlow et Keras pour la création, l'entraînement et l'évaluation des modèles de deep learning
- Scikit-learn pour le calcul des matrices de confusion et rapports de classification.

Deux modèles pré-entraînés de la bibliothèque Keras, ResNet50 et DenseNet121, sont également importés afin d'être utilisés dans une approche de transfert learning. Des callbacks comme EarlyStopping et ReduceLROnPlateau sont également définis pour optimiser l'entraînement.

```
[1]: import os
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.metrics import confusion_matrix, classification_report

     import tensorflow as tf
     from tensorflow.keras.models import Sequential, Model
     from tensorflow.keras.layers import Dense, Flatten, Dropout, GlobalAveragePooling2D
     from tensorflow.keras.applications import ResNet50, DenseNet121
     from tensorflow.keras.optimizers import Adam
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
```

Figure 39: Importation des bibliothèques nécessaires à la construction des modèles de classification.

- **Construction du DataFrame à partir des images**

Dans cette cellule, les chemins d'accès (filepaths) ainsi que les étiquettes (labels) associées aux images sont générés automatiquement à partir de la structure des répertoires contenant les données.

Le dossier racine contient trois sous-dossiers représentant les trois classes cibles :

- lung_aca pour l'adénocarcinome pulmonaire,

- lung_n pour le tissu pulmonaire sain,

- lung_scc pour le carcinome épidermoïde.

Chaque image est associée à une étiquette textuelle correspondant à sa classe. L'ensemble des chemins et des étiquettes est ensuite structuré sous forme d'un DataFrame à deux colonnes.

```
[2]: import os
import pandas as pd

filepaths = []
labels = []

data_dir = 'C:\\Users\\pc\\Desktop\\lung_image_sets'

for folder_name in os.listdir(data_dir):
    folder_path = os.path.join(data_dir, folder_name)

    if os.path.isdir(folder_path):
        for file in os.listdir(folder_path):
            if file.lower().endswith(('.jpeg', '.jpg', '.png')):
                fpath = os.path.join(folder_path, file)
                filepaths.append(fpath)

                if folder_name == 'lung_aca':
                    labels.append('Lung Adenocarcinoma')
                elif folder_name == 'lung_n':
                    labels.append('Lung Benign Tissue')
                elif folder_name == 'lung_scc':
                    labels.append('Lung Squamous Cell Carcinoma')

# Vérification
print("Nombre de filepaths :", len(filepaths))
print("Nombre de labels :", len(labels))

# Création du DataFrame
df = pd.DataFrame({
    'filepaths': filepaths,
    'labels': labels
})

df
```

Figure 40: Génération du DataFrame contenant les chemins des images et leurs étiquettes respectives.

- **Séparation du jeu de données en ensembles d'entraînement, de validation et de test**

Cette cellule réalise la division du jeu de données en trois sous-ensembles :

- Ensemble d’entraînement (80 %)
- Ensemble de validation (10 %)
- Ensemble de test (10 %)

La séparation est effectuée de manière stratifiée, garantissant ainsi la même répartition des classes dans chacun des sous-ensembles. La fonction train_test_split de Scikit-learn est utilisée avec une graine aléatoire (random_state) pour assurer la reproductibilité.

```
[3]: from sklearn.model_selection import train_test_split

strat = df['labels']
train_df, dummy_df = train_test_split(df,  train_size= 0.8, shuffle= True, random_state= 123, stratify= strat)

# valid and test dataframe
strat = dummy_df['labels']
valid_df, test_df = train_test_split(dummy_df,  train_size= 0.5, shuffle= True, random_state= 123, stratify= strat)
```

Figure 41: Répartition stratifiée du dataset groupe d'entraînement, de validation et de test.

- **Définition d'une fonction personnalisée d'affichage pour les diagrammes circulaires**

Cette cellule contient une fonction utilitaire appelée custom_autopct, permettant de personnaliser l'affichage des pourcentages et des valeurs absolues dans les graphiques circulaires. Elle est utilisée pour enrichir la lisibilité des diagrammes représentant la distribution des classes.

```
[4]: def custom_autopct(pct, data):
        total = sum(data)
        val = int(round(pct * total / 100.0))
        return "{:.1f}%\n({:d})".format(pct, val)
```

Figure 42: Fonction de formatage personnalisée pour les visualisations circulaires.

- **Analyse de l'équilibre des classes dans le jeu de données**

Une visualisation sous forme de diagramme circulaire est générée pour représenter la répartition des étiquettes dans le dataset global. La fonction définie précédemment est utilisée pour afficher à la fois les pourcentages et les effectifs de chaque classe. Cette visualisation permet de confirmer visuellement si les données sont équilibrées entre les trois catégories.

```
[5]: data_balance = df.labels.value_counts()
     # pie chart for Training data balance
     plt.pie(
         data_balance,
         labels=data_balance.index,
         autopct=lambda pct: custom_autopct(pct, data_balance),
         colors=["#2092E6", "#6D8CE6", "#20D0E6"]
     )
     plt.title("data balance")
     plt.axis("equal")
     plt.show()
```

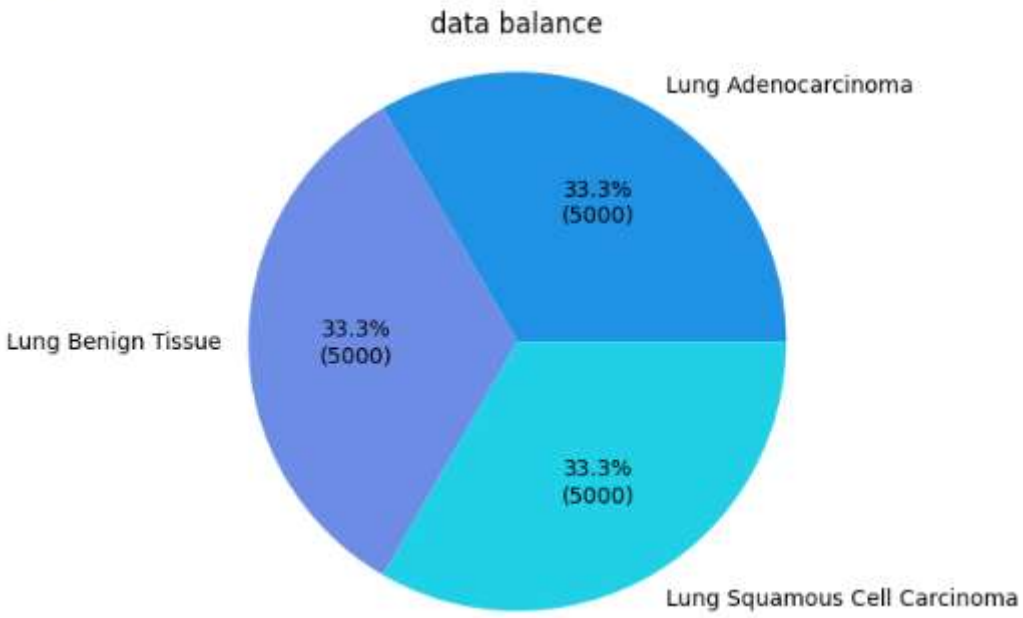


Figure 43: Répartition des classes dans l'ensemble des données utilisées.

- **Analyse de la répartition des classes dans l'ensemble d'entraînement**

Un diagramme circulaire est généré pour visualiser la distribution des classes au sein de l'ensemble d'entraînement. Cette étape permet de vérifier que le fractionnement du jeu de données respecte bien la stratification initiale, assurant un équilibre des classes indispensable à un entraînement efficace du modèle.

```
[6]: training_data_balance = train_df.labels.value_counts()
# pie chart for Training data balance
plt.pie(
    training_data_balance,
    labels=training_data_balance.index,
    autopct=lambda pct: custom_autopct(pct, training_data_balance),
    colors=["#2092E6", "#6D8CE6", "#20D0E6"]
)
plt.title("Training data balance")
plt.axis("equal")
plt.show()
```

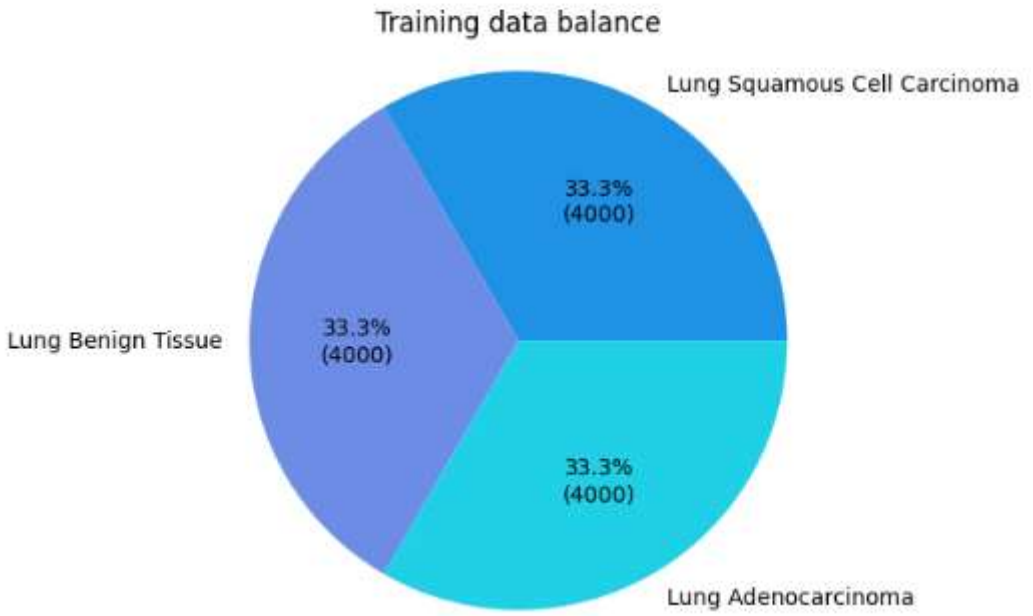


Figure 44: Répartition des classes dans l'ensemble d'entraînement.

- **Analyse de la répartition des classes dans l'ensemble de test**

De la même manière que pour l'ensemble d'entraînement, un diagramme circulaire illustre la distribution des classes dans l'ensemble de test. Cette visualisation permet de s'assurer que le jeu de test conserve une représentation équilibrée des catégories, garantissant ainsi une évaluation fiable des performances du modèle.

```
[7]: test_data_balance = test_df.labels.value_counts()
# pie chart for Training data balance
plt.pie(
    test_data_balance,
    labels=test_data_balance.index,
    autopct=lambda pct: custom_autopct(pct, test_data_balance),
    colors=["#2092E6", "#6D8CE6", "#20D0E6"]
)
plt.title("test data balance")
plt.axis("equal")
plt.show()
```

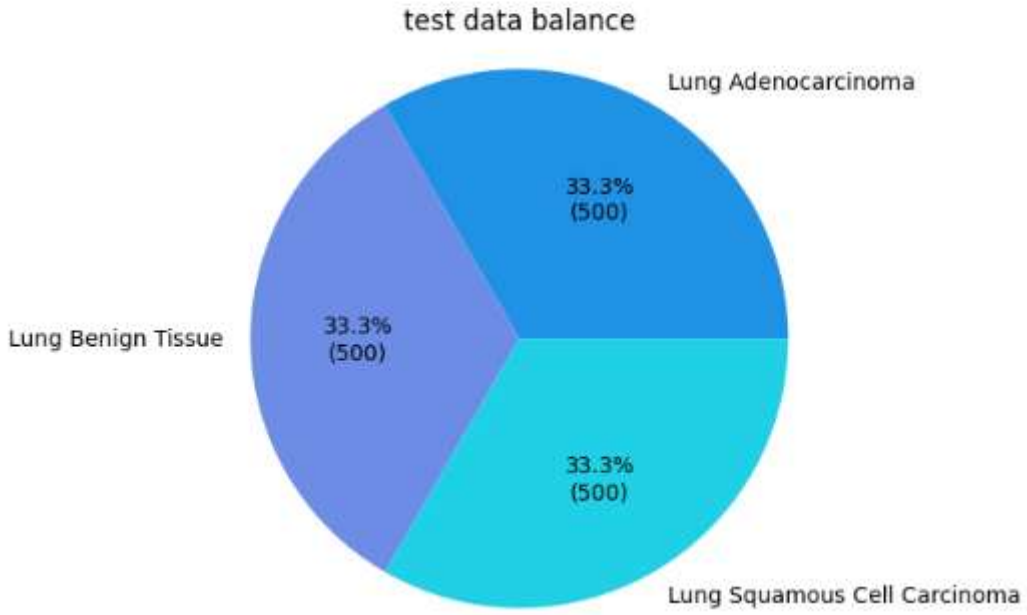


Figure 45: Répartition des classes dans l'ensemble de test.

- **Analyse de la répartition des classes dans l'ensemble de validation**

Un diagramme circulaire est également utilisé pour représenter la distribution des classes dans l'ensemble de validation. Cela assure que les données utilisées pour le réglage des hyperparamètres conservent une bonne représentativité de toutes les classes.

```
[8]: Valid_data_balance = valid_df.labels.value_counts()
     # pie chart for Training data balance
     plt.pie(
         Valid_data_balance,
         labels=Valid_data_balance.index,
         autopct=lambda pct: custom_autopct(pct, Valid_data_balance),
         colors=["#2092E6", "#6D8CE6", "#20D0E6"]
     )
     plt.title("Valid data balance")
     plt.axis("equal")
     plt.show()
```

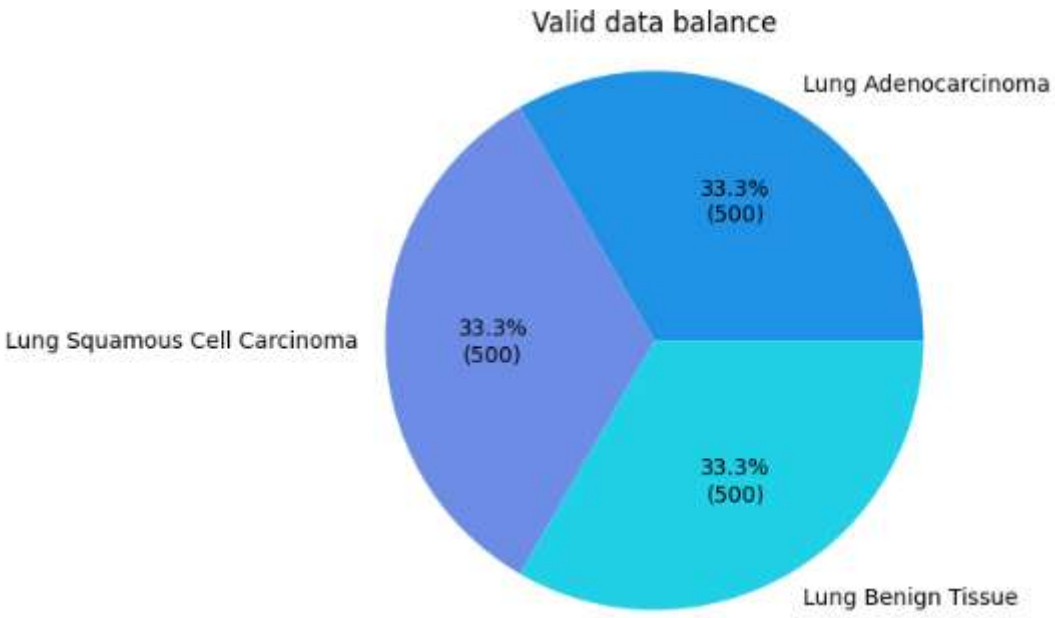


Figure 46: Répartition des classes dans l'ensemble de validation.

- **Définition des paramètres pour le prétraitement et la génération des lots d'images**

Les paramètres relatifs au traitement des images sont définis, notamment la taille des images redimensionnées (224 × 224 pixels), le nombre de canaux couleur (3 pour RGB) et la taille des lots d'images (batch size) pour l'entraînement.

Des objets ImageDataGenerator sont instanciés pour gérer la lecture, la mise à l'échelle et l'augmentation potentielle des images dans les différentes phases d'apprentissage.

```
[9]: # crobed image size
     batch_size = 64
     img_size = (224, 224)
     channels = 3
     img_shape = (img_size[0], img_size[1], channels)

     tr_gen = ImageDataGenerator()
     ts_gen = ImageDataGenerator()
```

Figure 47: Initialisation des paramètres d'entrée et création des générateurs d'images.

- **Création des générateurs d'images pour l'entraînement, la validation et le test**

Cette étape consiste à créer des générateurs à partir des DataFrames contenant les chemins d'accès et les étiquettes des images.

Les images sont automatiquement chargées, redimensionnées à la taille spécifiée, et prétraitées en lots adaptés à l'entraînement (avec mélange aléatoire pour l'entraînement et la validation). Pour l'ensemble de test, le mélange est désactivé afin d'assurer une évaluation cohérente. La sortie des générateurs est configurée pour fournir des étiquettes au format catégoriel, compatible avec la classification multi-classes.

```
[10]: train_gen = tr_gen.flow_from_dataframe( train_df, x_col= 'filepaths', y_col= 'labels', target_size= img_size, class_mode= 'categorical',
                                    color_mode= 'rgb', shuffle= True, batch_size= batch_size)

      valid_gen = ts_gen.flow_from_dataframe( valid_df, x_col= 'filepaths', y_col= 'labels', target_size= img_size, class_mode= 'categorical',
                                      color_mode= 'rgb', shuffle= True, batch_size= batch_size)

      test_gen = ts_gen.flow_from_dataframe( test_df, x_col= 'filepaths', y_col= 'labels', target_size= img_size, class_mode= 'categorical',
                                    color_mode= 'rgb', shuffle= False, batch_size= batch_size)

      Found 12000 validated image filenames belonging to 3 classes.
      Found 1500 validated image filenames belonging to 3 classes.
      Found 1500 validated image filenames belonging to 3 classes.
```

Figure 48: Générateurs d'images pour les phases d'entraînement, validation et test.

- **Visualisation d'un échantillon de l'ensemble d'entraînement**

Cette cellule permet d'afficher un échantillon visuel composé de seize images provenant de l'ensemble d'entraînement. Chaque image est accompagnée de son étiquette correspondante, extraite à partir de la classe prédominante dans le vecteur de vérité terrain (one-hot encoding).

Cette visualisation vise à inspecter manuellement la diversité des données d'apprentissage, à valider le bon étiquetage des classes et à confirmer que les images sont correctement prétraitées et normalisées.

```
[11]: g_dict = train_gen.class_indices
      classes = list(g_dict.keys())
      images, labels = next(train_gen)

      plt.figure(figsize= (20, 20))

      for i in range(16):
          plt.subplot(4, 4, i + 1)
          image = images[i] / 255
          plt.imshow(image)
          index = np.argmax(labels[i])
          class_name = classes[index]
          plt.title(class_name, color= 'blue', fontsize= 12)
          plt.axis('off')
      plt.show()
```

Figure 49: Échantillon d'images issues de l'ensemble d'entraînement avec leurs étiquettes.

- **Visualisation d'un échantillon de l'ensemble de validation**

À l'instar de l'ensemble d'entraînement, un échantillon de seize images est extrait de l'ensemble de validation.

Chaque image est affichée avec son étiquette associée. Cela permet de vérifier visuellement que la cohérence des classes est maintenue dans cet ensemble utilisé pour ajuster les paramètres du modèle durant l'apprentissage, sans contribuer directement à la mise à jour des poids.

```
[12]: g_dict = train_gen.class_indices
      classes = list(g_dict.keys())
      images, labels = next(valid_gen)

      plt.figure(figsize= (20, 20))

      for i in range(16):
          plt.subplot(4, 4, i + 1)
          image = images[i] / 255
          plt.imshow(image)
          index = np.argmax(labels[i])
          class_name = classes[index]
          plt.title(class_name, color= 'blue', fontsize= 12)
          plt.axis('off')
      plt.show()
```

Figure 50: Échantillon d'images issues de l'ensemble de validation avec leurs étiquettes.

- **Visualisation d'un échantillon de l'ensemble de test**

Cette cellule permet de visualiser seize images issues de l'ensemble de test, accompagnées de leurs classes réelles.

Cette étape facilite l'analyse exploratoire du jeu de test, utilisé exclusivement pour évaluer la capacité de généralisation du modèle après l'entraînement. Elle permet aussi de confirmer la représentativité et la qualité des images dans ce sous-ensemble.

```
[13]: g_dict = train_gen.class_indices
      classes = list(g_dict.keys())
      images, labels = next(test_gen)

      plt.figure(figsize= (20, 20))

      for i in range(16):
          plt.subplot(4, 4, i + 1)
          image = images[i] / 255
          plt.imshow(image)
          index = np.argmax(labels[i])
          class_name = classes[index]
          plt.title(class_name, color= 'blue', fontsize= 12)
          plt.axis('off')
      plt.show()
```

Figure 51: Échantillon d'images issues de l'ensemble de test avec leurs étiquettes.

- **Définition des fonctions de visualisation et d'évaluation du modèle**

Plusieurs fonctions d'analyse des performances des modèles entraînés :

- La fonction model_performance permet de visualiser les courbes d'apprentissage en traçant l'évolution de la perte et de la précision sur les ensembles d'entraînement et de validation au fil des époques.
- La fonction model_evaluation évalue quantitativement un modèle donné sur les ensembles d'entraînement, de validation et de test, en restituant les scores de perte et de précision.
- La fonction plot_confusion_matrix génère une matrice de confusion afin d'examiner en détail les performances du modèle en termes de prédictions correctes et d'erreurs par classe.

Cette matrice permet d'identifier les classes pour lesquelles le modèle présente des confusions fréquentes.

- **Initialisation des paramètres du modèle et définition de l'early stopping**

Cette cellule fixe les paramètres fondamentaux du modèle de classification :

- La taille d'entrée des images (224 × 224 pixels avec 3 canaux RGB),
- Le nombre de classes à prédire, déterminé dynamiquement à partir des classes présentes dans le générateur d'entraînement,
- Le nombre d'époques d'entraînement,
- La configuration du mécanisme d'early stopping, utilisé pour interrompre prématurément l'entraînement lorsque la performance en validation cesse de s'améliorer. Cette stratégie permet d'éviter le surapprentissage tout en conservant les poids correspondant aux meilleures performances atteintes.

```
[15]: img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)
class_count = len(list(train_gen.class_indices.keys()))
epochs = 10
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=5,
    restore_best_weights=True,
    verbose=1
)
```

Figure 52: Définition de la configuration d'entrée et des paramètres de régularisation pour l'entraînement du modèle.

2.2 Construction du modèle de classification basé sur ResNet50

Le modèle ResNet50 pré-entraîné sur ImageNet est utilisé comme base. Sa dernière couche de classification est retirée (include_top=False) pour permettre une personnalisation adaptée à notre jeu de données.

Le modèle de base est gelé (trainable=False) afin de préserver les poids appris, ce qui est particulièrement utile lors d'un entraînement initial avec un volume de données limité.

Des couches supplémentaires sont ajoutées au sommet du modèle.

- **Entraînement du modèle ResNet50**

Le modèle est entraîné sur l'ensemble d'entraînement à l'aide du générateur train_gen.

L'apprentissage est réalisé sur un nombre d'époques défini, avec une validation effectuée à chaque époque à l'aide de test_gen.

Le paramètre shuffle=False garantit que l'ordre des images dans l'ensemble de validation est respecté, ce qui est important pour la cohérence des évaluations.

Le callback early_stopping est activé afin d'arrêter l'entraînement dès que la performance sur l'ensemble de validation cesse de s'améliorer.

- **Calcul dynamique de la taille optimale des lots pour le test**

Cette cellule calcule automatiquement la taille la plus grande possible du lot pour le jeu de test, tout en respectant la contrainte que le nombre total d'images soit divisible sans reste. Cela permet d'assurer une prédiction complète sur l'ensemble de test sans troncature.

Elle fixe également le nombre d'étapes nécessaires (test_steps) pour traiter l'intégralité de ce jeu.

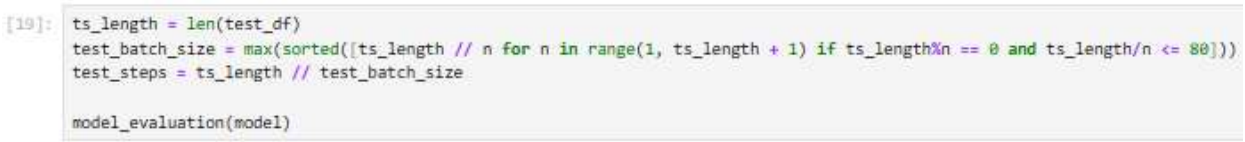


Figure 53: Détermination dynamique de la taille des lots pour l'évaluation du modèle.

- **Évaluation globale du modèle**

La fonction model_evaluation est appelée pour mesurer la perte (loss) et la précision (accuracy) sur les ensembles d'entraînement, de validation et de test.

Ces résultats chiffrés permettent de quantifier la performance globale du modèle et de détecter d'éventuels écarts entre les ensembles (sous-apprentissage ou surapprentissage).

- **Visualisation des courbes d'apprentissage**

La fonction model_performance est utilisée pour tracer les courbes d'évolution de la perte et de la précision au fil des époques.

Ces visualisations permettent d'analyser la convergence du modèle, la stabilité de l'entraînement et l'évolution du comportement sur l'ensemble de validation.

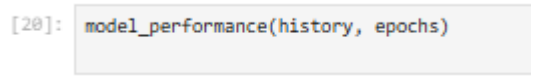


Figure 54: Évaluation du modèle ResNet50 sur les différents sous-ensembles de données.

- **Prédiction sur l'ensemble de test**

Le modèle prédit les probabilités d'appartenance à chaque classe pour les images de test. Les prédictions finales sont obtenues en appliquant argmax pour extraire la classe avec la probabilité la plus élevée pour chaque image.

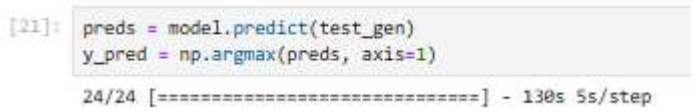


Figure 55: Prédictions de classe effectuées par le modèle sur l'ensemble de test.

- **Génération de la matrice de confusion**

Cette cellule prépare la liste des classes à partir du générateur test_gen et appelle la fonction plot_confusion_matrix pour afficher la matrice de confusion.

Cette matrice met en évidence les performances du modèle pour chaque classe, en distinguant les prédictions correctes des erreurs de classification. Elle constitue un outil essentiel pour diagnostiquer les faiblesses spécifiques du modèle.

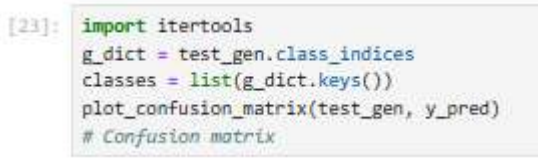


Figure 56: Matrice de confusion pour l'évaluation des prédictions du modèle.

- **Rapport de classification détaillé**

Le rapport de classification généré par classification_report fournit des mesures de performance détaillées :

- Précision (precision) ;
- Rappel (recall) ;
- Score F1 (f1-score) ;

- Support (nombre d'éléments par classe).

Ce rapport permet d'évaluer finement les performances pour chaque catégorie, mettant en lumière les éventuelles disparités de traitement entre les classes.

```
[24]: print(classification_report(test_gen.classes, y_pred, target_names= classes))
```

Figure 57: Rapport de classification du modèle sur l'ensemble de test.

- **Sauvegarde du modèle entraîné**

Le modèle ResNet50 personnalisé est sauvegardé au format HDF5 (.h5).
Cela permet une réutilisation ultérieure du modèle sans nécessiter un réentraînement, facilitant ainsi le déploiement ou l'évaluation comparative avec d'autres architectures.

```
[25]: model.save('model_ResNet50.h5')
```

Figure 58: Sauvegarde du modèle entraîné au format HDF5.

2.3 Construction du modèle de classification basé sur DenseNet121

Une procédure analogue à celle employée pour ResNet50 a été utilisée pour construire le modèle basé sur DenseNet121, avec quelques ajustements spécifiques :

- Le modèle de base utilisé est DenseNet121 (au lieu de ResNet50), pré-entraîné sur ImageNet.
- La structure des couches ajoutées reste identique sans changement.
- Le reste du pipeline (entraînement, prédiction, évaluation, visualisation) est rigoureusement le même.

2.4 La construction du modèle ensembliste

Un modèle ensembliste est construit à partir de deux modèles pré-entraînés : ResNet50 et DenseNet121. Ces deux architectures, déjà ajustées individuellement sur le jeu d'images pulmonaires, sont fusionnées afin d'exploiter la complémentarité de leurs représentations.

Les deux modèles sont chargés depuis leurs fichiers .h5, sans recompiler leur architecture (compile=False). Chaque couche de chaque modèle est renommée de manière unique pour éviter tout conflit lors de la fusion (resnet_i_nom et densenet_i_nom).

Un input tensor commun est défini, sur lequel les deux modèles sont appliqués. Leurs sorties respectives sont fusionnées à l'aide d'une couche Average, permettant une agrégation des prédictions de manière équilibrée.

Le modèle final est compilé avec :

- L'optimiseur Adamax, reconnu pour sa stabilité sur les petits lots ;

- La fonction de perte categorical_crossentropy ;
- La métrique d'évaluation accuracy.

Ce modèle vise à améliorer la robustesse et réduire la variance des prédictions individuelles.

```python
# 5. Chargement et préparation des modèles
resnet_model = load_model('model_ResNet50.h5', compile=False)
densenet_model = load_model('model_DenseNet121.h5', compile=False)

# Renommage des modèles et couches
resnet_model._name = "resnet50_model"
densenet_model._name = "densenet121_model"

for i, layer in enumerate(resnet_model.layers):
    layer._name = f"resnet_{i}_{layer.name}"

for i, layer in enumerate(densenet_model.layers):
    layer._name = f"densenet_{i}_{layer.name}"

# 6. Création du modèle ensembliste
def create_ensemble(model1, model2, input_shape):
    input_tensor = Input(shape=input_shape, name='ensemble_input')
    out1 = model1(input_tensor)
    out2 = model2(input_tensor)
    output = Average(name='ensemble_average')([out1, out2])
    model = Model(inputs=input_tensor, outputs=output, name='ensemble_model')
    return model

ensemble = create_ensemble(resnet_model, densenet_model, img_shape)

# 7. Compilation
ensemble.compile(
    optimizer=Adamax(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Figure 59 : Architecture du modèle ensembliste basé sur ResNet50 et DenseNet121.

- **Préparation des données pour l'évaluation du modèle ensembliste**

Le jeu de test, précédemment isolé lors du découpage initial des données, est injecté dans un générateur d'images ImageDataGenerator, configuré pour :

- Ne pas appliquer d'augmentation (augmentation=None) ;
- Ne pas mélanger les données (shuffle=False) afin de préserver l'ordre des échantillons ;
- Générer des lots d'images normalisées de taille (224, 224) avec 3 canaux RGB ;
- Produire des étiquettes sous forme catégorielle (one-hot encoded).

Ce générateur assure une entrée compatible avec l'architecture du modèle ensembliste.

```
# 8. Préparation des données de test
ts_gen = ImageDataGenerator()
test_gen = ts_gen.flow_from_dataframe(
    test_df,
    x_col='filepaths',
    y_col='labels',
    target_size=img_size,
    class_mode='categorical',
    color_mode='rgb',
    shuffle=False,
    batch_size=batch_size
)
```

Figure 60: Générateur d'images pour l'évaluation du modèle ensembliste.

- **Évaluation du modèle ensembliste**

L'évaluation du modèle ensembliste est réalisée à l'aide d'une fonction dédiée evaluate_ensemble(), qui permet d'analyser les performances globales du système sur le jeu de test. Cette fonction effectue plusieurs étapes clés :

- Prédictions sur le jeu de test : les sorties du modèle ensembliste sont fusionnées, puis transformées en classes prédictives via la fonction argmax, ce qui permet d'obtenir les étiquettes finales.

- Génération d'un rapport de classification : la fonction classification_report fournit les métriques principales (précision, rappel, F1-score et support) pour chaque classe, permettant une évaluation quantitative détaillée, notamment en cas de classes déséquilibrées.

- Affichage de la matrice de confusion : celle-ci met en évidence les correspondances et erreurs entre les classes prédites et les classes réelles. L'utilisation d'un affichage coloré améliore la lisibilité et permet d'identifier rapidement les confusions fréquentes entre certaines classes.

- Évaluation globale du modèle : les métriques standards telles que la perte (loss) et l'exactitude (accuracy) sont calculées à l'aide de la méthode evaluate(), fournissant une mesure globale des performances sur l'ensemble de test.

Cette évaluation complète permet de juger la qualité du modèle ensembliste (basé sur la combinaison ResNet50 + DenseNet121) et de le comparer efficacement aux modèles individuels.

```python
# 9. Fonction d'évaluation
def evaluate_ensemble(model, test_gen):
    # Prédictions
    preds = model.predict(test_gen)
    y_pred = np.argmax(preds, axis=1)

    # Rapport de classification
    print("\nClassification Report:")
    print(classification_report(test_gen.classes, y_pred, target_names=list(test_gen.class_indices.keys())))

    # Matrice de confusion
    plot_confusion_matrix(test_gen, y_pred)

    # Évaluation standard
    test_score = model.evaluate(test_gen, verbose=1)
    print("\nTest Loss:", test_score[0])
    print("Test Accuracy:", test_score[1])

# 10. Évaluation du modèle
print("\nÉvaluation du modèle ensembliste:")
evaluate_ensemble(ensemble, test_gen)
```

Figure 61: Évaluation du modèle ensembliste avec fonction d'analyse des performances globales

3 Création du site web DiagnoLung

Dans le cadre de ce projet, une plateforme web nommée *DiagnoLung* a été développée afin d'assister les professionnels de santé dans deux axes complémentaires : le diagnostic précoce du cancer pulmonaire à partir de données cliniques et les images de CT scan , et l'identification du cancer pulmonaire non à petites cellules (CPNPC), en s'appuyant sur des approches multimodales d'intelligence artificielle. Ce site web a été développé à l'aide du framework Django dans l'éditeur de code Visual Studio Code (VS Code), et structuré en plusieurs applications pour une meilleure organisation.

3.1 Structure et organisation du site web

Le projet Django a été initialisé dans un environnement virtuel pour isoler les dépendances du projet. Il est structuré en trois applications principales :

3.1.1 our_site

cette application gère les pages statiques du site, notamment la page d'accueil (home), la présentation du projet (about), les services proposés (services), l'équipe (Team), ainsi que la page de contact (Contact).

3.1.2 account

cette application s'occupe de l'authentification, de l'enregistrement et de la gestion des comptes des médecins utilisateurs du site.

3.1.3 Diagnostic

Cette application constitue le cœur fonctionnel de la plateforme. Elle intègre une page de détection multimodale du cancer du poumon, basée sur l'analyse des données cliniques et des images médicales (CT Scan), ainsi qu'une page dédiée à un modèle ensembliste de détection du CPNPC à partir d'images histopathologiques. Elle permet également l'affichage des résultats aux médecins, qui peuvent ensuite transmettre le compte rendu au patient.

Pour l'interface de la plateforme, deux modèles (templates) issus de sources libres ont été utilisés

- Le premier template, issu de BootstrapMade [1], a été utilisé comme base pour la structure globale du site.
- Le second template, provenant d'Elzero Web School [2], a été intégré pour la gestion des interfaces de profils des médecins et des patients.

Les deux templates ont été personnalisés et adaptés selon les besoins fonctionnels et esthétiques de la plateforme.

3.2 Langages de programmation utilisés

Le développement de la plateforme a nécessité l'utilisation de plusieurs langages de programmation, chacun ayant un rôle spécifique :

- Python : utilisé pour le développement du logique métier, la gestion des vues, des modèles de données, et l'intégration des modèles de prédiction via le framework Django.
- HTML (HyperText Markup Language) : utilisé pour structurer le contenu des pages web.
- CSS (Cascading Style Sheets) : utilisé pour le stylage et la mise en forme des pages web, assurant une interface utilisateur agréable et responsive.
- JavaScript : utilisé pour dynamiser certaines parties du site, gérer des interactions et améliorer l'expérience utilisateur.

3.3 Base de données : PostgreSQL 17

Pour la gestion des données, nous avons utilisé PostgreSQL version 17, un système de gestion de base de données relationnelle fiable, sécurisé et performant. Il permet de stocker efficacement les données des utilisateurs (médecins et patients) ainsi que les résultats de prédiction.

4 Matériel et logiciel utilisé

4.1 Configuration matérielle

Tableau 5: Configurations matérielles des machines utilisées.

Paramètre	PC 1	PC 2
Modèle du système	Dell Latitude 5480	Dell Latitude E7470
Système d'exploitation	Windows 10 Professionnel 64 bits (build 19045)	Windows 10 Professionnel 64 bits (build 19045)
Processeur	Intel(R) Core(TM) i5-7300U @ 2.60GHz (4 CPUs)	Intel(R) Core(TM) i5-6300U @ 2.40GHz (4 CPUs)
Fréquence du processeur	~2.7 GHz	~2.5 GHz
Mémoire RAM	8 Go (8192 MB)	8 Go (8192 MB)

4.2 Environnement de développement

Tableau 6: Environnement de développement.

Outil / Logiciel	Version (PC1)	Version (PC2)	Utilisation principale
Visual Studio Code	1.100.3 (user setup)	1.90.1 (user setup)	Développement de la plateforme web (interface)
Anaconda	conda 25.3.0	conda 24.1.2	Gestion des environnements et bibliothèques Python
Jupyter Notebook	7.3.2	7.3.2	Développement et exécution des modèles d'IA

4.3 Bibliothèques et frameworks

Tableau 7: Bibliothèques et frameworks utilisés.

Bibliothèque / Framework	Version (PC 1)	Version (PC 2)	Utilisation principale
Python	3.10.11 (Anaconda)	3.10.8 (Conda-forge)	Langage principal pour le développement
NumPy	1.24.3	1.24.3	Manipulation de tableaux numériques
Pandas	2.2.3	2.2.3	Manipulation et analyse des données (DataFrames)
Matplotlib	3.10.1	3.10.3	Visualisation de données et images
Seaborn	0.13.2	0.13.2	Visualisation statistique avancée
Django	5.1.3	5.2.3	Développement de la plateforme web DiagnoLung
Scikit-learn	1.6.1	1.6.1	Prétraitement, séparation des données, évaluation des modèles

TensorFlow	2.10.0	2.10.0	Framework de deep learning pour la modélisation
Keras (via TensorFlow)	Inclus (2.10.0)	Inclus (2.10.0)	Construction de modèles CNN avec ResNet50 et DenseNet121
os (module Python natif)	os (standard)	nt (standard)	Gestion des chemins, répertoires et fichiers système

Résultats et Discussion

Résultats et Discussion

Afin d'évaluer la performance des modèles développés dans ce projet, cette partie est structurée selon les différentes approches mises en œuvre : l'approche multimodale intégrant à la fois données cliniques et imagerie, ainsi que l'approche ensembliste combinant deux modèles de deep learning appliqués aux images de biopsie.

Pour chaque approche, les principales métriques d'évaluation sont présentées, notamment l'accuracy, la précision, le rappel et le F1-score, accompagnées de visualisations telles que les courbes d'apprentissage et les matrices de confusion. Ces résultats permettent de mesurer la capacité de chaque modèle à distinguer les cas de cancer pulmonaire, à anticiper les diagnostics précoces, et le cancer pulmonaire non à petites cellules (CPNPC). Les performances des différentes combinaisons de données et de modèles sont également comparées afin d'identifier l'approche la plus efficace et la plus fiable pour une application clinique.

1 Modèle multimodal de diagnostic précoce

1.1 Résultats du modèle basé sur les données cliniques (Random Forest)

L'évaluation du modèle a été réalisée selon trois axes complémentaires : le rapport de classification, la matrice de confusion et la courbe ROC, afin de fournir une analyse complète de ses performances sur l'ensemble de test.

```
Test Accuracy Finale: 0.96
Rapport de classification (Test):
              precision    recall  f1-score   support

           0       0.92      1.00      0.96        48
           1       1.00      0.92      0.96        48

    accuracy                           0.96        96
   macro avg       0.96      0.96      0.96        96
weighted avg       0.96      0.96      0.96        96

Matrice de confusion (Test):
 [[48  0]
 [ 4 44]]
```

Figure 62: Performance du modèle Random Forest sur données cliniques - Matrice de confusion- Rapport de Classification (Jeu de test).

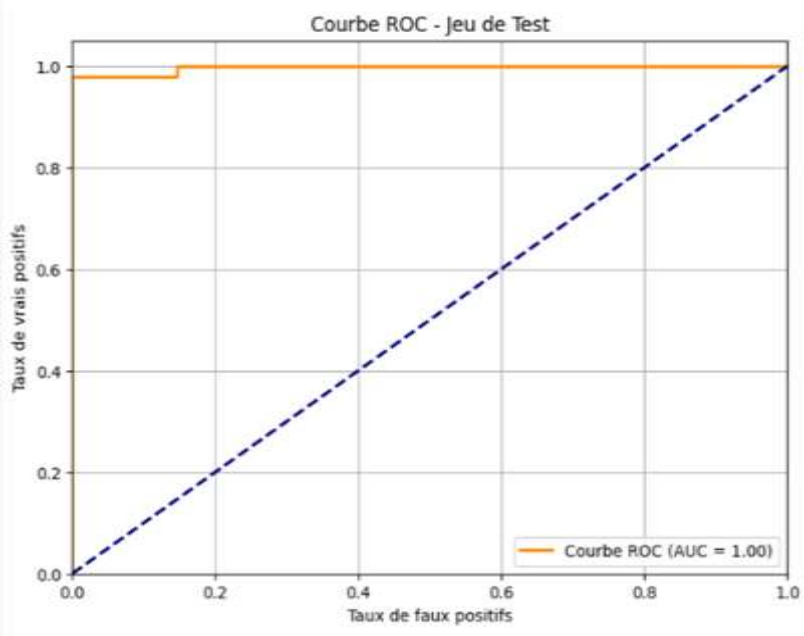


Figure 63: Courbe ROC effectué sur le jeu du test.

Sur l'ensemble de test, le modèle Random Forest a montré une excellente capacité à prédire la présence ou l'absence de cancer pulmonaire, avec une accuracy de 96 %, ce qui signifie que 96 % des prédictions étaient correctes.

Dans un premier temps, le rapport de classification met en évidence des performances équilibrées entre les deux classes :

- **Classe 0 (patients non atteints de cancer)** :
  - Précision : 0.92 → 92 % des patients prédits comme « non atteints » le sont réellement.
  - Rappel : 1.00 → Tous les patients réellement non atteints ont bien été détectés par le modèle.
  - F1-score : 0.96 → Bon équilibre entre précision et rappel pour cette classe.
- **Classe 1 (patients atteints de cancer)** :
  - Précision : 1.00 → Tous les patients prédits comme « atteints » étaient bien atteints.
  - Rappel : 0.92 → Le modèle a détecté 92 % des vrais cas de cancer, mais en a manqué 8 %.
  - F1-score : 0.96 → Excellent compromis global pour la classe positive.
- **Moyennes globales :**
  - Macro average : 0.96 → Moyenne simple des scores des deux classes, montrant un bon équilibre de performance.
  - Weighted average : 0.96 → Moyenne pondérée tenant compte du nombre d'exemples dans chaque classe, ce qui confirme la stabilité du modèle même en cas de classes déséquilibrées.

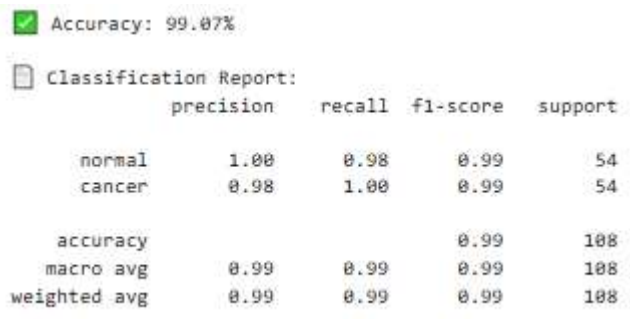
Ensuite, les résultats issus de la matrice de confusion confirment ces observations :

- Accuracy (0.96) : Cela indique que 96 % des cas ont été correctement classés par le modèle (cancer ou non-cancer).
- Vrais positifs (VP = 44) : Le modèle a correctement identifié 44 patients atteints de cancer.
- Faux positifs (FP = 0) : Aucun patient non atteint de cancer n'a été faussement classé comme atteint, ce qui montre une spécificité parfaite.
- Vrais négatifs (VN = 48) : Le modèle a correctement identifié 48 patients non atteints de cancer.
- Faux négatifs (FN = 4) : 4 patients atteints de cancer ont été mal classés comme non atteints, ce qui indique une petite perte de sensibilité.

Enfin, l'analyse de la courbe ROC montre une capacité de discrimination parfaite, cela indique que le modèle est capable de bien discriminer entre les patients atteints de cancer pulmonaire et ceux qui ne le sont pas avec une AUC (Area Under Curve) égale à 1.00.

1.2 Résultats du modèle basé sur les images de CT scan (ResNet50)

L'évaluation du modèle ResNet50 appliqué aux images de scanner thoracique a été réalisée selon plusieurs axes complémentaires : le rapport de classification, la matrice de confusion, ainsi que l'analyse des courbes d'apprentissage, incluant la courbe de précision (convergence de l'entraînement et de la validation) et la courbe de perte (évolution de la perte au cours de l'entraînement), afin de fournir une vue globale de ses performances.



Accuracy: 99.07%

Classification Report:

	precision	recall	f1-score	support
normal	1.00	0.98	0.99	54
cancer	0.98	1.00	0.99	54
accuracy			0.99	108
macro avg	0.99	0.99	0.99	108
weighted avg	0.99	0.99	0.99	108

Figure 64 : Rapport de Classification du modèle ResNet50 sur les images de CT Scan (Jeu de test).

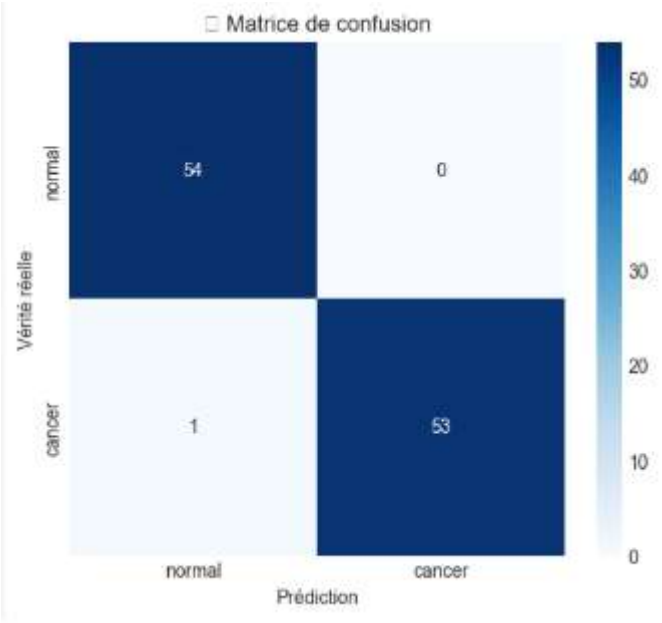


Figure 65 : Matrice de Confusion du modèle ResNet50 sur les images de CT Scan (Jeu de test).

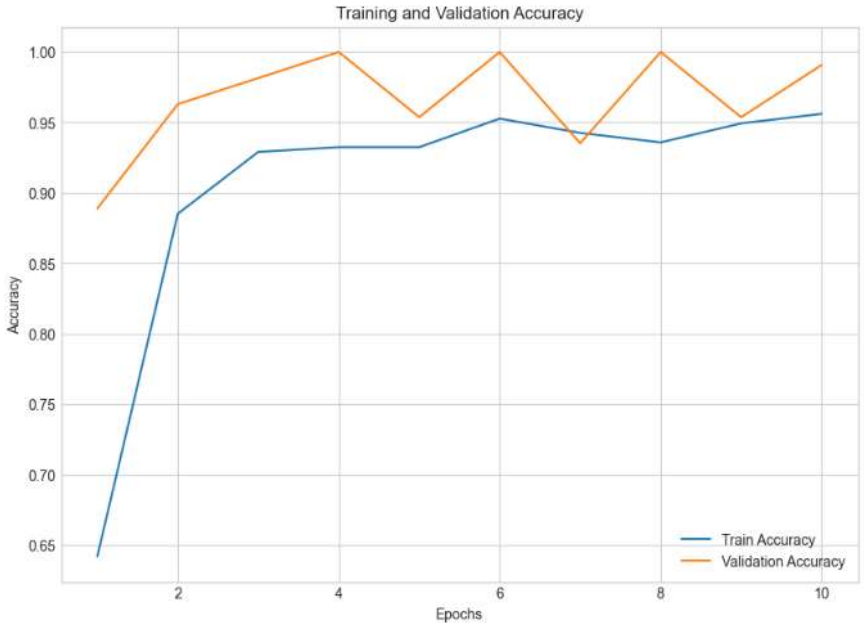


Figure 66: Courbe de Précision : Convergence de l'entraînement et de la validation.

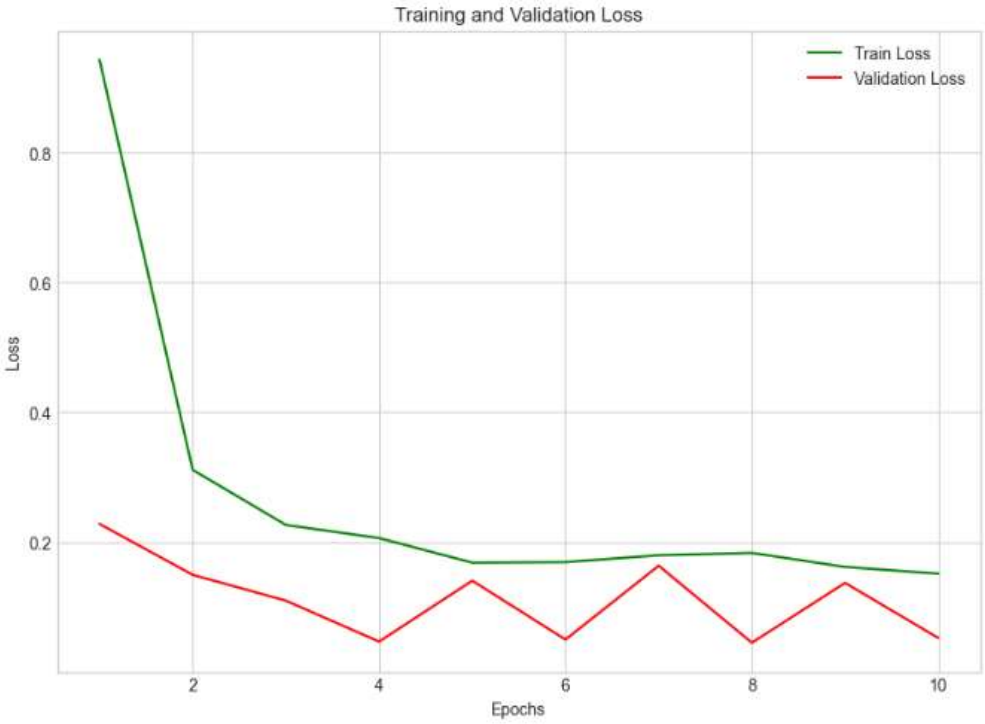


Figure 67: Courbe de Perte : Évolution de la perte sur l'entraînement et la validation.

Le modèle ResNet50 atteint une accuracy globale exceptionnelle de 99.07 %, ce qui signifie que près de 99 % des images de scanner ont été correctement classées en tissu normal ou en cancer.

- **Classe « normal » (tissu pulmonaire sain) :**

- Précision : 1.00 → Toutes les images prédites comme « normales » le sont effectivement, ce qui montre une précision parfaite pour cette classe.

- Rappel : 0.98 → 98 % des images réellement normales ont été correctement détectées. Une petite partie (2 %) a été confondue avec des cas de cancer.

- F1-score : 0.99 → Excellent équilibre entre précision et rappel.

- **Classe « cancer » (tissu cancéreux) :**

- Précision : 0.98 → 98 % des images identifiées comme cancéreuses sont effectivement atteintes.

- Rappel : 1.00 → Le modèle a détecté 100 % des cas de cancer, ce qui est un point fort crucial en contexte médical.

- F1-score : 0.99 → Indique une performance très fiable pour la détection du cancer à partir des CT scans.

- **Moyennes globales :**

- Macro average : 0.99 → Moyenne simple des scores des deux classes, illustrant un bon équilibre de performance quel que soit le type de tissu.
- Weighted average : 0.99 → Moyenne pondérée en fonction du nombre d'exemples dans chaque classe, confirmant la robustesse du modèle même en présence d'un léger déséquilibre.

Ces résultats sont consolidés par l'analyse de la matrice de confusion, qui précise le comportement du modèle face aux vrais cas :

- Vrais positifs (VP = 55) : le modèle a correctement identifié 55 cas de cancer comme étant cancéreux.
- Faux positifs (FP = 0) : aucun cas sain n'a été faussement classé comme cancéreux, ce qui reflète une spécificité parfaite.
- Vrais négatifs (VN = 54) : le modèle a correctement classé 54 images normales comme étant saines.
- Faux négatifs (FN = 1) : 1 cas de cancer a été mal classé comme normal, ce qui représente une erreur minime.

Enfin, l'interprétation des courbes d'apprentissage permet de mieux comprendre la dynamique d'entraînement et de validation du modèle au fil des époques.

La courbe de précision (Accuracy) montre une amélioration rapide durant les premières époques :

- Côté entraînement, la précision progresse rapidement de ~0.65 à environ 0.95, puis se stabilise autour de cette valeur.

- Côté validation, la précision suit une trajectoire similaire, atteignant parfois 1.00, mais oscillant globalement autour de 0.95, témoignant d'une bonne généralisation.

- L'écart modéré entre les deux courbes est normal, et l'absence de divergence notable confirme une stabilité satisfaisante.

Quant à la courbe de perte (Loss), elle reflète une convergence efficace du modèle :

- La perte sur l'entraînement diminue de façon marquée pour atteindre environ 0.2 dès les premières époques, puis se stabilise.

- La perte sur la validation suit une tendance similaire, atteignant un minimum proche de 0.1 avant de remonter légèrement autour de 0.2.

- Ces courbes indiquent que le modèle n'est pas affecté par un surapprentissage important, et qu'il maintient une performance cohérente sur des données non vues.

Globalement, l'évolution conjointe des courbes de précision et de perte confirme une convergence efficace du modèle et une excellente capacité de généralisation. À la fin de l'entraînement, la précision atteint 95 % sur les deux ensembles (entraînement et validation), et la perte reste faible (~0.2), ce qui témoigne de la robustesse et de la fiabilité du modèle ResNet50 pour la détection du cancer pulmonaire à partir des images de scanner thoracique.

1.3 Résultats du Modèle Multimodale

	precision	recall	f1-score	support
0	0.90	1.00	0.95	26
1	1.00	0.91	0.95	34
accuracy			0.95	60
macro avg	0.95	0.96	0.95	60
weighted avg	0.96	0.95	0.95	60

Figure 68 : Performance du modèle multimodale – Rapport de Classification (Jeu de test).

```
1/1 [==============================] - 0s 200ms/step
Ligne 18: Prédiction = 0, Vérité = 0 => ✔ (Proba = 0.06)
1/1 [==============================] - 0s 193ms/step
Ligne 19: Prédiction = 1, Vérité = 0 => ✘ (Proba = 0.62)
1/1 [==============================] - 0s 202ms/step
Ligne 20: Prédiction = 0, Vérité = 0 => ✔ (Proba = 0.04)
1/1 [==============================] - 0s 223ms/step
Ligne 21: Prédiction = 1, Vérité = 1 => ✔ (Proba = 0.97)
1/1 [==============================] - 0s 192ms/step
Ligne 22: Prédiction = 1, Vérité = 1 => ✔ (Proba = 0.97)
1/1 [==============================] - 0s 203ms/step
Ligne 23: Prédiction = 0, Vérité = 0 => ✔ (Proba = 0.08)
1/1 [==============================] - 0s 253ms/step
Ligne 24: Prédiction = 0, Vérité = 0 => ✔ (Proba = 0.03)
1/1 [==============================] - 0s 285ms/step
Ligne 25: Prédiction = 1, Vérité = 1 => ✔ (Proba = 0.97)

✅ Accuracy totale sur l'ensemble du dataset : 0.96
```

Figure 69 : Prédiction et leur probabilité sur un dataset de test.

Le modèle multimodal atteint une accuracy globale de 95 %, ce qui signifie que 95 % des prédictions sur l'ensemble de test ont été correctes. Cette performance confirme l'intérêt de la fusion des données cliniques et d'imagerie pour renforcer la précision du diagnostic.

Dans un premier temps, le rapport de classification met en évidence des résultats équilibrés entre les deux classes :

- **Classe 0 (patients non atteints de cancer)** :
  - Précision : 0.90 → 90 % des cas prédits comme « non atteints » sont effectivement sains.
  - Rappel : 1.00 → Tous les patients réellement non atteints ont été correctement détectés.
  - F1-score : 0.95 → Très bon équilibre entre précision et rappel.
- **Classe 1 (patients atteints de cancer)** :
  - Précision : 1.00 → Toutes les prédictions positives (cancer) étaient correctes.

- Rappel : 0.91 → 91 % des vrais cas de cancer ont été correctement identifiés, mais 9 % ont été manqués.
- F1-score : 0.95 → Excellent compromis global pour la détection de la classe positive.

- **Moyennes globales** :
  - Macro average : 0.95 → Moyenne simple des scores des deux classes, montrant un bon équilibre de performance.
  - Weighted average : 0.96 (précision), 0.95 (rappel & F1) → Moyennes pondérées en fonction du nombre d'échantillons dans chaque classe, soulignant la stabilité du modèle même en présence d'un déséquilibre modéré.

Dans un second temps, une évaluation individuelle sur un jeu de test jamais vu, composé de 26 échantillons, a permis d'observer la robustesse du modèle en conditions réelles. Sur l'ensemble de ces données, une seule erreur de prédiction a été relevée : à la ligne 19, le modèle a prédit la classe 1 (cancer) alors que la vérité était la classe 0 (normal). Cette erreur pourrait être attribuée à une incohérence entre les données cliniques (symptômes proches du cancer) et l'image, pourtant normale.

L'analyse des probabilités de prédiction vient compléter cette évaluation. Elle montre que :

- Pour les cas positifs (cancer), les probabilités prédites sont très élevées, souvent supérieures à 0.95, ce qui renforce la confiance dans ces décisions.
- Pour les cas négatifs (normal), les probabilités sont généralement inférieures à 0.10, ce qui traduit une bonne séparation des classes. Toutefois, certains cas affichent des probabilités intermédiaires autour de 0.60, traduisant un degré d'incertitude potentiel, souvent lié à la complexité des données cliniques.

Dans notre travail, nous avons développé un modèle multimodal combinant données cliniques et images CT-scan pour améliorer le diagnostic du cancer pulmonaire.
Nous avons comparé notre approche avec deux travaux récents qui se basent chacun sur une modalité unique.

• Liu et al. (2020)**,** dans l'article « Prognostic Prediction Models Based on Clinicopathological Indices in Patients With Resectable Lung Cancer », ont proposé un modèle basé uniquement sur les données cliniques [74]. Leur approche est pertinente pour l'analyse pronostique, mais elle ne prend pas en compte les informations visuelles issues de l'imagerie.

• Shatnawi et al. (2025)**,** dans « Deep learning-based approach to diagnose lung cancer using CT-scan images», se sont appuyés uniquement sur les images médicales (CT-scan) avec des

techniques de deep learning [75]. Ce modèle est performant sur le plan visuel, mais il ne tient pas compte des facteurs cliniques du patient.

Notre approche multimodale cherche à combiner ces deux sources complémentaires d'information pour fournir un diagnostic plus complet et potentiellement plus précis.

2 Modèle Ensembliste de diagnostic du CPNPC

2.1 Résultats du modèle ResNet50 sur les images histopathologiques

	precision	recall	f1-score	support
Lung Adenocarcinoma	0.98	1.00	0.99	500
Lung Benign Tissue	1.00	1.00	1.00	500
Lung Squamous Cell Carcinoma	1.00	0.98	0.99	500
accuracy			0.99	1500
macro avg	0.99	0.99	0.99	1500
weighted avg	0.99	0.99	0.99	1500

Figure 70 : Rapport de classification du modèle ResNet50 sur le jeu de test.

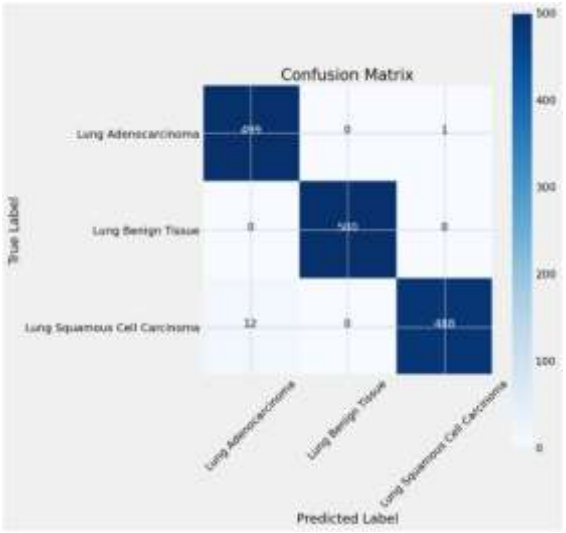


Figure 71 : Matrice de confusion du modèle ResNet50 sur le jeu de test.

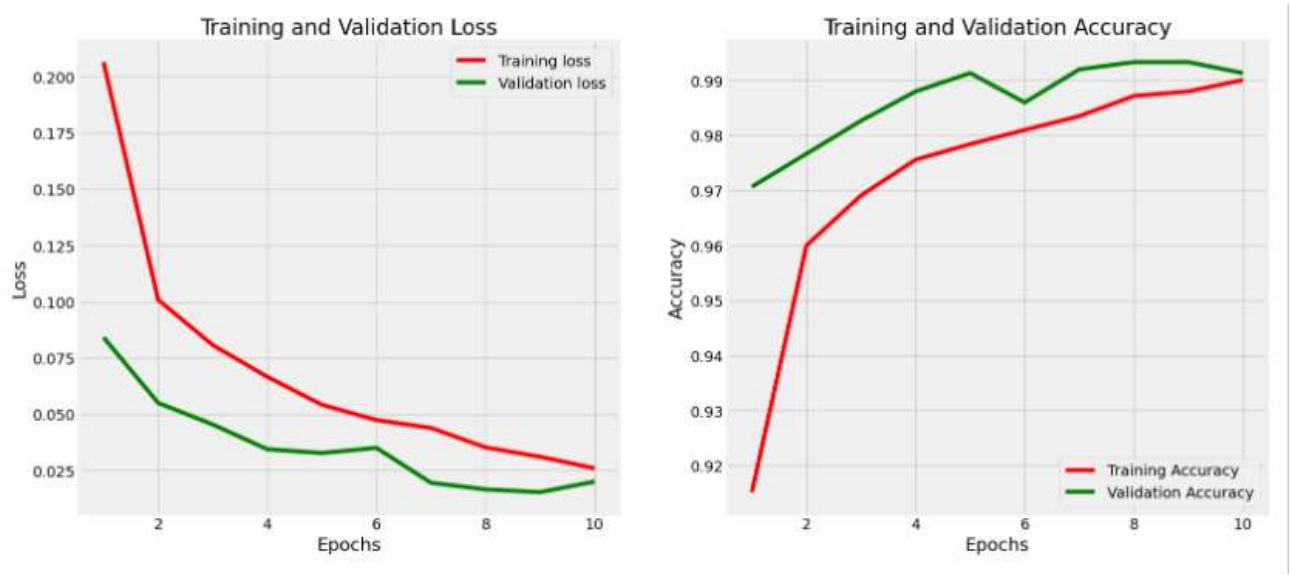


Figure 72 : Courbes d'évolution de la perte et de l'accuracy du modèle ResNet50 sur les données d'entraînement et de validation.

Le modèle ResNet50 atteint une **accuracy globale de 99 %** sur l'ensemble du test, ce qui signifie que 99 % des 1500 images histopathologiques ont été correctement classées selon leur type tissulaire. Ces performances témoignent d'une excellente capacité du modèle à distinguer les différents types de tissus pulmonaires, y compris les formes cancéreuses.

Dans un premier temps, le rapport de classification met en évidence des résultats très élevés sur l'ensemble des classes :

- **Classe : Lung Adenocarcinoma (Adénocarcinome pulmonaire)**
  - Précision : 0.98 → 98 % des images prédites comme « adénocarcinome » le sont effectivement.
  - Rappel : 1.00 → Tous les cas réels d'adénocarcinome ont été détectés sans erreur.
  - F1-score : 0.99 → Très bon équilibre entre précision et rappel, avec une performance quasi parfaite.
- **Classe : Lung Benign Tissue (Tissu pulmonaire bénin)**
  - Précision : 1.00 → Aucune image saine n'a été mal classée.
  - Rappel : 1.00 → Tous les tissus bénins ont été correctement identifiés.
  - F1-score : 1.00 → Performance parfaite sur cette classe, avec zéro erreur.

- **Classe : Lung Squamous Cell Carcinoma (Carcinome épidermoïde pulmonaire**)
  - Précision : 1.00 → Toutes les images classées comme carcinome épidermoïde étaient correctes.
  - Rappel : 0.98 → 98 % des vrais cas ont été détectés, 2 % ayant été classés à tort comme autre type.
  - F1-score : 0.99 → Très haut niveau de performance, avec un faible taux d'erreur.
- **Moyennes globales**
  - Macro average : 0.99 (précision, rappel, F1) → Moyenne simple des scores sur les trois classes, illustrant un bon équilibre global.
  - Weighted average : 0.99 → Moyenne pondérée tenant compte du nombre d'échantillons par classe, ce qui confirme la robustesse du modèle même en présence de légères variations de distribution.

Dans un second temps, les performances ont été confirmées par l'analyse de la matrice de confusion, qui révèle les détails des erreurs résiduelles :

- **Classe (Adénocarcinome pulmonaire)** :
  - Vrais positifs (VP) : 499 → Le modèle a correctement identifié 499 images comme étant des adénocarcinomes.
  - Faux négatifs (FN) : 1 → Une seule image d'adénocarcinome a été classée à tort comme carcinome épidermoïde.
  - Faux positifs (FP) : 12 → 12 images de carcinome épidermoïde ont été incorrectement classées comme adénocarcinome.
  - → Conclusion : Excellente reconnaissance, avec un très faible taux d'erreur.
- **Classe (Tissu pulmonaire bénin)** :
  - Vrais positifs (VP) : 500 → Toutes les images de tissu bénin ont été correctement classées.
  - Faux positifs / Faux négatifs : 0 → Aucun cas d'erreur.
  - → Conclusion : Spécificité et sensibilité parfaites pour cette classe, ce qui est remarquable.
- **Classe (Carcinome épidermoïde pulmonaire)** :
  - Vrais positifs (VP) : 488 → Le modèle a correctement classé 488 cas sur 500.
  - Faux négatifs (FN) : 12 → 12 images de carcinome épidermoïde ont été classées à tort comme adénocarcinome.

- Faux positifs (FP) : 1 → Une seule image d'adénocarcinome a été confondue avec cette classe.
- → Conclusion : Très bonne performance, mais une légère confusion avec l'adénocarcinome est observée, ce qui est compréhensible biologiquement du fait de similitudes visuelles entre ces deux types.

Enfin, les courbes d'apprentissage viennent compléter cette évaluation en illustrant la dynamique d'entraînement du modèle.

- **Perte (Loss) :**

- Perte d'entrainement (Train Loss) : Diminue de manière constante, passant de 0.2065 (Époque 1) à 0.0259 (Époque 10). Cela indique que le modèle apprend bien à minimiser l'erreur sur les données d'entraînement.
- Perte de validation (Validation Loss) : Diminue également, passant de 0.0841 à 0.0201, avec une légère fluctuation à l'Époque 6. La convergence des courbes de train et de validation suggère qu'il n'y a pas de surapprentissage (overfitting).

- **Précision (Accuracy) :**

- Précision d'entrainement (Train Accuracy): Augmente de 0.9152 à 0.9901, montrant une amélioration constante.
- Précision de validation (Validation Accuracy) : Passe de 0.9707 à 0.9933, avec des performances légèrement meilleures que sur les données d'entraînement à certaines époques. Cela confirme la robustesse du modèle.

Le modèle ResNet50 converge bien, avec une performance élevée et stable sur les données de validation. L'absence de surapprentissage est un point positif.

2.2 Résultats du modèle DenseNet121 sur les images histopathologiques

	precision	recall	f1-score	support
Lung Adenocarcinoma	0.95	0.86	0.91	500
Lung Benign Tissue	0.98	1.00	0.99	500
Lung Squamous Cell Carcinoma	0.89	0.96	0.92	500
accuracy			0.94	1500
macro avg	0.94	0.94	0.94	1500
weighted avg	0.94	0.94	0.94	1500

Figure 73 : Rapport de classification du modèle DenseNet121 sur le jeu de test.

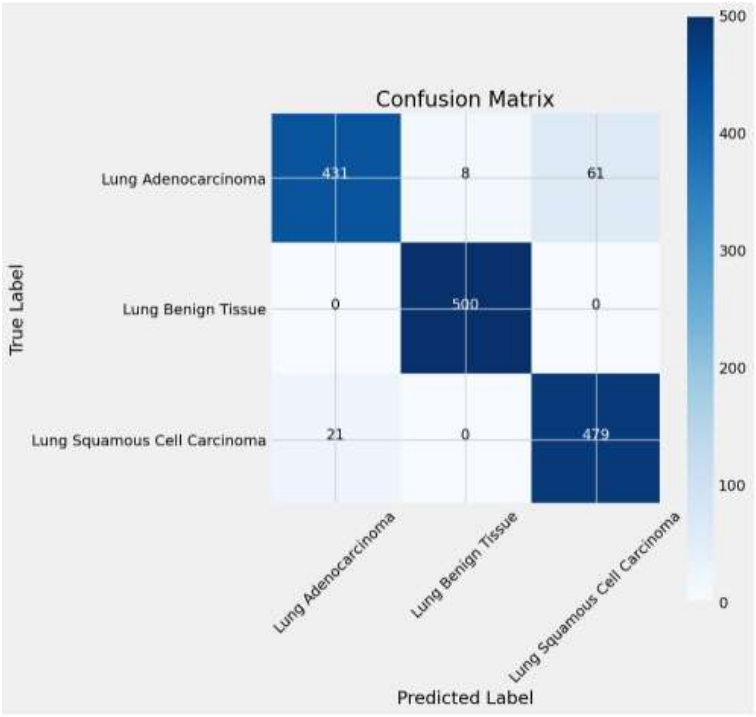


Figure 74 : Matrice de confusion du modèle DenseNet121 sur le jeu de test.

Figure 75 : Courbes d'évolution de la perte et de l'exactitude du modèle DenseNet121 sur les données d'entraînement et de validation.

Le modèle DenseNet121 atteint une accuracy globale de 94 % sur l'ensemble de test, ce qui signifie que 1 410 images sur 1 500 ont été correctement classées selon leur type histologique. Bien que légèrement inférieur à celui de ResNet50, ce résultat reste très satisfaisant et montre que le modèle est capable de distinguer avec une bonne fiabilité les différentes classes de tissus pulmonaires.

L’analyse du rapport de classification met en évidence des performances globalement élevées mais plus hétérogènes selon les classes :

- **Classe : Lung Adenocarcinoma (Adénocarcinome pulmonaire)**

- Précision : 0.95 → 95 % des images prédites comme adénocarcinomes étaient effectivement correctes.

- Rappel : 0.86 → Le modèle a correctement identifié 86 % des cas réels d’adénocarcinome, mais a manqué 14 % de ces cas.

- F1-score : 0.91 → Bon équilibre général, mais la baisse du rappel indique une sensibilité modérément réduite pour cette classe.

- **Classe : Lung Benign Tissue (Tissu pulmonaire bénin)**

- Précision : 0.98 → 98 % des images classées comme bénignes sont effectivement saines.

- Rappel : 1.00 → Tous les tissus bénins ont été correctement détectés, sans aucune erreur.

- F1-score : 0.99 → Excellente performance globale, avec un modèle parfaitement sensible pour cette classe.

- **Classe : Lung Squamous Cell Carcinoma (Carcinome épidermoïde pulmonaire)**

- Précision : 0.89 → 89 % des images prédites comme carcinome épidermoïde étaient correctes.

- Rappel : 0.96 → Le modèle a identifié 96 % des vrais cas de carcinome, ce qui montre une très bonne sensibilité.

- F1-score : 0.92 → Bonne performance générale, avec un léger compromis entre précision et rappel.

- **Moyennes globales**

- Macro average : 0.94 (précision, rappel, F1) → Moyenne simple des scores sur les trois classes, ce qui indique un bon équilibre de traitement.

- Weighted average : 0.94 → Moyenne pondérée par le nombre d’échantillons dans chaque classe, ce qui reflète une stabilité correcte du modèle sur l’ensemble du dataset.

En complément, la matrice de confusion permet d'analyser en détail les erreurs de prédiction :

- **Classe Adénocarcinome pulmonaire**

- Vrais positifs (VP) : 431 → 431 images d’adénocarcinome ont été correctement identifiées.
- Faux positifs (FP) : 21 → 21 images de carcinome épidermoïde ont été confondues avec cette classe.

- Faux négatifs (FN) : 69 (8+61) → 8 images ont été mal classées comme bénignes, et 61 comme carcinome épidermoïde.

→ Conclusion : Le modèle a montré une bonne précision, mais une sensibilité plus faible sur cette classe, avec une confusion importante vers le carcinome épidermoïde.

- **Classe Tissu pulmonaire bénin**

- Vrais positifs (VP) : 500 → Tous les cas bénins ont été correctement identifiés.
- Faux positifs / Faux négatifs : 0 → Aucune erreur de classification.

→ Conclusion : Le modèle présente une performance parfaite pour détecter les tissus sains, ce qui est essentiel pour éviter les faux diagnostics de cancer.

- **Classe Carcinome épidermoïde pulmonaire**

- Vrais positifs (VP) : 479 → 479 images de carcinome épidermoïde ont été correctement classées.
- Faux positifs (FP) : 61 → 61 images d'adénocarcinome ont été à tort classées comme carcinome épidermoïde.
- Faux négatifs (FN) : 21 → 21 images de carcinome épidermoïde ont été prises à tort pour de l'adénocarcinome.

→ Conclusion : Très bonne sensibilité sur cette classe, mais le modèle confond parfois cette forme avec l'adénocarcinome, ce qui peut être dû à la ressemblance visuelle entre certains profils histologiques.

Enfin, les courbes d'apprentissage apportent des indications supplémentaires sur le comportement du modèle pendant l'entraînement.

- **Perte d'entraînement (Train Loss) :**

- Diminue de manière rapide et constante au début, passant de 1,18 (Époque 1) à environ 0,20 (Époque 5), puis continue à diminuer plus lentement jusqu'à atteindre une valeur stable autour de 0,18–0,20 vers la fin de l'entraînement.
- Cette tendance indique que le modèle apprend bien à minimiser l'erreur sur les données d'entraînement.

- **Perte de validation (Perte de validation) :**

- Diminue également rapidement au début, passant de 0,31 (Époque 1) à environ 0,20 (Époque 5), puis stabilise autour de 0,15–0,18 vers la fin de l'entraînement.
- Il y a une légère fluctuation entre les époques 6 et 10, mais globalement, la convergence des courbes de train et de validation suggère qu'il n'y a pas de surapprentissage (overfitting). Les deux courbes restent proches, ce qui est un bon signe de généralisation.

- **Précision d'entraînement (Train Accuracy) :**
  - Augmentation de manière constante, passant de 0,7931 (Époque 1) à environ 0,9346 (Époque 19).
  - Cela montre une progression de la performance du modèle sur les données d'entraînement.
- **Précision de validation (Validation Accuracy) :**
  - Augmente également, passant de 0.8893 (Époque 1) à environ 0.9520 (Époque 19).
  - La précision de validation atteint même une valeur légèrement supérieure à celle de l'entraînement à certaines époques (par exemple, Époque 18 avec une précision de 0.9513), ce qui peut indiquer une bonne robustesse du modèle sur des données non vues.

Dans notre travail, nous avons développé un modèle multimodal combinant données cliniques et images CT-scan pour améliorer le diagnostic du cancer pulmonaire. Nous avons comparé notre approche avec deux travaux récents qui se basent chacun sur une modalité unique.

- Liu et al. (2020)**,** dans l'article « Prognostic Prediction Models Based on Clinicopathological Indices in Patients With Resectable Lung Cancer », ont proposé un modèle basé uniquement sur les données cliniques (âge, stade, statut tumoral, etc.). Leur approche est pertinente pour l'analyse pronostique, mais elle ne prend pas en compte les informations visuelles issues de l'imagerie.
- Shatnawi et al. (2025)**,** dans « Deep learning-based approach to diagnose lung cancer using CT-scan images», se sont appuyés uniquement sur les images médicales (CT-scan) avec des techniques de deep learning. Ce modèle est performant sur le plan visuel, mais il ne tient pas compte des facteurs cliniques du patient.

Notre approche multimodale cherche à combiner ces deux sources complémentaires d'information pour fournir un diagnostic plus complet et potentiellement plus précis.

2.3 Résultats du modèle ensembliste sur les images histopathologiques

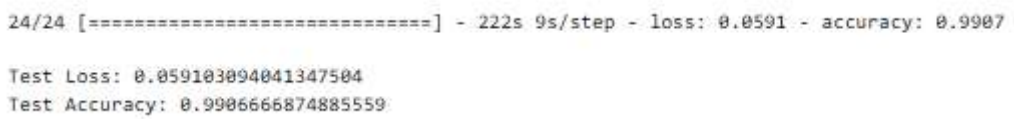


Figure 76 : Performances du modèle ensembliste.

Classification Report:

	precision	recall	f1-score	support
Lung Adenocarcinoma	0.99	0.98	0.99	500
Lung Benign Tissue	1.00	1.00	1.00	500
Lung Squamous Cell Carcinoma	0.99	0.99	0.99	500
accuracy			0.99	1500
macro avg	0.99	0.99	0.99	1500
weighted avg	0.99	0.99	0.99	1500

Figure 77 : Rapport de classification du modèle ensembliste.

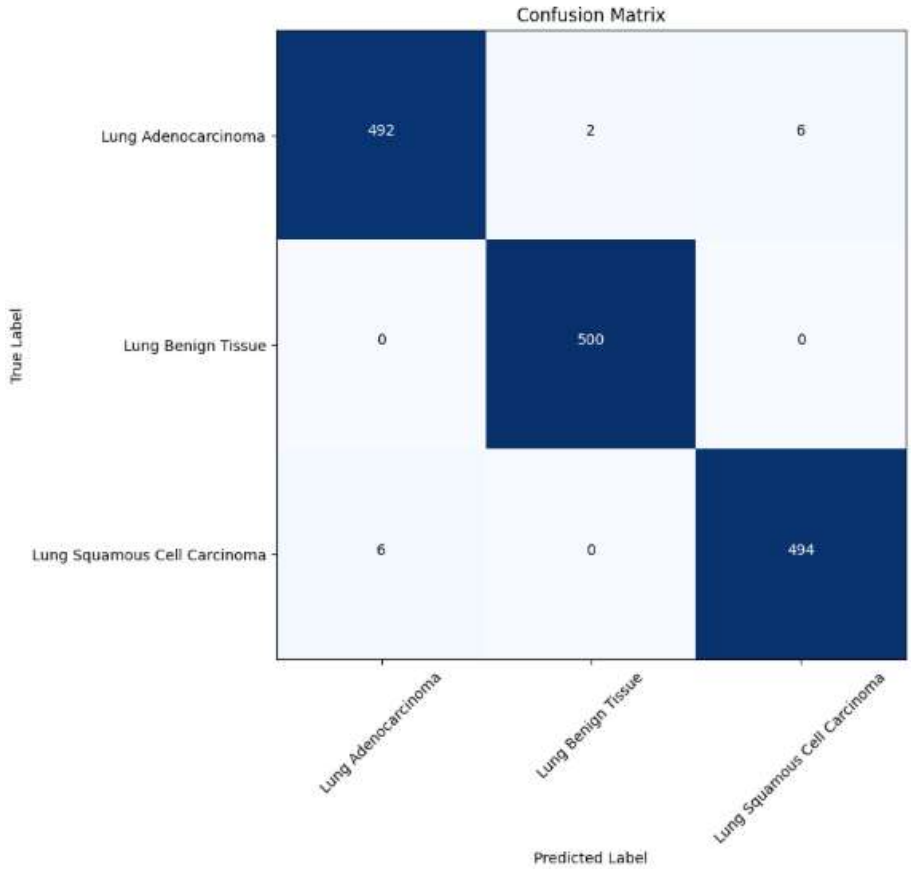


Figure 78 : Matrice de confusion du modèle ensembliste ResNet50 + DenseNet121.

Les performances du modèle ensembliste ont été évaluées sur le jeu de test (1 500 images), avec les résultats suivants :

- Accuracy globale : 99.07 %
- Loss : 0.0591

Le rapport de classification présente une excellente performance pour les trois classes :

- **Classe : Lung Adenocarcinoma (Adénocarcinome pulmonaire)**
  - Précision : 0.99 → 99 % des images classées comme adénocarcinomes l'étaient effectivement.
  - Rappel : 0.98 → 98 % des véritables cas d'adénocarcinome ont été correctement identifiés.
  - F1-score : 0.99 → Très bon équilibre entre précision et rappel, traduisant une excellente performance globale sur cette classe.
- **Classe : Lung Benign Tissue (Tissu pulmonaire bénin)**

- Précision : 1.00 → Aucun tissu sain n'a été mal classé.
- Rappel : 1.00 → Tous les tissus bénins ont été parfaitement détectés.
- F1-score : 1.00 → Le modèle affiche ici une performance parfaite, ce qui est essentiel en contexte clinique pour éviter les faux diagnostics de cancer.

- **Classe : Lung Squamous Cell Carcinoma (Carcinome épidermoïde pulmonaire)**

- Précision : 0.99 → 99 % des images prédites comme carcinome épidermoïde étaient correctes.
- Rappel : 0.99 → 99 % des vrais cas de cette classe ont été identifiés.
- F1-score : 0.99 → Performance équilibrée et très élevée, avec très peu d'erreurs de classification.

- **Moyennes globales**

- Macro average : 0.99 → Moyenne simple des trois classes, démontrant une excellente régularité du modèle.
- Weighted average : 0.99 → Moyenne pondérée selon la taille des classes, confirmant une performance très stable sur l'ensemble des données.

La matrice de confusion confirme ces résultats avec un total de 14 erreurs seulement sur 1500 prédictions :

- **Classe Adénocarcinome pulmonaire**

- Vrais positifs (VP) : 492 → Le modèle a correctement identifié 492 cas d'adénocarcinome.
- Faux négatifs (FN) : 8 (2 + 6) → 2 images ont été confondues avec le tissu bénin, et 6 avec le carcinome épidermoïde.
- Faux positifs (FP) : 6 → 6 cas de carcinome épidermoïde ont été à tort prédits comme adénocarcinome.

→ Conclusion : Très bonne performance avec un faible taux de confusion, principalement avec le carcinome épidermoïde, ce qui est compréhensible biologiquement.

- **Classe Tissu pulmonaire bénin**

- Vrais positifs (VP) : 500 → Tous les cas de tissu bénin ont été correctement classés.
- Faux positifs / Faux négatifs : 0 → Aucune erreur sur cette classe.

→ Conclusion : Le modèle atteint une précision et une sensibilité parfaites sur les tissus sains, ce qui est crucial pour éviter les diagnostics erronés.

- **Classe Carcinome épidermoïde pulmonaire**

- Vrais positifs (VP) : 494 → Le modèle a correctement identifié 494 cas.
- Faux négatifs (FN) : 6 → 6 images de cette classe ont été prises à tort pour de l'adénocarcinome.
- Faux positifs (FP) : 6 → 6 adénocarcinomes ont été prédits à tort comme épidermoïdes.

→ Conclusion : Le modèle montre une très bonne sensibilité, avec une confusion croisée limitée avec l'adénocarcinome, ce qui reste faible compte tenu de la complexité visuelle entre ces deux types de cancer.

Dans le cadre de la classification histopathologique du cancer pulmonaire non à petites cellules (CPNPC), notre étude met en avant l'efficacité d'une approche ensembliste simple et robuste, reposant sur la combinaison des architectures ResNet50 et DenseNet121. Cette stratégie, fondée sur une moyenne des probabilités prédictives, présente des avantages majeurs en termes de simplicité, de performances, et de facilité d'intégration clinique.

L'évaluation comparative de nos trois modèles – ResNet50, DenseNet121, et leur combinaison – révèle une progression nette des performances : ResNet50 atteint une accuracy de 99 %, DenseNet121 de 94 %, et le modèle ensembliste culmine à 99.07 %, avec une stabilité remarquable sur les trois classes (tissus normaux, adénocarcinome, carcinome épidermoïde). Cette combinaison permet de corriger les biais individuels de chaque modèle, notamment les erreurs de confusion fréquentes entre adénocarcinome et carcinome épidermoïde, tout en maintenant une précision parfaite sur les tissus bénins. Cela en fait une solution particulièrement adaptée aux environnements hospitaliers, où la fiabilité des diagnostics est cruciale.

Cette approche se distingue par sa faible complexité algorithmique comparée à d'autres travaux récents. Par exemple, Alotaibi et al. (2024) ont proposé une méthode multimodale avancée (HIELCC-EDL), intégrant CA-ResNet50, ELM, CNN, LSTM et optimisation métaheuristique (Tuna Swarm Optimization) [76]. Leur architecture atteint une accuracy légèrement supérieure (99.60 %), mais au prix d'une complexité élevée, d'un coût computationnel important, et d'une difficulté d'implémentation clinique, notamment en raison du besoin de réglages poussés et d'une infrastructure spécialisée.

De manière similaire, Talukder et al. (2022) ont opté pour une approche hybride combinant deep features extraits d'un ResNet et des classifieurs traditionnels (SVM, Random Forest), atteignant une accuracy de 99.05 % [77]. Toutefois, leur pipeline implique une étape de transformation intermédiaire et une gestion manuelle des features, compromettant ainsi l'avantage du modèle end-to-end que permet notre méthode.

En comparaison, notre modèle ensembliste :

- Utilise des réseaux éprouvés, bien documentés et accessibles.
- S'appuie sur une fusion simple, rapide à implémenter et explicable.
- Offre des résultats compétitifs, avec seulement 14 erreurs sur 1500 images, démontrant sa robustesse.

Ces performances suggèrent un fort potentiel clinique, notamment pour une intégration dans une plateforme d'aide au diagnostic comme DiagnoLung. Une telle solution pourrait assister efficacement les pathologistes dans l'analyse des biopsies pulmonaires, en combinant précision, rapidité et transparence du processus décisionnel.

3 Notre plateforme : DiagnoLung

DiagnoLung est une plateforme dédiée au diagnostic du cancer du poumon en général, et du cancer pulmonaire non à petites cellules (CPNPC) en particulier. Elle a été conçue pour aider les médecins (les oncologues, les anatomo-pathologistes etc,.) dans le diagnostic du CPNPC. La plateforme permet également le stockage et l'organisation structurée des dossiers médicaux des patients, facilitant ainsi leur consultation à tout moment et améliorant le suivi médical. Ci-dessous, une capture d'écran illustrant la structure générale de DiagnoLung.

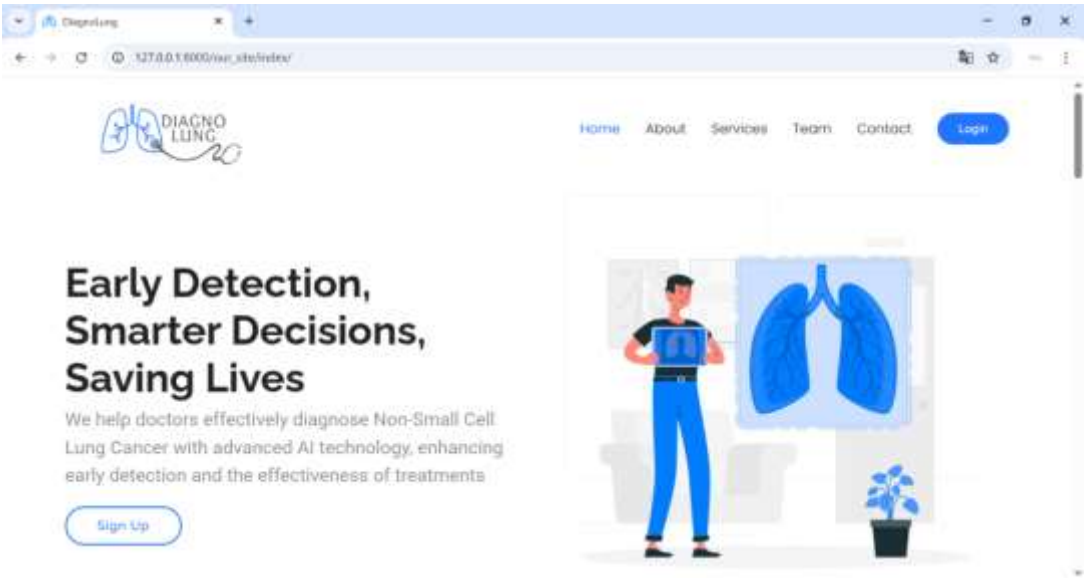


Figure 79: Structure Générale de DiagnoLung.

3.1 Création de compte des médecins

Cette page permet aux nouveaux médecins de s'inscrire sur la plateforme.

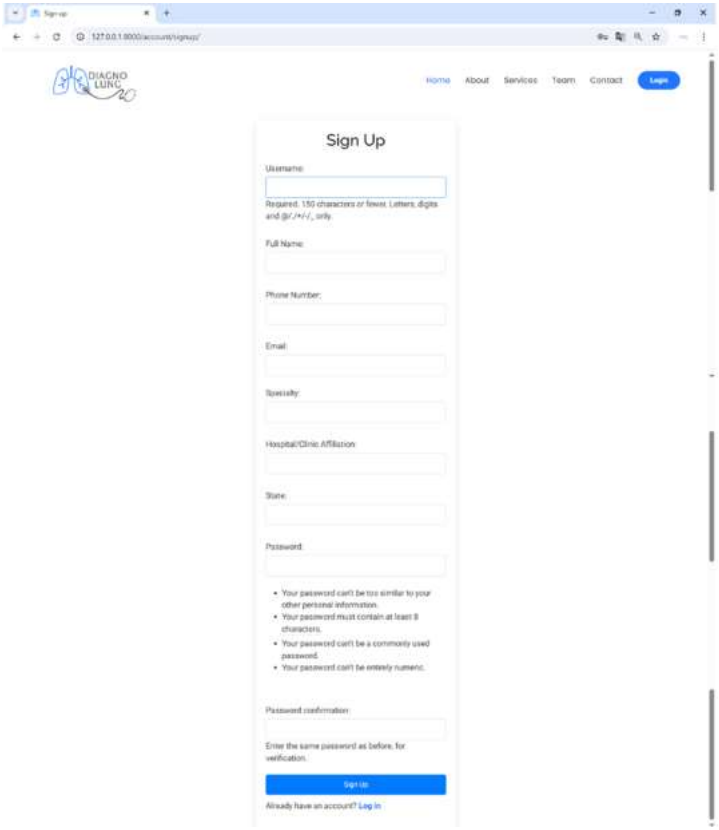


Figure 80 : Page d'inscription des médecins.

3.2 Connexion

Cette page permet aux médecins de se connecter à leur compte (figure).

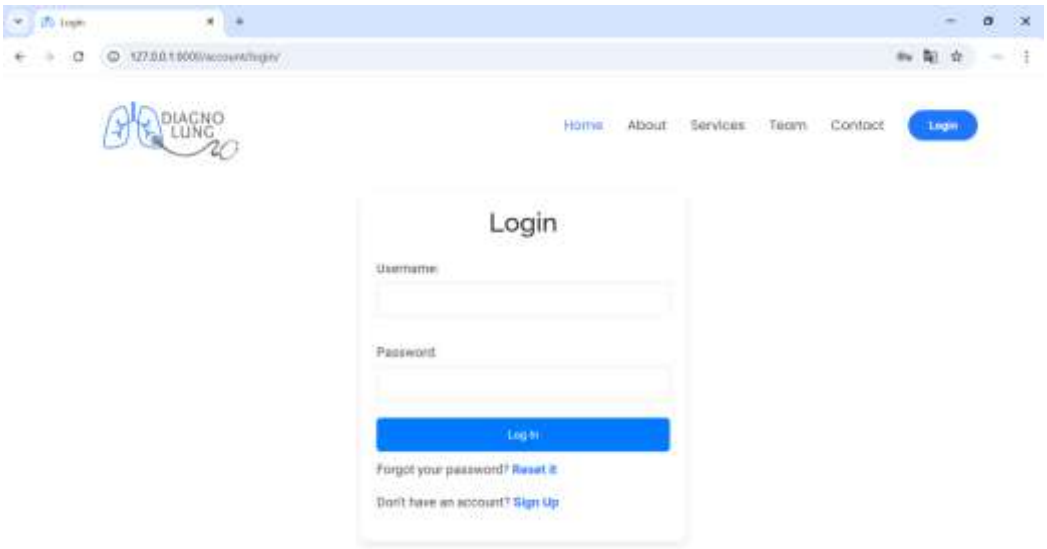


Figure 81 : Page de connexion des médecins.

3.3 Interface de diagnostic

3.3.1 Diagnostic du cancer du poumon

Ce modèle multimodal combine les données cliniques et les images médicales pour évaluer la probabilité de présence d'un cancer du poumon.

- **Interface des données cliniques**

Formulaire permettant au médecin de renseigner les données cliniques du patient.

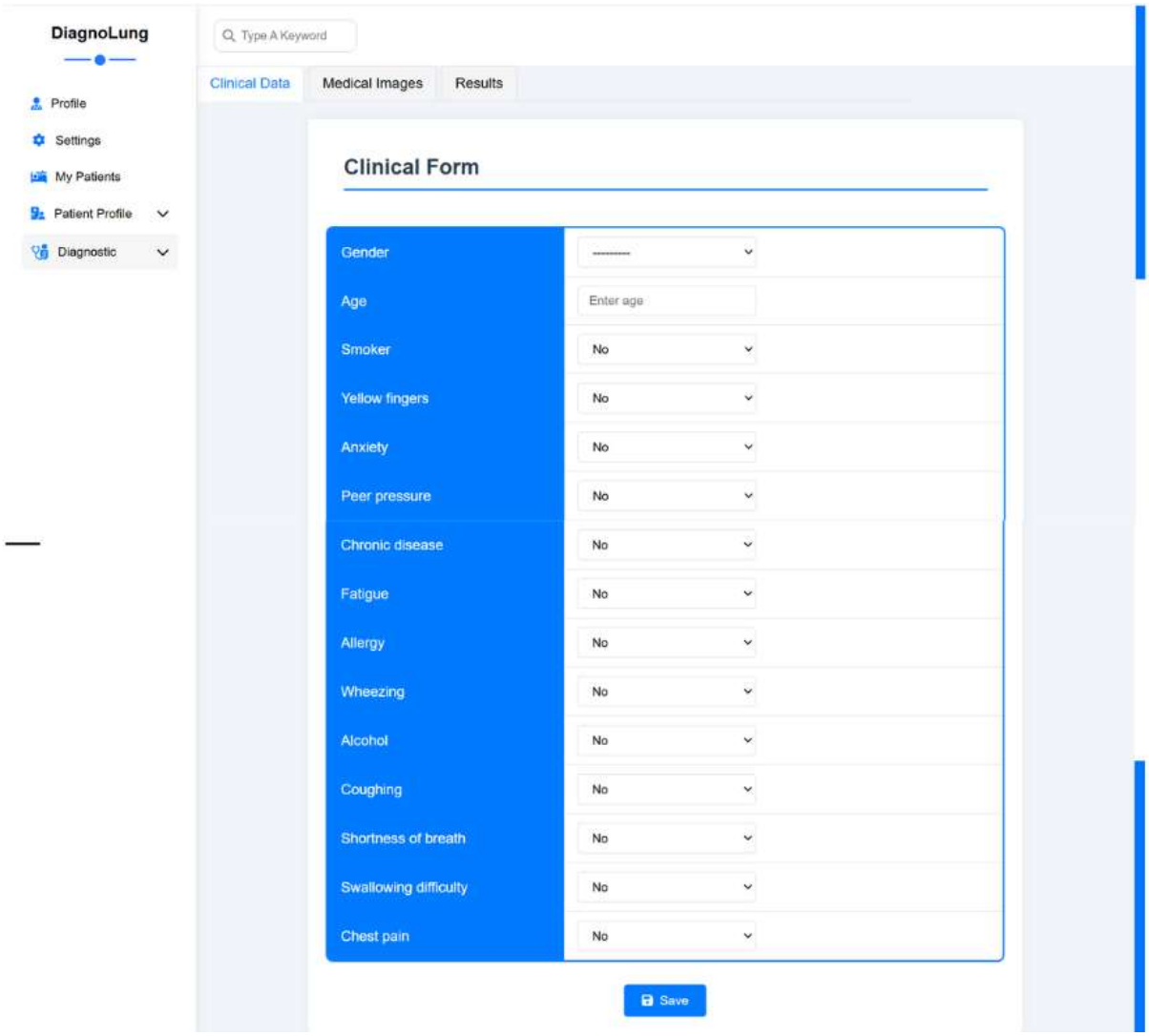


Figure 82: Formulaire des données cliniques.

- **Interface d'imagerie thoracique (CT-Scan)**

Interface permettant au médecin de télécharger l'image CT-Scan du patient.

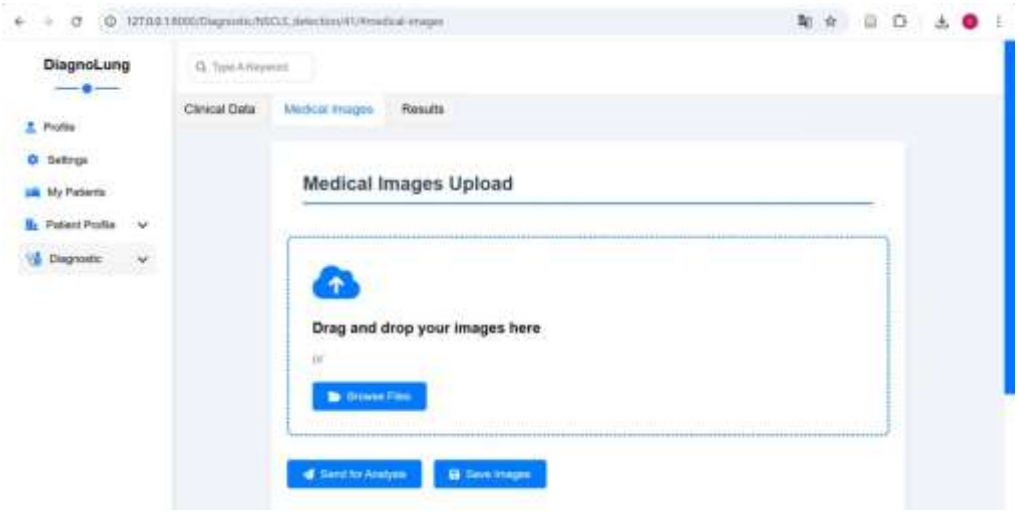


Figure 83: Image Médicale (CT Scan).

- **Interface des résultats**

Affichage du résultat de prédiction indiquant la présence ou l'absence d'un cancer du poumon.

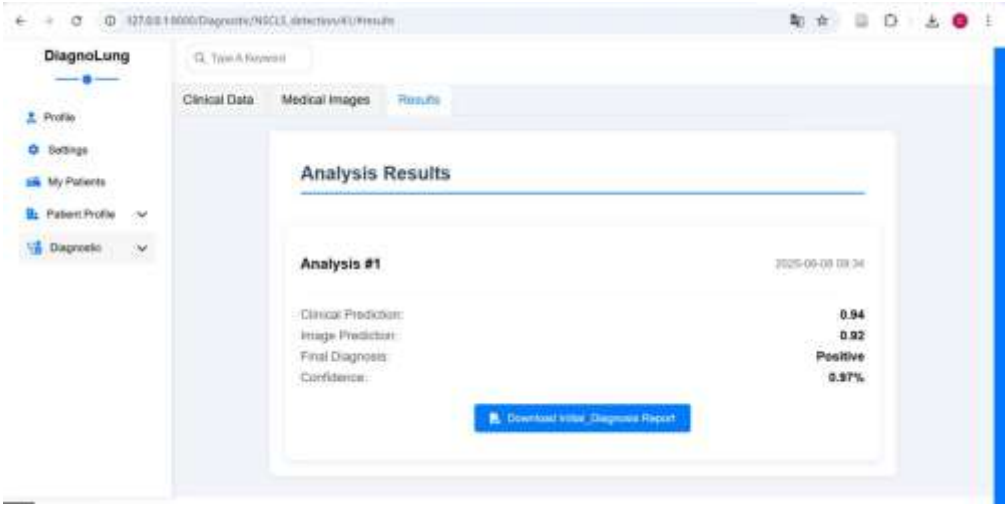


Figure 84: Page de résultat de prédiction du cancer pulmonaire.

Si le résultat est négatif, le patient ne présente pas de signes évocateurs de cancer du poumon. En revanche, si le résultat est positif, une suspicion de cancer est émise ; une biopsie pulmonaire est alors recommandée. Les lames issues de cette biopsie sont ensuite analysées par le second modèle dédié à la détection du CPNPC.

3.3.2 Diagnostic du CPNPC

Ce second modèle repose sur l'analyse d'images histopathologiques obtenues à partir de biopsies pulmonaires.

- **Interface d'image histopathologique**

Interface permettant au médecin de télécharger une image histopathologique du patient.

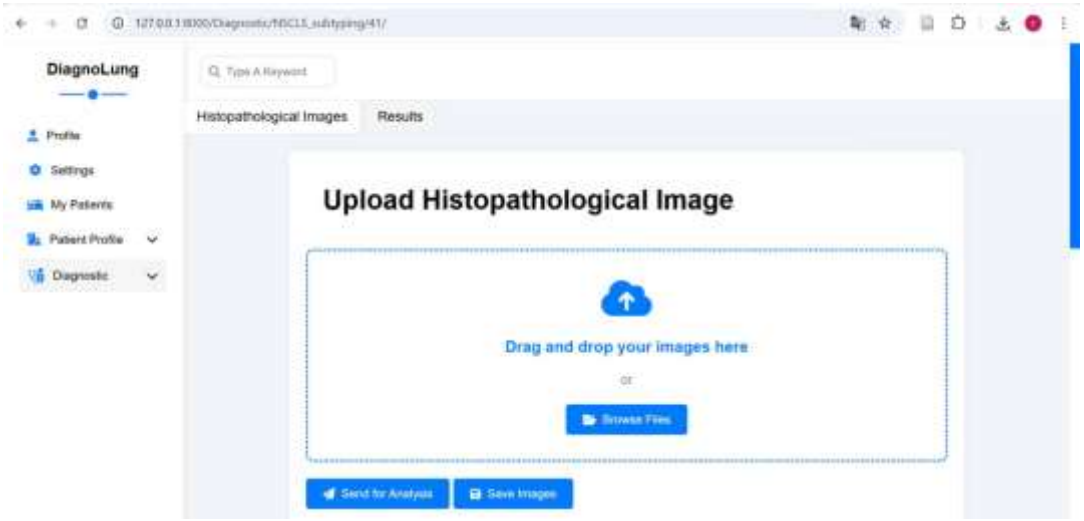


Figure 85: Image Histopathologique.

- **Interface des résultats**

Affichage du résultat de prédiction indiquant la présence ou non d'un CPNPC, ainsi que le type histologique détecté (adénocarcinome ou carcinome épidermoïde).

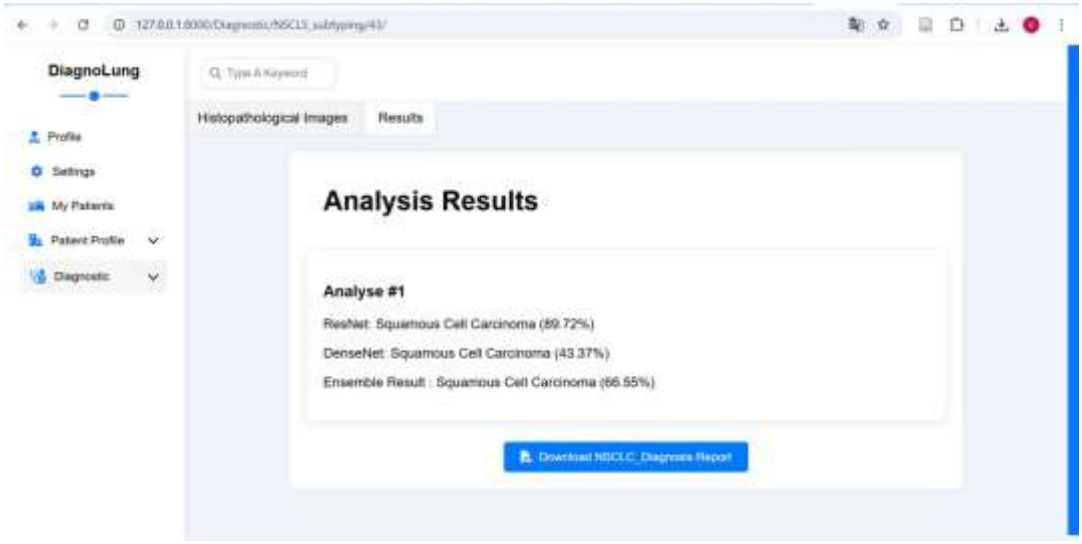


Figure 86 : Page de résultat de prédiction du CPNPC.

Conclusion

Conclusion

Le projet a débuté par la conception de modèles d'intelligence artificielle pour aider au diagnostic précoce du cancer pulmonaire. Nous avons développé un modèle multimodal combinant données cliniques (Random Forest) et images CT-Scan (ResNet50), puis un modèle ensembliste (ResNet50 + DenseNet121) pour le sous-typage du CPNPC. Ces modèles ont été intégrés dans une plateforme interactive nommée DiagnoLung , facilitant leur utilisation par les médecins.

Les résultats obtenus sont très encourageants, avec une précision de 95 % pour le modèle multimodal et 99,07 % pour le modèle ensembliste, démontrant l'efficacité de nos approches. DiagnoLung offre une interface conviviale permettant de charger des données et d'obtenir des prédictions rapides et interprétables.

Ce travail s'inscrit dans un domaine stratégique à la croisée de l'oncologie, de la bio-informatique et de l'intelligence artificielle, en proposant des outils innovants pour améliorer le dépistage et la classification des cancers pulmonaires.

De nombreuses perspectives sont envisageables : l'extension aux autres types de cancers, l'ajout de nouvelles architectures, l'amélioration de l'interprétabilité des résultats, ainsi que le déploiement de la plateforme en milieu clinique pour une validation réelle.

Références

Références

[1] « Lung cancer ». Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://www.who.int/news-room/fact-sheets/detail/lung-cancer

[2] « Lung Cancer Survival Rates: By Stage, Age, Type, and More ». Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://www.healthline.com/health/lung-cancer-stages-survival-rates?utm_source=chatgpt.com

[3] J. G. Elmore *et al.*, « Diagnostic Concordance Among Pathologists Interpreting Breast Biopsy Specimens », *JAMA*, vol. 313, nº 11, p. 1122, mars 2015, doi: 10.1001/jama.2015.1405.

[4] « Les poumons - Cancer du poumon ». Consulté le: 28 janvier 2025. [En ligne]. Disponible sur: https://www.e-cancer.fr/Patients-et-proches/Les-cancers/Cancer-du-poumon/Les-poumons

[5] « Cancer du poumon ». Consulté le: 28 janvier 2025. [En ligne]. Disponible sur: https://www.who.int/fr/news-room/fact-sheets/detail/lung-cancer

[6] « Les_traitements_des_cancers_du_poumon_mel_20171222 ».

[7] A. N. Wilkinson, « Cours accéléré sur le cancer du poumon », *Canadian Family Physician*, vol. 69, nº 4, p. e74-e77, 2023.

[8] « Développement du cancer du poumon - Cancer du poumon ». Consulté le: 2 février 2025. [En ligne]. Disponible sur: https://www.e-cancer.fr/Patients-et-proches/Les-cancers/Cancer-du-poumon/Developpement-du-cancer

[9] R. Hartmann, « Espérance de vie & stades des cancers du poumon », Institut de Radiothérapie et de Radiochirurgie H. Hartmann | SENY. Consulté le: 5 février 2025. [En ligne]. Disponible sur: https://radiotherapie-hartmann.fr/actualites/cancer-poumon/les-stades-des-cancers-du-poumon-et-lesperance-de-vie/

[10] Y. Xu *et al.*, « Artificial intelligence: A powerful paradigm for scientific research », *The Innovation*, vol. 2, nº 4, Art. nº 4, nov. 2021, doi: 10.1016/j.xinn.2021.100179.

[11] G. Briganti, « Intelligence artificielle : une introduction pour les cliniciens », *Revue des Maladies Respiratoires*, vol. 40, nº 4, Art. nº 4, avr. 2023, doi: 10.1016/j.rmr.2023.02.005.

[12] N. Sabouk et M. L. Sidmou, « L'INTELLIGENCE ARTIFICIELLE ; VERS UN NOUVEAU PARADIGME INTERDISCIPLINAIRE : ETAT DE SYNTHESE », *Revue Internationale du Marketing et Management Stratégique*, vol. 1, nº 4, Art. nº 4, 2019, Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://revue-rimms.org/index.php/home/article/view/90

[13] C. Janiesch, P. Zschech, et K. Heinrich, « Machine learning and deep learning », *Electron Markets*, vol. 31, nº 3, Art. nº 3, sept. 2021, doi: 10.1007/s12525-021-00475-2.

[14] J. Kufel *et al.*, « What Is Machine Learning, Artificial Neural Networks and Deep Learning?—Examples of Practical Applications in Medicine », *Diagnostics (Basel)*, vol. 13, nº 15, p. 2582, août 2023, doi: 10.3390/diagnostics13152582.

[15] G. Saint-Cirgue, « Apprendre le Machine Learning en une semaine ». Consulté le: 3 février 2025. [En ligne]. Disponible sur: https://www.machinelearnia.com/apprendre-le-machine-learning-en-une-semaine

[16] L. Adlung, Y. Cohen, U. Mor, et E. Elinav, « Machine learning in clinical decision making », *Med*, vol. 2, nº 6, p. 642-665, juin 2021, doi: 10.1016/j.medj.2021.04.006.

[17] A. Garg et V. Mago, « Role of machine learning in medical research: A survey », *Computer Science Review*, vol. 40, p. 100370, mai 2021, doi: 10.1016/j.cosrev.2021.100370.

[18] Z. ISMAILI, « Apprentissage Supervisé Vs. Non Supervisé », BrightCape. Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://brightcape.co/apprentissage-supervise-vs-non-supervise/

[19] H. Belyadi et A. Haghighat, « Chapter 5 - Supervised learning », in *Machine Learning Guide for Oil and Gas Using Python*, H. Belyadi et A. Haghighat, Éd., Gulf Professional Publishing, 2021, p. 169-295. doi: 10.1016/B978-0-12-821929-4.00004-4.

[20] « Machine Learning », Google for Developers. Consulté le: 26 avril 2025. [En ligne]. Disponible sur: https://developers.google.com/machine-learning/decision-forests/random-forests?hl=fr

[21] « Random Forest : comment ça fonctionne ? », Formation Tech et Data en ligne | Blent.ai. Consulté le: 26 avril 2025. [En ligne]. Disponible sur: https://blent.ai/blog/a/random-forest-comment-ca-marche
[22] I. Antonopoulos *et al.*, « Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review », *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109899, sept. 2020, doi: 10.1016/j.rser.2020.109899.
[23] K. Karađuzović-Hadžiabdić et A. Peters, « Chapter 15 - Artificial intelligence in clinical decision-making for diagnosis of cardiovascular disease using epigenetics mechanisms », in *Epigenetics in Cardiovascular Disease*, vol. 24, Y. Devaux et E. L. Robinson, Éd., in Translational Epigenetics, vol. 24. , Academic Press, 2021, p. 327-345. doi: 10.1016/B978-0-12-822258-4.00020-1.
[24] D. Karimi, H. Dou, S. K. Warfield, et A. Gholipour, « Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis », *Medical Image Analysis*, vol. 65, p. 101759, oct. 2020, doi: 10.1016/j.media.2020.101759.
[25] « Réseaux de neurones artificiels : aperçu | ScienceDirect Topics ». Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/artificial-neural-network?utm_source=chatgpt.com
[26] R. Yamashita, M. Nishio, R. K. G. Do, et K. Togashi, « Convolutional neural networks: an overview and application in radiology », *Insights Imaging*, vol. 9, nº 4, p. 611-629, août 2018, doi: 10.1007/s13244-018-0639-9.
[27] Q. Zhang, L. T. Yang, Z. Chen, et P. Li, « A survey on deep learning for big data », *Information Fusion*, vol. 42, p. 146-157, juill. 2018, doi: 10.1016/j.inffus.2017.10.006.
[28] A. D. Torres, H. Yan, A. H. Aboutalebi, A. Das, L. Duan, et P. Rad, « Patient Facial Emotion Recognition and Sentiment Analysis Using Secure Cloud With Hardware Acceleration », in *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, Elsevier, 2018, p. 61-89. doi: 10.1016/B978-0-12-813314-9.00003-7.
[29] Q. Zhang, L. T. Yang, Z. Chen, et P. Li, « A survey on deep learning for big data », *Information Fusion*, vol. 42, p. 146-157, juill. 2018, doi: 10.1016/j.inffus.2017.10.006.
[30] I. Pacal, D. Karaboga, A. Basturk, B. Akay, et U. Nalbantoglu, « A comprehensive review of deep learning in colon cancer », *Computers in Biology and Medicine*, vol. 126, p. 104003, nov. 2020, doi: 10.1016/j.compbiomed.2020.104003.
[31] S. Chen, C. He, Y. Li, Z. Li, et C. E. Melançon, « A computational toolset for rapid identification of SARS-CoV-2, other viruses and microorganisms from sequencing data », *Briefings in Bioinformatics*, vol. 22, nº 2, p. 924-935, mars 2021, doi: 10.1093/bib/bbaa231.
[32] « Couche convolutive : aperçu | ScienceDirect Topics ». Consulté le: 6 mai 2025. [En ligne]. Disponible sur: https://www.sciencedirect.com/topics/computer-science/convolutional-layer
[33] L. A. Bugnon, E. Fenoy, A. A. Edera, J. Raad, G. Stegmayer, et D. H. Milone, « Transfer learning: The key to functionally annotate the protein universe », *Patterns*, vol. 4, nº 2, p. 100691, févr. 2023, doi: 10.1016/j.patter.2023.100691.
[34] J. Praveen Gujjar, H. R. Prasanna Kumar, et N. N. Chiplunkar, « Image classification and prediction using transfer learning in colab notebook », *Global Transitions Proceedings*, vol. 2, nº 2, p. 382-385, nov. 2021, doi: 10.1016/j.gltp.2021.08.068.
[35] E. Theodorou, E. Spiliotis, et V. Assimakopoulos, « Optimizing inventory control through a data-driven and model-independent framework », *EURO Journal on Transportation and Logistics*, vol. 12, p. 100103, janv. 2023, doi: 10.1016/j.ejtl.2022.100103.
[36] J. Robert, « Transfer Learning : Qu'est-ce que c'est ? », DataScientest. Consulté le: 27 avril 2025. [En ligne]. Disponible sur: https://datascientest.com/transfer-learning
[37] K. He, X. Zhang, S. Ren, et J. Sun, « Deep Residual Learning for Image Recognition », in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, juin 2016, p. 770-778. doi: 10.1109/CVPR.2016.90.

[38] K. Sahinbas et F. O. Catak, « Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images », *Data Science for COVID-19*, p. 451-466, 2021, doi: 10.1016/B978-0-12-824536-1.00003-4.
[39] « Fig. 1 A schematic view of ResNet architecture [15], decomposed into... », ResearchGate. Consulté le: 7 mai 2025. [En ligne]. Disponible sur: https://www.researchgate.net/figure/A-schematic-view-of-ResNet-architecture-15-decomposed-into-three-blocks-embedding_fig1_333475917
[40] « ResNet (Residual Networks) expliqué | Ultralytics ». Consulté le: 6 mai 2025. [En ligne]. Disponible sur: https://www.ultralytics.com/fr/glossary/residual-networks-resnet
[41] P. Aggarwal, N. K. Mishra, B. Fatimah, P. Singh, A. Gupta, et S. D. Joshi, « COVID-19 image classification using deep learning: Advances, challenges and opportunities », *Computers in Biology and Medicine*, vol. 144, p. 105350, mai 2022, doi: 10.1016/j.compbiomed.2022.105350.
[42] M. A. Morid, A. Borjali, et G. Del Fiol, « A scoping review of transfer learning research on medical image analysis using ImageNet », *Computers in Biology and Medicine*, vol. 128, p. 104115, janv. 2021, doi: 10.1016/j.compbiomed.2020.104115.
[43] « Densenet | PDF | Chiens | Races », Scribd. Consulté le: 7 mai 2025. [En ligne]. Disponible sur: https://fr.scribd.com/document/695197591/Densenet
[44] A. Sarkar, « Creating DenseNet 121 with TensorFlow », TDS Archive. Consulté le: 7 mai 2025. [En ligne]. Disponible sur: https://medium.com/data-science/creating-densenet-121-with-tensorflow-edbc08a956d8
[45] A. Jean, « Une brève introduction à l'intelligence artificielle », *Med Sci (Paris)*, vol. 36, nº 11, p. 1059-1067, nov. 2020, doi: 10.1051/medsci/2020189.
[46] l'équipe C. médiforce, « Quelles sont les applications de l'IA en médecine ? », CMV médiforce. Consulté le: 24 avril 2025. [En ligne]. Disponible sur: https://www.cmvmediforce.fr/publications/par-themes/technologie/quelles-sont-les-applications-de-lia-en-medecine/
[47] S. Aradhya *et al.*, « Applications of artificial intelligence in clinical laboratory genomics », *American J of Med Genetics Pt C*, vol. 193, nº 3, p. e32057, sept. 2023, doi: 10.1002/ajmg.c.32057.
[48] T. C. Reis, « Deep learning in oncology: Transforming cancer diagnosis, prognosis, and treatment », *Emerging Trends in Drugs, Addictions, and Health*, vol. 5, p. 100171, déc. 2025, doi: 10.1016/j.etdah.2025.100171.
[49] D. S.-L. Martinez *et al.*, « Artificial intelligence opportunities in cardio-oncology: Overview with spotlight on electrocardiography », *American Heart Journal Plus: Cardiology Research and Practice*, vol. 15, p. 100129, mars 2022, doi: 10.1016/j.ahjo.2022.100129.
[50] M. Khalifa et M. Albadawy, « Artificial Intelligence for Clinical Prediction: Exploring Key Domains and Essential Functions », *Computer Methods and Programs in Biomedicine Update*, vol. 5, p. 100148, 2024, doi: 10.1016/j.cmpbup.2024.100148.
[51] L. Feng *et al.*, « Development and validation of a radiopathomics model to predict pathological complete response to neoadjuvant chemoradiotherapy in locally advanced rectal cancer: a multicentre observational study », *The Lancet Digital Health*, vol. 4, nº 1, p. e8-e17, janv. 2022, doi: 10.1016/S2589-7500(21)00215-6.
[52] I. Jurisica, « Explainable biology for improved therapies in precision medicine: AI is not enough », *Best Practice & Research Clinical Rheumatology*, vol. 38, nº 4, p. 102006, déc. 2024, doi: 10.1016/j.berh.2024.102006.
[53] « Top 10 des modèles d'IA et de soins de santé pour garantir des solutions médicales réactives ». Consulté le: 9 mai 2025. [En ligne]. Disponible sur: https://www.slideteam.net/blog/top-10-des-modeles-dia-et-de-soins-de-sante-pour-garantir-des-solutions-medicales-reactives?lang=French

[54] M. Tahir *et al.*, « A Comprehensive AI-Based Approach in Classifying Breast Lesions: Focusing on Improving Pathologists' Accuracy and Efficiency », *Clinical Breast Cancer*, p. S1526820925000837, mars 2025, doi: 10.1016/j.clbc.2025.03.016.
[55] S. Lee, J.-Y. Jung, A. Mahatthanatrakul, et J.-S. Kim, « Artificial Intelligence in Spinal Imaging and Patient Care: A Review of Recent Advances », *Neurospine*, vol. 21, nº 2, p. 474-486, juin 2024, doi: 10.14245/ns.2448388.194.
[56] S. O. Zayed *et al.*, « The innovation of AI-based software in oral diseases: clinical-histopathological correlation diagnostic accuracy primary study », *BMC Oral Health*, vol. 24, nº 1, p. 598, mai 2024, doi: 10.1186/s12903-024-04347-x.
[57] J. Schwarzmaier *et al.*, « Validation of an Artificial Intelligence-Based Model for Early Childhood Caries Detection in Dental Photographs », *J Clin Med*, vol. 13, nº 17, p. 5215, sept. 2024, doi: 10.3390/jcm13175215.
[58] A. Nayak *et al.*, « Use of Voice-Based Conversational Artificial Intelligence for Basal Insulin Prescription Management Among Patients With Type 2 Diabetes: A Randomized Clinical Trial », *JAMA Netw Open*, vol. 6, nº 12, p. e2340232, déc. 2023, doi: 10.1001/jamanetworkopen.2023.40232.
[59] C. Leclercq *et al.*, « Wearables, telemedicine, and artificial intelligence in arrhythmias and heart failure: Proceedings of the European Society of Cardiology Cardiovascular Round Table », *Europace*, vol. 24, nº 9, p. 1372-1383, oct. 2022, doi: 10.1093/europace/euac052.
[60] J. Tantray, A. Patel, S. N. Wani, S. Kosey, et B. G. Prajapati, « Prescription Precision: A Comprehensive Review of Intelligent Prescription Systems », *Curr Pharm Des*, vol. 30, nº 34, p. 2671-2684, 2024, doi: 10.2174/0113816128321623240719104337.
[61] M. Nambiar *et al.*, « A drug mix and dose decision algorithm for individualized type 2 diabetes management », *NPJ Digit Med*, vol. 7, nº 1, p. 254, sept. 2024, doi: 10.1038/s41746-024-01230-5.
[62] R. Nimri *et al.*, « Insulin dose optimization using an automated artificial intelligence-based decision support system in youths with type 1 diabetes », *Nat Med*, vol. 26, nº 9, p. 1380-1384, sept. 2020, doi: 10.1038/s41591-020-1045-7.
[63] S. Medanki *et al.*, « Artificial intelligence powered glucose monitoring and controlling system: Pumping module », *World J Exp Med*, vol. 14, nº 1, p. 87916, mars 2024, doi: 10.5493/wjem.v14.i1.87916.
[64] T. Shiwani *et al.*, « New Horizons in artificial intelligence in the healthcare of older people », *Age Ageing*, vol. 52, nº 12, p. afad219, déc. 2023, doi: 10.1093/ageing/afad219.
[65] M. Salvi *et al.*, « Multi-modality approaches for medical support systems: A systematic review of the last decade », *Information Fusion*, vol. 103, p. 102134, mars 2024, doi: 10.1016/j.inffus.2023.102134.
[66] F. Krones, U. Marikkar, G. Parsons, A. Szmul, et A. Mahdi, « Review of multimodal machine learning approaches in healthcare », *Information Fusion*, vol. 114, p. 102690, févr. 2025, doi: 10.1016/j.inffus.2024.102690.
[67] D. Lahat, T. Adali, et C. Jutten, « Challenges in Multimodal Data Fusion », in *EUSIPCO 2014 - 22th European Signal Processing Conference*, Lisbonne, Portugal, sept. 2014, p. 101-105. Consulté le: 19 avril 2025. [En ligne]. Disponible sur: https://hal.science/hal-01062366
[68] A. Waqas, A. Tripathi, R. P. Ramachandran, P. A. Stewart, et G. Rasool, « Multimodal data integration for oncology in the era of deep neural networks: a review », *Front. Artif. Intell.*, vol. 7, p. 1408843, juill. 2024, doi: 10.3389/frai.2024.1408843.
[69] J. R. Teoh, J. Dong, X. Zuo, K. W. Lai, K. Hasikin, et X. Wu, « Advancing healthcare through multimodal data fusion: a comprehensive review of techniques and applications », *PeerJ Computer Science*, vol. 10, p. e2298, oct. 2024, doi: 10.7717/peerj-cs.2298.
[70] « (PDF) Multimodal Data Fusion Techniques », ResearchGate. Consulté le: 19 avril 2025. [En ligne]. Disponible sur: https://www.researchgate.net/publication/383887675_Multimodal_Data_Fusion_Techniques

[71] « Lung Cancer ». Consulté le: 21 mai 2025. [En ligne]. Disponible sur: https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer
[72] « Chest CT-Scan images Dataset ». Consulté le: 21 mai 2025. [En ligne]. Disponible sur: https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images
[73] « Lung and Colon Cancer Histopathological Images ». Consulté le: 23 mai 2025. [En ligne]. Disponible sur: https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images
[74] Y. Liu *et al.*, « Prognostic Prediction Models Based on Clinicopathological Indices in Patients With Resectable Lung Cancer », *Front. Oncol.*, vol. 10, oct. 2020, doi: 10.3389/fonc.2020.571169.
[75] M. Q. Shatnawi, Q. Abuein, et R. Al-Quraan, « Deep learning-based approach to diagnose lung cancer using CT-scan images », *Intelligence-Based Medicine*, vol. 11, p. 100188, janv. 2025, doi: 10.1016/j.ibmed.2024.100188.
[76] M. Alotaibi *et al.*, « Exploiting histopathological imaging for early detection of lung and colon cancer via ensemble deep learning model », *Sci Rep*, vol. 14, n° 1, p. 20434, sept. 2024, doi: 10.1038/s41598-024-71302-9.
[77] A. Talukder, M. Islam, A. Uddin, A. Akhter, K. F. Hasan, et M. A. Moni, « Machine Learning-based Lung and Colon Cancer Detection using Deep Feature Extraction and Ensemble Learning ».

Année universitaire : 2024-2025	**Présenté par :** MAYOUF Roua SAKRAOUI Chourouk

Analyse de données bioinformatiques et prédiction par approches basées sur l'IA dans le diagnostic du cancer pulmonaire non à petites cellules.

Mémoire pour l'obtention du diplôme de Master en Bioinformatique

Le cancer du poumon non à petites cellules (CPNPC) constitue un enjeu crucial en oncologie, particulièrement pour ce qui est d'établir un diagnostic rapide et exact. Le travail de recherche que nous exposons dans ce mémoire aborde la création et l'implémentation de DiagnoLung, une plateforme interactive fondée sur l'intelligence artificielle, plus précisément sur des modèles multimodaux dans l'aide à la détection et au diagnostic du CPNPC. DiagnoLung combine et analyse des données cliniques ainsi que des images médicales, comme les images histopathologiques de biopsies de tumeurs solides et les tomodensitométries thoraciques (CT Scan), dans le but de soutenir les oncologues dans la détection précoce et le diagnostic du CPNPC. En intégrant diverses sources d'informations médicales, et l'intelligence artificielle nous participons à l'amélioration du diagnostic par une optimisation de la décision clinique dans le contexte du cancer du poumon et plus largement les autres cancers solides.

Mots-clefs : Cancer du poumon, CPNPC, Intelligence artificielle, Modèles multimodaux, Données cliniques, Imagerie médicale.

Laboratoires de recherche :laboratoire de ……..…….(U Constantine 1 Frères Mentouri).

Président du jury : Dr AMINE KHODJA Ihsene Rokia (MC(B) / PROF- UFM Constantine 1).

Encadrant : Dr BENSAADA Mostafa (MC(A) / PROF- UFM Constantine 1).

Examinateur(s) : Dr Mohamed Skander DAAS (MC(A)/ PROF - UFM Constantine 1),