



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université des Sciences et de la Technologie Houari Boumediene**

Faculté d’Informatique  
Département d’Intelligence Artificielle et  
Sciences des Données (DIASD)

Mémoire de Master

Filière : Informatique

Spécialité : Informatique Visuelle

---

**Thème**

**Prédiction et Détection des cancers de poumons basée sur  
La théorie des jeux  
Et l'apprentissage automatique**

---

**Sujet Proposé par :**

**Mme DAHMANI Djamilia**

**Soutenu le : 20/09/2022**

**Présenté par :**

**Devant le jury composé de :**

**SEGUENI Seif Eddine  
TOUABTI Mohamed**

**Mme BAHA Nadia Présidente**

**Mme TAMEN Zahia Membre**

# Remerciements

Tout d'abord, nous tenons à remercier Dieu de nous avoir donné la force et la motivation pour mener à bien ce projet et la volonté de continuer dans les moments les plus difficiles. Nous remercions notre promotrice Mme Dahmani d'avoir accepté de nous encadrer, de nous avoir suivi tout au long du projet, de nous avoir conseillé et guidé et d'avoir été disponible à tout moment. Nous remercions les membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner. Nous tenons à remercier tous les enseignants qui ont participé à notre formation depuis notre première année depuis notre première année, pour leur travail et leur dévouement. Et enfin merci nos familles et collègues qui nous ont accompagnés tout au long de cette période en nous aidant et en nous soutenant.

# Dédicaces

Je dédie ce succès à :

A ma famille qui m'a soutenu et encouragé durant ces années d'études et qui est la raison pour laquelle je suis à ce stade dans les meilleures conditions possibles.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès et de bonheur.

A Madame Dahmani avec qui nous avons eu la chance de travailler et d'apprendre, merci pour cette belle expérience.

J'aimerais remercier tout particulièrement mon collègue Salim Aissi pour son soutien et son aide et pour avoir été un ami formidable tout au long de notre formation.

Enfin, j'aimerais remercier mon binôme Touabti Mohamed que je considère comme un frère pour avoir fait partie de cette réussite et partagé tous les bons et durs moments depuis le jour où je t'ai connu, et sûrement d'autres à venir avec plus de succès.

Seif-Eddine

## Dédicaces

Dans le cadre de l'achèvement de notre projet de fin d'études, je tiens à dédier ce travail :  
À mes très chers parents, que je remercie profondément pour tous leurs sacrifices et leurs encouragements de tous les instants.

À Madame Dahmani avec qui nous avons eu la chance de travailler et d'apprendre, merci pour cette grande expérience.

À tous mes amis et toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce à la réussite de ce projet.

À mon binôme Segueni Seif Eddine pour son soutien ,sa patience et sa compréhension tout au long de ce projet.

Touabti Mohamed

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 État de l'art</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Généralités sur le cancer du poumon . . . . .	2
1.2.1 Le poumon . . . . .	2
1.2.2 Nodule pulmonaire . . . . .	3
1.2.3 Le cancer du poumon . . . . .	3
1.2.3.1 Les cancers du poumon à petites cellules . . . . .	3
1.2.3.2 Les cancers du poumon à non petites cellules . . . . .	4
1.2.4 les causes du cancer du poumon . . . . .	5
1.2.5 Statistiques sur le cancer du poumon . . . . .	5
1.2.6 Diagnostic précoce . . . . .	5
1.3 L'imagerie médicale . . . . .	5
1.3.1 Tomodensitométrie du thorax . . . . .	5
1.3.2 L'examen tomodensitométrique . . . . .	6
1.3.3 Les limites de la tomodensitométrie . . . . .	6
1.4 La détection du cancer du poumon . . . . .	7
1.4.1 Détection assistée par ordinateur . . . . .	7
1.4.2 Les travaux connexes . . . . .	8
1.4.2.1 DeepEM : ConvNets 3D profonds avec EM pour détection faiblement supervisée de nodules pulmonaires (2018) . . . . .	8
1.4.2.2 DeepSeed : Réseaux neuronaux convolutifs encodeur-décodeur 3D à compression et excitation pour la détection des nodules pulmonaires (2019) . . . . .	8
1.4.2.3 Modèles Genesis : Modèles génériques auto didactiques pour l'analyse d'images médicales 3D (2019) . . . . .	8
1.4.2.4 DeepLung : Réseaux 3D profonds à double chemin pour la détection et la classification automatisées des nodules pulmonaires (2018) . . . . .	9
1.4.2.5 CNN multi-niveaux pour la classification des nodules pulmonaires avec Optimisation des hyperparamètres assistée par le processus gaussien (2019) . . . . .	9
1.5 Conclusion . . . . .	9

<b>2 Outils de conception</b>	<b>10</b>
2.1 Introduction . . . . .	10
2.2 Traitement d'image . . . . .	10
2.2.1 Définition . . . . .	10
2.2.2 Historique . . . . .	10
2.2.3 L'image . . . . .	11
2.2.4 La numérisation . . . . .	11
2.2.4.1 Échantillonnage . . . . .	11
2.2.4.2 Quantification . . . . .	11
2.2.5 Histogramme . . . . .	11
2.2.6 Opérations arithmétiques sur les images . . . . .	12
2.2.6.1 Addition d'images . . . . .	12
2.2.6.2 Soustraction d'images . . . . .	13
2.2.6.3 Division d'images . . . . .	13
2.2.6.4 Multiplication d'images . . . . .	13
2.2.7 Représentation des couleurs . . . . .	14
2.2.7.1 L'espace RGB . . . . .	14
2.2.7.2 L'espace CIE XYZ . . . . .	14
2.2.7.3 L'espace HSV . . . . .	15
2.2.8 Les unités Hounsfield (HU) . . . . .	15
2.2.9 Filtrage . . . . .	16
2.2.9.1 Les filtres passe-bas . . . . .	16
2.2.9.1.1 Les filtres linéaires . . . . .	16
2.2.9.1.2 Les filtres non linéaires . . . . .	17
2.2.9.2 Les filtres passe-haut . . . . .	19
2.2.9.2.1 Le gradient . . . . .	19
2.2.9.2.2 Laplacien . . . . .	19
2.2.10 Segmentation . . . . .	19
2.2.10.1 Approche basée région . . . . .	19
2.2.10.1.1 Méthode de seuillage . . . . .	19
2.2.10.1.2 Méthode de croissance de région (Region growing) .	19
2.2.10.1.3 Méthode de division et fusion . . . . .	20
2.2.10.2 Approche basée countour . . . . .	20
2.2.10.2.1 Méthode de Sobel . . . . .	20
2.2.10.2.2 Méthode de Canny . . . . .	21
2.2.10.3 Approche hybride . . . . .	21
2.2.10.3.1 Ligne des partages des eaux (Water-shed) . . . . .	21
2.2.10.3.2 Contour actif (Snake) . . . . .	22
2.3 Apprentissage profond . . . . .	22
2.3.1 Définition . . . . .	22
2.3.2 Réseaux de neurones artificielles (ANN) . . . . .	23
2.3.3 Réseaux de neurones convolutifs (CNN) . . . . .	23
2.3.3.1 Définition . . . . .	23
2.3.3.2 Structure générale . . . . .	24
2.3.3.3 Les fonctions d'activations . . . . .	25
2.3.3.3.1 Fonction à pas binaire . . . . .	25
2.3.3.3.2 Linéaire . . . . .	26
2.3.3.3.3 Sigmoid . . . . .	27
2.3.3.3.4 Tangente hyperbolique (Tanh) . . . . .	27

## TABLE DES MATIÈRES

---

2.3.3.3.5	Unités linéaires rectifiées (ReLU) . . . . .	28
2.3.3.3.6	Leaky ReLU . . . . .	28
2.3.3.3.7	ReLU paramétré . . . . .	29
2.3.3.3.8	Unité linéaire exponentielle (ELU) . . . . .	29
2.3.3.3.9	Swish . . . . .	30
2.3.3.3.10	SoftMax . . . . .	31
2.3.3.4	Technique de régularisation . . . . .	31
2.3.3.4.1	Dropout . . . . .	31
2.3.3.4.2	Drop-Weights . . . . .	32
2.3.3.4.3	Augmentation des données . . . . .	32
2.3.3.4.4	Batch Normalization . . . . .	32
2.3.3.5	Les architectures du CNN . . . . .	33
2.3.3.5.1	U-Net . . . . .	33
2.3.3.5.2	ResNet . . . . .	34
2.3.3.5.3	DenseNet . . . . .	35
2.3.3.5.4	Inception :ResNet and Inception-V3/4 . . . . .	36
2.4	Théorie des jeux . . . . .	37
2.4.1	Définition . . . . .	37
2.4.2	Historique . . . . .	37
2.4.3	Composants d'un jeu . . . . .	38
2.4.4	Types d'un jeu . . . . .	38
2.4.4.1	Un jeu coopératif . . . . .	38
2.4.4.2	Un jeu non coopératif . . . . .	38
2.4.5	Les formes d'un jeu . . . . .	38
2.4.5.1	La forme normale . . . . .	38
2.4.5.2	La forme extensive . . . . .	39
2.4.6	Les stratégies dominantes . . . . .	39
2.4.7	Les stratégies dominées . . . . .	39
2.4.8	L'équilibre de Nash . . . . .	39
2.4.9	Jeu à somme nulle . . . . .	40
2.4.10	MaxMin et MinMax . . . . .	40
2.4.10.1	MaxMin . . . . .	40
2.4.10.2	MinMax . . . . .	40
2.4.11	Stratégies prudentes . . . . .	40
2.4.11.1	Stratégie prudente en pure . . . . .	40
2.4.11.2	Stratégie prudente en mixte . . . . .	41
2.4.12	Valeur du jeu . . . . .	41
2.4.12.1	Valeur du jeu en pure . . . . .	41
2.4.12.2	Valeur du jeu en mixte . . . . .	41
2.4.13	Exemples d'applications de la théorie des jeux . . . . .	41
2.4.13.1	Décisions relatives à la fixation des prix des produits . . . . .	41
2.4.13.2	Stratégies de guerre . . . . .	41
2.4.13.3	Utilisation de la théorie des jeux dans la Segmentation d'image . . . . .	42
2.4.13.4	Utilisation de la théorie des jeux dans la conception de jeux vidéo . . . . .	42
2.5	Conclusion . . . . .	42

<b>3 Conception du système</b>	<b>43</b>
3.1 Introduction . . . . .	43
3.2 Pré-traitement des données . . . . .	44
3.2.1 Chargement et conversion des scans en HU . . . . .	44
3.2.2 L'extraction du masque des poumons . . . . .	44
3.2.2.1 Binarisation des images par seuillage . . . . .	45
3.2.2.2 La sélection du domaine connecté correspondant aux poumons	45
3.2.3 L'enveloppe convexe et dilatation . . . . .	46
3.2.4 Normalisation de l'intensité . . . . .	48
3.2.5 Préparation des informations des nodules . . . . .	49
3.3 La Segmentation des nodules . . . . .	49
3.3.1 Segmentation 2d des nodules pulmonaires par tranches . . . . .	49
3.3.1.1 Pré-traitement des données d'entrée . . . . .	49
3.3.1.2 Architecture du modèle . . . . .	50
3.3.2 Segmentation 3d des nodules pulmonaires par blocs . . . . .	52
3.3.2.1 Pré-traitement des données d'entrée . . . . .	52
3.3.2.2 Architecture du modèle . . . . .	53
3.4 La classification des nodules pulmonaires . . . . .	55
3.4.1 Les modèles d'apprentissage profond . . . . .	55
3.4.2 Entraînement des modèles . . . . .	56
3.4.3 Modèle de la théorie des jeux . . . . .	57
3.4.3.1 Les joueurs . . . . .	57
3.4.3.2 Les stratégies . . . . .	57
3.4.3.3 Fonction d'utilité . . . . .	58
3.4.3.4 Valeur finale du jeu . . . . .	58
3.5 Conclusion . . . . .	59
<b>4 Résultats et implémentation</b>	<b>60</b>
4.1 Introduction . . . . .	60
4.2 Environnements de travail . . . . .	60
4.2.1 Environnement matériel . . . . .	60
4.2.2 Environnement logiciel . . . . .	61
4.2.2.1 Python . . . . .	61
4.2.2.2 OpenCV . . . . .	61
4.2.2.3 Pydicom . . . . .	62
4.2.2.4 SimpleITK . . . . .	62
4.2.2.5 TensorFlow . . . . .	62
4.2.2.6 Keras . . . . .	63
4.2.2.7 Scikit-learn . . . . .	63
4.2.2.8 Nashpy . . . . .	63
4.2.2.9 PyQt . . . . .	64
4.2.2.10 Visual Studio Code . . . . .	64
4.3 Les bases de données utilisées . . . . .	64
4.3.1 LUNA16 . . . . .	64
4.3.2 DSB 2017 . . . . .	65
4.3.3 Différence entre DSB et LUNA . . . . .	65
4.4 Présentation de l'application . . . . .	66
4.4.1 Écran d'accueil . . . . .	66
4.4.2 Écran d'information . . . . .	66

## TABLE DES MATIÈRES

---

4.4.3	Écran de chargement des scans . . . . .	67
4.4.4	Écran de visualisation du scan . . . . .	68
4.4.4.1	La barre d'outils . . . . .	69
4.4.4.2	Le tableau des métadonnées . . . . .	70
4.4.4.3	Le visualiseur de scan . . . . .	70
4.4.4.4	Le bouton de pré-traitement . . . . .	71
4.4.5	Page de visualisation du scanner pré-traité . . . . .	72
4.4.6	Page des résultats finaux . . . . .	73
4.4.6.1	Tableau des nodules détectés . . . . .	75
4.4.6.2	Tableau d'informations sur les nodules . . . . .	76
4.4.6.3	Visualiseur de blocs de nodules . . . . .	76
4.4.7	Écran de visualisation du modèle 3D . . . . .	77
4.5	Expérimentation et validation . . . . .	77
4.5.1	Les critères d'évaluation . . . . .	77
4.5.1.1	Accuracy . . . . .	78
4.5.1.2	Précision . . . . .	78
4.5.1.3	Rappel (Sensibilité) . . . . .	78
4.5.1.4	Spécificité . . . . .	78
4.5.1.5	La Matrice de confusion . . . . .	78
4.5.1.6	F-Measure . . . . .	79
4.5.1.7	Courbe ROC et AUROC . . . . .	79
4.5.1.8	Courbe de précision-rappel . . . . .	80
4.5.1.9	Erreur absolue moyenne(MAE) . . . . .	80
4.5.1.10	Intersection-Over-Union . . . . .	80
4.5.1.11	Dice Coefficient . . . . .	81
4.5.2	L'évaluation des modèles de la segmentation . . . . .	82
4.5.2.1	Affichage des Résultats de la segmentation 2d et 3d . . . . .	83
4.5.3	Évaluation des résultats de la classification . . . . .	84
4.5.3.1	Courbes ROC des résultats des modèles . . . . .	85
4.5.3.2	Courbes Précision-Rappel des résultats des modèles . . . . .	86
4.5.3.3	Comparaison des AUC-ROC et AUC-Précision-Rappel entre les modèles . . . . .	87
4.5.3.4	Des exemples de la classification . . . . .	87
4.6	Conclusion . . . . .	89
	<b>Conclusion générale</b>	<b>90</b>
	<b>Bibliographie</b>	<b>98</b>

# Table des figures

1.1	structure du poumon [103] . . . . .	2
1.2	Exemples des scans avec des nodules pulmonaires [38] . . . . .	3
1.3	Les cancers du poumon à petites cellules [21] . . . . .	4
1.4	Les cancers du poumon à non petites cellules [19] . . . . .	4
1.5	Des exemple sur les CT scans du poumons [63] . . . . .	6
1.6	Exemple d'un patient subissant un CT scan [37] . . . . .	6
1.7	Scanner CT moderne 'Canon' [96] . . . . .	7
2.1	Exemple d'une opération d'L'échantillonnage [115] . . . . .	11
2.2	Deux images avec leurs histogrammes [119] . . . . .	12
2.3	Addition de deux images [91] . . . . .	12
2.4	Soustraction de deux images [91] . . . . .	13
2.5	Division de deux images [91] . . . . .	13
2.6	Cube de Maxwell [120] . . . . .	14
2.7	L'espace XYZ [95] . . . . .	15
2.8	L'espace HSV [120] . . . . .	15
2.9	L'échelle de Hounsfield [10] . . . . .	16
2.10	Exemple du filtre moyenneur [106] . . . . .	17
2.11	Exemple du filtre Gaussien [54] . . . . .	17
2.12	Méthode du filtre médian . . . . .	18
2.13	Exemple du différents filtres morphologiques. . . . .	18
2.14	Exemple de la méthode Region growing. . . . .	20
2.15	Exemple de la méthode division et fusion. . . . .	20
2.16	Exemple d'application du filtre sobel [18] . . . . .	21
2.17	Exemple d'application du filtre canny [110] . . . . .	21
2.18	Exemple de la méthode Water-shed . . . . .	22
2.19	Exemple de la méthode Snake [6] . . . . .	22
2.20	La relation entre l'apprentissage profond,apprentissage automatique et l'intelligence artificielle [86] . . . . .	23
2.21	Exemple d'un simple réseau de neurones . . . . .	23
2.22	Exemple d'une opération de convolution [41] . . . . .	24
2.23	Exemple d'un max pooling [78] . . . . .	25
2.24	Une architecture simple d'un CNN [61] . . . . .	25
2.25	La fonction à pas binaire . . . . .	26
2.26	La fonction linéaire . . . . .	26
2.27	La fonction Sigmoid . . . . .	27

## TABLE DES FIGURES

---

2.28 La fonction Tanh . . . . .	27
2.29 La fonction ReLU . . . . .	28
2.30 La fonction Leaky ReLU . . . . .	29
2.31 La fonction ReLU paramétré . . . . .	29
2.32 La fonction ELU . . . . .	30
2.33 La fonction Swish . . . . .	30
2.34 Exemple d'application du Dropout [80] . . . . .	31
2.35 Exemple d'application de l'augmentation de données [64] . . . . .	32
2.36 Architecture U-net [28] . . . . .	34
2.37 Le schéma fonctionnel de ResNet [25] . . . . .	35
2.38 L'architecture de DenseNet [32] . . . . .	36
2.39 Le schéma fonctionnel de base de l'unité Inception Residual [32] . . . . .	37
2.40 Représentation graphique de la forme extensive d'un jeu . . . . .	39
 3.1 L'architecture de notre système de détection et de classification des nodules pulmonaires . . . . .	43
3.2 Un exemple de coupe de CT-scan après le chargement et la conversion des données en HU . . . . .	44
3.3 Un exemple de masque de poumons lié au fichier '.mhd' . . . . .	45
3.4 Un exemple d'une tranche binarisée du scan. . . . .	45
3.5 Une tranche du masque final extrait du 3d scan . . . . .	46
3.6 Un exemple des masques de poumons séparés afin d'appliquer le L'enveloppe convexe . . . . .	47
3.7 Un exemple des masques des poumons gauche et droit après application de L'enveloppe convexe . . . . .	47
3.8 Un exemple de masque pulmonaire complet regroupé après dilatation . . . . .	48
3.9 Une coupe du CT-scan après la phase de pré-traitement . . . . .	48
3.10 Représentation 3D du CT-scan après le pré-traitement des données . . . . .	49
3.11 Exemples des tranches 2d du scan qui contiennent des nodules pulmonaires avec leurs masques . . . . .	50
3.12 L'architecture du modèle U-net 2D . . . . .	51
3.13 Exemple d'un nodule pulmonaire à l'intérieur d'un bloc 64x64x64 du scan . .	53
3.14 L'architecture du modèle U-net 3D . . . . .	54
3.15 Les 4 architectures utilisés . . . . .	56
3.16 Les étapes du K-fold cross-validation [47] . . . . .	57
 4.1 Python . . . . .	61
4.2 OpenCV . . . . .	62
4.3 Pydicom . . . . .	62
4.4 SimpleITK . . . . .	62
4.5 TensorFlow . . . . .	63
4.6 Keras . . . . .	63
4.7 Scikit-learn . . . . .	63
4.8 PyQt . . . . .	64
4.9 Visual Studio Code . . . . .	64
4.10 Écran d'accueil . . . . .	66
4.11 Écran d'informations . . . . .	67
4.12 Écran de chargement des scans . . . . .	67
4.13 Écran de chargement d'un scan . . . . .	68

## TABLE DES FIGURES

---

4.14 Écran de visualisation du scan . . . . .	69
4.15 La barre d'outils . . . . .	69
4.16 Table de métadonnées . . . . .	70
4.17 Le visualiseur de scan . . . . .	71
4.18 Le bouton de pré-traitement . . . . .	71
4.19 Écran de chargement du pré-traitement du scan. . . . .	72
4.20 Écran de visualisation du scan pré-traité . . . . .	73
4.21 Écran de chargement de la phase de détection . . . . .	73
4.22 Écran de visualisation du résultats de détection . . . . .	74
4.23 Écran de visualisation du résultats de détection lorsque il n'y a pas de nodule détecté . . . . .	75
4.24 Tableau des nodules détectés . . . . .	75
4.25 Tableau des informations du nodule . . . . .	76
4.26 Visualiseur de blocs du nodule . . . . .	76
4.27 Fenêtre de visualisation 3D . . . . .	77
4.28 Exemple d'une matrice de confusion [51] . . . . .	78
4.29 Courbe ROC [77] . . . . .	79
4.30 Quelques courbes théoriques de précision-rappel [66] . . . . .	80
4.31 Visualisation du calcul de l'IoU [82] . . . . .	81
4.32 Illustration du Dice Coefficient [68] . . . . .	81
4.33 Les courbes ROC de tous les modèles dans les ensembles de test différents . . . . .	85
4.34 les courbes Précision-Rappel de tous les modèles dans les ensembles de test différents . . . . .	86

# Liste des tableaux

2.1	Table du jeu dilemme du prisonnier . . . . .	38
3.1	Matrice d'un jeu du notre system . . . . .	58
4.1	Kaggle GPU specs . . . . .	60
4.2	Spécifications des machines personnels . . . . .	61
4.3	Tableau récapitulatif des résultats de la segmentation 2d et 3d en utilisant des métriques différentes . . . . .	82
4.4	Tableau récapitulatif d'exemples de la segmentation 2d et 3d pour des scans aléatoires . . . . .	83
4.5	Tableau des résultats de l'évaluation test moyen pour tout les modeles . . . . .	84
4.6	Tableau récapitulatif des 5 modèles avec leurs AUC et AU-Précision-Rappel . . . . .	87

## Résumé

Le diagnostic automatique du cancer du poumon à partir d'une tomographie (CT) comporte deux étapes : la détection de toutes les lésions suspectes (nODULES pulmonaires) et l'évaluation de la malignité pulmonaire globale. Nous avons proposé un système qui peut faire les deux en utilisant à la fois l'apprentissage profond et la théorie des jeux. En commençant par prétraiter les données d'entrée, qui sont des CT scan, puis en les soumettant à un modèle U-net 3D afin de segmenter les régions suspectes (nODULES). Ensuite, les nODULES détectés seront classés en utilisant un modèle de théorie des jeux qui consiste en un jeu à somme nulle entre 4 architectures de classification : "ResNet50 , DensNet121 , Inception V3 , InceptionResNet V2". Le système a été testé sur les jeux de données DSB 2017 [98] et LUNA16 [104] et il donne des résultats prometteurs.

**Mots clés** : NODULES, Cancer de poumons ; Apprentissage profond, théorie des jeux.

# Abstract

Automatic diagnosis of lung cancer from computed tomography (CT) involves two steps : detection of all suspicious lesions (lung nodules) and evaluation of lung malignancy as a whole. We proposed a system that can do both using both deep learning and game theory. By first preprocessing the input data, which are CT scans, and then subjecting them to a 3D U-net model to segment the suspicious regions (nodules). Afterwards, the detected nodules will be classified using a game theory model which consists of a zero-sum game between 4 classification architectures "ResNet50 , DensNet121 , Inception V3 , InceptionResNet V2". The system has been tested on the DSB 2017 [98] and LUNA16 [104] datasets and shows promising results.

**Keywords** : Nodules, Lung cancer, Deep learning, Game theory.

# Introduction générale

Le cancer du poumon est l'un des cancers malins les plus courants et les plus mortels. Comme pour les autres cancers, la meilleure solution pour le cancer du poumon est un diagnostic précoce et un traitement opportun. Des examens réguliers sont donc nécessaires. La tomographie volumétrique thoracique (CT) est un outil d'imagerie courant pour le diagnostic du cancer du poumon [11]. Elle visualise tous les tissus en fonction de leur absorption des rayons X. Les lésions pulmonaires sont appelées nodules pulmonaires. Un nodule présente généralement le même niveau d'absorption que les tissus normaux, mais sa forme est particulière : les bronches et les vaisseaux sont des systèmes de tuyaux continus, épais à la racine et fins à la ramifications, et les nodules sont généralement sphériques et isolés. Il faut généralement une dizaine de minutes à un médecin expérimenté pour effectuer un examen complet d'un patient, car certains nodules sont petits et difficiles à trouver. En outre, il existe de nombreux sous-types de nodules, et les probabilités de cancer sont différentes selon les sous-types. Les médecins peuvent évaluer la malignité des nodules sur la base de leur morphologie, mais la précision dépend fortement de l'expérience des médecins, et des médecins différents peuvent donner des prédictions différentes [17].

Le diagnostic assisté par ordinateur est adapté à cette tâche. car les modèles de vision par ordinateur peuvent rapidement balayer partout avec une qualité égale et ils ne sont pas affectés par la fatigue et les émotions. Les progrès récents de l'apprentissage profond ont permis modèles de vision par ordinateur d'aider les médecins à diagnostiquer différents problèmes. Divers problèmes et, dans certains cas, les modèles ont fait preuve d'une performance compétitive par rapport aux médecins [20][36][31][39][2].

Dans le cadre de notre projet de fin d'étude nous avons réalisé un système qui détecte d'abord un nodule polmonaire en effectuant une segmentation 3D à l'aide de l'apprentissage profond, puis en utilisant la théorie des jeux afin de prendre la décision de le classer comme malin ou bénin. Toutes ces fonctions ont été réunies dans une application qui facilite l'exploitation de l'information et la visualisation des résultats. Le projet se reportera sur ces quatre chapitres :

- Chapitre 1 : Dans ce chapitre nous présentons en général quelques notions sur le poumon , des généralités sur le cancer du poumon et l'imagerie médicale ainsi que les méthodes de détection mentionnons quelques travaux connexes à la fin.
- Chapitre 2 : Dans ce chapitre nous abordons quelques notions de base de traitement d'image puis l'apprentissage profond et enfin la théorie des jeux.
- Chapitre 3 : dans ce chapitre, nous expliquons le concept et les différentes étapes pour réaliser notre solution.
- Chapitre 4 : Dans ce chapitre, nous présentons l'environnement de développement et les jeux de données utilisés, puis l'évaluation de nos modèles avec quelques résultats.

# État de l'art

## 1.1 Introduction

Pour la mise en œuvre d'un modèle de prédiction du cancer , il est nécessaire de survoler ce qui est relié dans la littérature. Par conséquent, nous présentons dans ce chapitre un état de l'art en commençant par un aperçu sur le cancer du poumon. Par la suite, nous mettons l'accent sur l'imagerie médicale. Enfin, nous parlerons de l'utilisation de la vision par ordinateur et de l'apprentissage profond pour la détection automatique du cancer.

## 1.2 Généralités sur le cancer du poumon

### 1.2.1 Le poumon

Les poumons sont deux organes spongieux remplis d'air, situés de part et d'autre de la poitrine (thorax) [102]. Les poumons sont les principaux organes du système respiratoire, ce qui fait qu'ils sont considérés comme un organe très important parce qu'ils ajoutent de l'oxygène et éliminent le dioxyde de carbone du sang. Sans eux, la vie humaine serait en danger, il est donc indispensable d'en prendre soin.

Chaque poumon est divisé en lobes (sections) [103] :

- Le poumon droit a 3 lobes et est légèrement plus large que le poumon gauche.
- Le poumon gauche a 2 lobes. Le cœur s'appuie sur l'incisure cardiaque, qui est un creux dans le lobe inférieur.

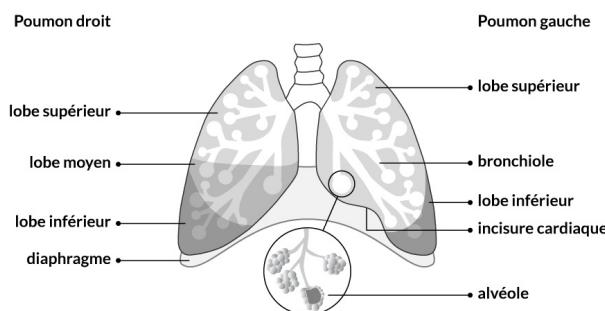


FIGURE 1.1 – structure du poumon [103]

## 1.2.2 Nodule pulmonaire

Les nodules pulmonaires sont de petits amas de cellules dans les poumons. Ils sont très courants. La plupart des nodules pulmonaires sont des tissus cicatriciels résultant d'anciennes infections pulmonaires.

Les nodules pulmonaires ne provoquent généralement pas de symptômes. Ils sont souvent découverts par accident lors d'une radiographie pulmonaire ou d'un scanner effectué pour une autre raison.

Si un nodule pulmonaire est petit et qu'il ne grossit pas, il ne s'agit probablement pas d'un cancer. Il ne nécessite probablement pas de traitement. Votre prestataire de soins de santé peut examiner des examens d'imagerie antérieurs pour voir si le nodule est nouveau ou a changé. Il se peut que vous deviez passer périodiquement des tomodensitogrammes pour voir si le nodule se développe [79].

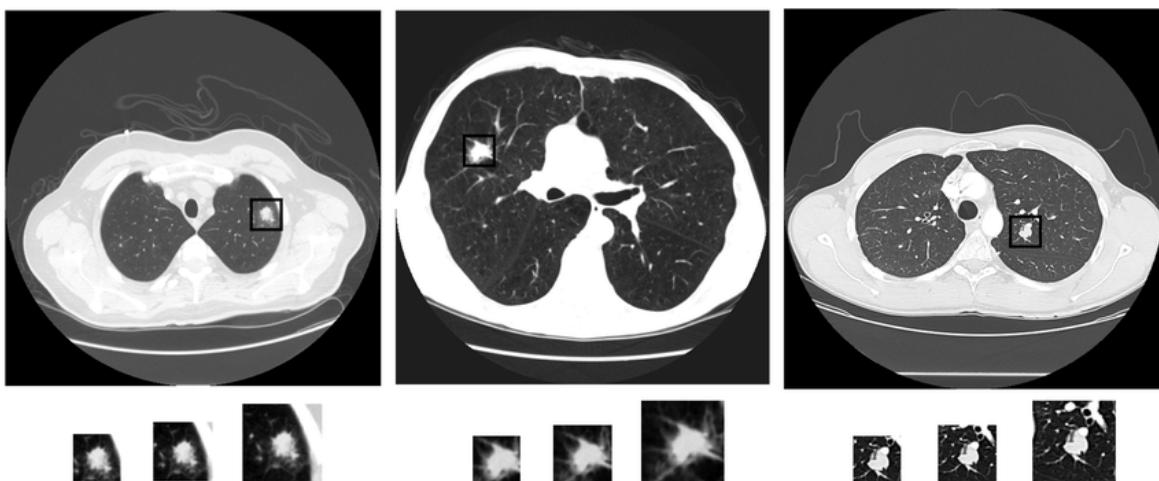


FIGURE 1.2 – Exemples des scans avec des nodules pulmonaires [38]

## 1.2.3 Le cancer du poumon

Le cancer du poumon survient lorsque les cellules des poumons se divisent de manière incontrôlée, entraînant la croissance de tumeurs. Les tumeurs peuvent réduire la capacité d'une personne à respirer, et elles peuvent se propager à d'autres parties du corps [105].

On distingue deux types de cancer du poumon selon l'aspect microscopique des cellules qui en sont à l'origine :

### 1.2.3.1 Les cancers du poumon à petites cellules

Les cancers du poumon à petites cellules représentent environ 15% des cancers du poumon et sont difficiles à soigner. Ces cancers du poumon évoluent rapidement et, lors de leur diagnostic, il est fréquent que des cellules cancéreuses se soient déjà disséminées dans le reste du corps pour former des métastases (des tumeurs secondaires). Dans 95% des cas, les cancers du poumon à petites cellules sont liés au tabagisme [93].

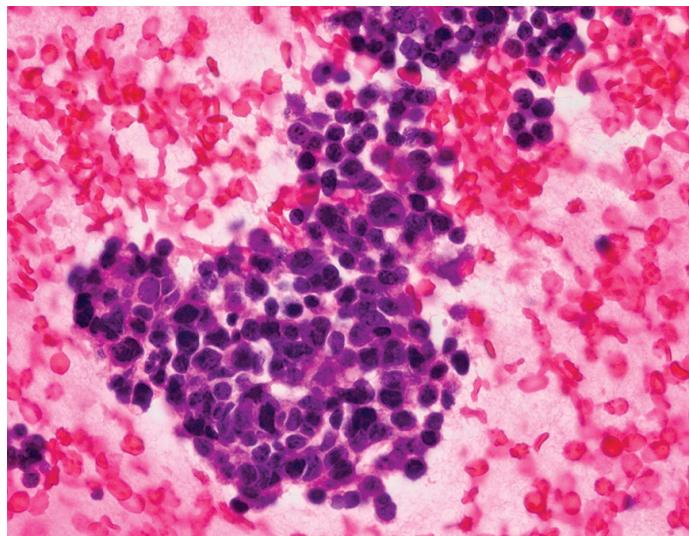


FIGURE 1.3 – Les cancers du poumon à petites cellules [21]

#### 1.2.3.2 Les cancers du poumon à non petites cellules

Les cancers du poumon non à petites cellules représentent environ 85% des cancers du poumon et se soignent plus facilement car ils se développent plus lentement. Ces cancers sont essentiellement de trois types :

- **les adénocarcinomes** : qui représentent 40% des cancers non à petites cellules, touchent parfois les alvéoles et sont un peu plus fréquents chez les non fumeurs et les femmes [93].
- **les carcinomes épidermoïdes** : qui représentent également 40 % des cancers non à petites cellules, touchent plutôt les grosses bronches et sont liés au tabagisme [93].
- **les carcinomes à grandes cellules indifférenciées** : qui représentent 20% des cancers non à petites cellules, ont une croissance plus rapide que les deux autres types et sont liés au tabagisme dans 90% des cas [93].

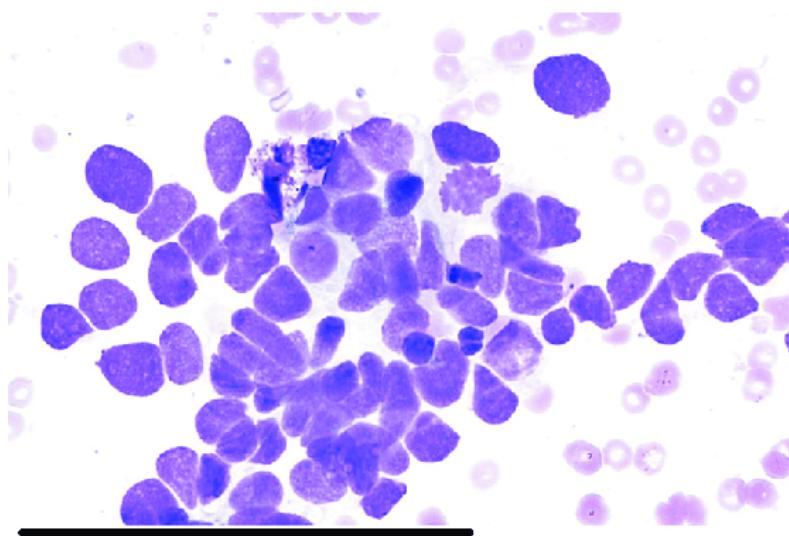


FIGURE 1.4 – Les cancers du poumon à non petites cellules [19]

### 1.2.4 les causes du cancer du poumon

Le tabagisme est la première cause de cancer du poumon, mais plusieurs facteurs peuvent favoriser l'apparition de ce cancer :

- Exposition à la fumée secondaire.
- Radiothérapie antérieure.
- Exposition au gaz radon.
- Exposition à l'amiante et à d'autres substances cancérigènes.
- Antécédents familiaux de cancer du poumon.

### 1.2.5 Statistiques sur le cancer du poumon

Le cancer du poumon constitue actuellement un des principaux problèmes de santé publique avec une incidence mondiale de 1,2 million de nouveaux cas par an. En plus de cela, c'est l'un des cancers les plus meurtriers avec une mortalité touchant 1,3 million de personnes. En Algérie, le cancer du poumon constitue un véritable fléau avec environ 35 000 nouveaux cas/an et 20 000 décès/an. En effet et malgré toutes les avancées thérapeutiques, le pronostic du cancer bronchique reste très mauvais [33].

### 1.2.6 Diagnostic précoce

Plus le cancer est détecté tôt, plus il est facile de le traiter. Pour cela il existe deux méthodes de diagnostic :

- **Examens d'imagerie** : Une masse anormale peut être vue sur la radiographie : IRM, CT et TEP. Ces scans produisent plus de détails et trouvent des lésions plus petites.
- **Cytologie des expectorations** : Examen au microscope des cellules trouvées dans les expectorations. Le test vérifie les cellules anormales, telles que les cellules cancéreuses du poumon [108].

Ces deux diagnostics peuvent être appliqués à des stades précoces pour détecter les tumeurs naissantes, si elles existent, et aider les médecins à prendre des décisions et des mesures précoces pour éviter toute progression indésirable des tumeurs pour le patient, d'autant que ce cancer est considéré comme un type de cancer mortel.

## 1.3 L'imagerie médicale

L'imagerie médicale fait référence à plusieurs technologies différentes qui sont utilisées pour visualiser le corps humain afin de diagnostiquer, de surveiller ou de traiter des conditions médicales. Chaque type de technologie donne des informations différentes sur la zone du corps étudiée ou traitée, liée à une éventuelle maladie, blessure ou à l'efficacité d'un traitement médical [99]. Dans le cas du diagnostic de cancer du poumon, la technologie d'acquisition la plus courante est la tomodensitométrie.

### 1.3.1 Tomodensitométrie du thorax

En anglais c'est “**computed tomography scan**”(CT scan) est le dépistage pulmonaire par tomodensitométrie est une procédure non invasive et indolore qui utilise des rayons X à faible dose pour dépister le cancer des poumons en seulement 30 secondes. Un dépistage pulmonaire CT permet au radiologue d'examiner différents niveaux, ou tranches, des poumons à l'aide d'un faisceau de rayons X rotatif. Elle est réalisée sur un scanner de tomodensitométrie

(TDM) en spirale multi-coupe et peut détecter des nodules ou un cancer plus petit que les radiographies pulmonaires standard [97].

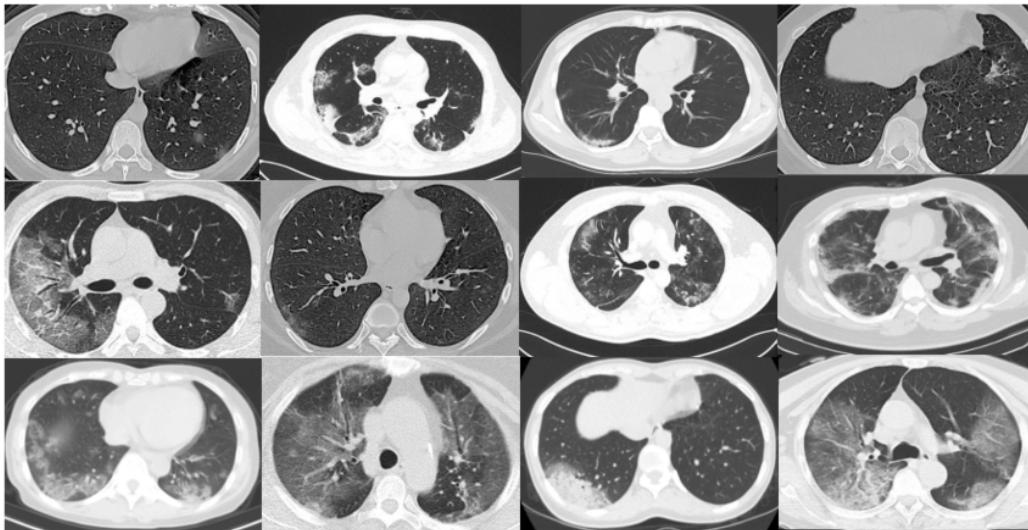


FIGURE 1.5 – Des exemple sur les CT scans du poumons [63]

### 1.3.2 L'examen tomodensitométrique

Pendant l'examen, les patients sont allongés sur le dos sur une table d'examen, les bras au-dessus de la tête. Ils doivent retenir brièvement leur souffle pendant que les photos sont prises. Pendant une courte période, le corps peut être couvert par un scanner, mais le scanner est ouvert à l'arrière et à l'avant pour que le patient puisse voir. Le technologue est toujours en mesure de voir et d'entendre le patient pendant la procédure de cinq minutes [97].



FIGURE 1.6 – Exemple d'un patient subissant un CT scan [37]

### 1.3.3 Les limites de la tomodensitométrie

Même avec la bonne précision et les fonctionnalités utiles du scanner, il a encore quelques limitations et inconvénients, en réduisant certains :

- Excessivement cher.
- Pas toujours disponible.
- Il ne peut pas remplacer un historique médical ou un examen physique pertinent.

- Il ne peut pas fournir de preuves histologiques.



FIGURE 1.7 – Scanner CT moderne ‘Canon’ [96]

## 1.4 La détection du cancer du poumon

Le cancer du poumon est bien connu pour son comportement mortel, aussi sa détection précoce est-elle la meilleure solution, bien que ce soit un véritable défi en raison du comportement progressif de la tumeur.

Heureusement, ces dernières années, grâce au développement rapide de la technologie des logiciels et du matériel informatiques et de la technologie d'apprentissage profond et la et l'expansion des techniques de la vision par ordinateur, la technologie du diagnostic assisté par ordinateur a obtenu des résultats révolutionnaires et a progressivement démontré sa valeur clinique dans le diagnostic. Un système de diagnostic de cancer du poumon se base principalement sur les processus suivants :

- **Acquisition des images** : Capturez les images et les informations nécessaires et regroupez-les dans un ensemble de données pour pouvoir travailler avec elles.
- **Pré-traitements des images** : Cette phase est très importante et c'est le principal facteur de réussite ou d'échec du modèle de détection. Dans cette phase on applique des filtres pour diminuer le bruit... etc.
- **Segmentation** : isoler les parties suspectes et ne conserver que les informations nécessaires afin de faciliter le travail et de réduire les ressources nécessaires à l'exécution du processus.
- **Classification** : En utilisant les résultats de la segmentation comme entrée afin de raffiner la classification.

### 1.4.1 Détection assistée par ordinateur

La détection assistée par ordinateur (**Computer aided detection CAD**) est une technologie conçue pour réduire les erreurs d'observation - et donc le taux de faux négatifs - des médecins qui interprètent des images médicales. Des études cliniques prospectives ont démontré une augmentation de la détection du cancer en général avec l'aide de la CAD. Cet aperçu décrit brièvement les paramètres qui ont été utilisés pour définir les performances des systèmes de CAD [7]. La CAD est progressivement intégrée dans le flux de travail de l'hôpital

comme un "second lecteur". Grâce à l'image médicale, la technologie de traitement de l'image, combinée aux capacités d'analyse de l'ordinateur, améliore considérablement l'efficacité et la précision du diagnostic des médecins de la lecture des scans [83].

## 1.4.2 Les travaux connexes

Dans cette section, nous citerons certains des travaux relatifs à la détection et à la classification du cancer du poumon qui ont été publiés de certaines études de recherche :

### 1.4.2.1 DeepEM : ConvNets 3D profonds avec EM pour détection faiblement supervisée de nodules pulmonaires (2018)

L'auteur de ce travail a proposé DeepEM, un nouveau cadre de réseau convolutif 3D profond augmenté de l'expectation-maximisation (EM), pour exploiter des annotations faiblement supervisées dans les dossiers médicaux électroniques pour la détection de nodules pulmonaires. Les résultats expérimentaux montrent que DeepEM peut conduire à une amélioration moyenne de 1,5% et 3,9% des scores FROC (free-response receiver operating characteristic) sur les ensembles de données LUNA16 [104] démontrer l'utilité des informations incomplètes contenues dans les dossiers médicaux électroniques pour améliorer les algorithmes d'apprentissage profond [48].

### 1.4.2.2 DeepSeed : Réseaux neuronaux convolutifs encodeur-décodeur 3D à compression et excitation pour la détection des nodules pulmonaires (2019)

Dans ce travail, ils développent un nouveau réseau neuronal convolutif 3D profond avec une structure encodeur-décodeur en conjonction avec un réseau de proposition de région. En particulier, ils utilisent une perte d'entropie croisée à échelle dynamique pour réduire le taux de faux positifs et combattre le problème de déséquilibre de l'échantillon associé à la détection des nodules. Ils adoptent la structure squeeze-and-excitation pour apprendre des caractéristiques d'image efficaces et utilisent les informations d'interdépendance de différentes cartes de caractéristiques. Ils ont validé leur méthode sur la base de tomodensitogrammes accessibles au public avec des données de base étiquetées manuellement obtenues à partir du jeu de données LIDC/IDRI [14] et de son sous-ensemble LUNA16 [104] avec des tranches plus fines. Les études d'ablation et les résultats expérimentaux ont démontré que leur méthode pouvait surpasser largement les méthodes de détection de nodules les plus récentes [56].

### 1.4.2.3 Modèles Genesis : Modèles génériques auto didactiques pour l'analyse d'images médicales 3D (2019)

Ce travail propose un ensemble de modèles, appelés "**Generic Autodidactic Models**", surnommés Modèles Genesis, car ils sont créés sans annotations manuelles, autodidactes (apris par auto-supervision), et génériques (ils servent de modèles sources pour la génération de modèles cibles d'applications spécifiques). Ces modèles surpassent de manière significative l'apprentissage à partir de zéro dans les cinq applications 3D cibles, couvrant à la fois la segmentation et la classification. Plus important encore, l'apprentissage d'un modèle à partir de zéro simplement en 3D ne donne pas nécessairement de meilleures performances que l'apprentissage par transfert à partir d'ImageNet en 2D, mais les Modèles Genesis surpassent systématiquement toutes les approches 2D, y compris le réglage fin des modèles pré-entraînés à partir d'ImageNet [60].

#### **1.4.2.4 DeepLung : Réseaux 3D profonds à double chemin pour la détection et la classification automatisées des nodules pulmonaires (2018)**

DeepLung se compose de deux éléments, la détection des nodules (identification de l'emplacement des nodules candidats) et la classification (classification des nodules candidats en bénins ou malins). Compte tenu de la nature 3D des données de tomodensitométrie pulmonaire et de la compacité des réseaux à double chemin (DPN), deux DPN 3D profonds sont conçus pour la détection des nodules et la classification respectivement. Plus précisément, un réseau de neurones convolutifs 3D à régions rapides (R-CNN) est conçu pour la détection des nodules avec des blocs 3D à double chemin et une structure d'encodage-décodage de type U-net pour apprendre efficacement les caractéristiques des nodules. Pour la classification des nodules, une machine d'amplification du gradient (GBM) avec des caractéristiques de réseau à double chemin en 3D est proposée. Le sous-réseau de classification des nodules a été validé sur un ensemble de données publiques du LIDC-IDRI [14], sur lequel il a obtenu de meilleures performances que les approches de pointe et a dépassé les performances des médecins expérimentés en fonction de la modalité d'image. Dans le système DeepLung, les nodules candidats sont d'abord détectés par le sous-réseau de détection des nodules, puis le diagnostic des nodules est effectué par le sous-réseau de classification. Des résultats expérimentaux détaillés montrent que la méthode du DeepLung possède une précision remarquable et de bons résultats [49].

#### **1.4.2.5 CNN multi-niveaux pour la classification des nodules pulmonaires avec Optimisation des hyperparamètres assistée par le processus gaussien (2019)**

Cet article étudie la classification des nodules pulmonaires à l'aide de réseaux de neurones profonds (DNN). Les DNN ont montré leur supériorité sur plusieurs problèmes de traitement d'images médicales, comme la segmentation d'images médicales, la synthèse d'images médicales, etc., mais leurs performances dépendent fortement de la configuration appropriée des hyperparamètres. L'optimisation des hyperparamètres dans les DNN est un problème coûteux en calcul, où l'évaluation d'une configuration optimale peut prendre plusieurs heures, voire des jours. L'optimisation bayésienne a été récemment introduite pour la recherche automatique des configurations optimales. Elle applique des modèles de substitution probabilistes pour approximer la fonction d'erreur de validation, tels que les processus gaussiens, et réduit la complexité de calcul [59].

## **1.5 Conclusion**

Nous avons présenté dans ce chapitre, un état de l'art sur les différentes parties directement liées au contexte de notre travail à savoir un aperçu général sur le cancer du poumon puis l'imagerie médicale (en particulier les CT scans), ensuite nous avons expliqué les applications de la vision par ordinateur et de l'apprentissage profond dans la détection automatique des nodules malins. Enfin nous avons exposé certains travaux connexes sur le sujet. Le chapitre suivant sera consacré aux outils de conception.

# Chapitre 2

## Outils de conception

### 2.1 Introduction

Dans le domaine médical, le traitement d’images ainsi que l’apprentissage profond jouent des rôles très importants. Ils permettent en effet d’observer la structure et le comportement de la maladie grâce à un ensemble de techniques efficaces pour un diagnostic rapide et précis des maladies. Ils fournissent ainsi une assistance automatique aux cliniciens.

Ce chapitre est consacré à la présentation des concepts et des méthodologies de traitement d’images. Ensuite nous allons découvrir les fondamentaux de l’apprentissage profond y compris les détails sur les modèles utilisés dans notre travail pour la segmentation et la classification. Enfin, nous allons découvrir les fondamentaux de la théorie des jeux que nous avons utilisée comme outil de décision en cas de conflit de classification.

### 2.2 Traitement d’image

#### 2.2.1 Définition

Le traitement d’image est une méthode pour effectuer certaines opérations sur une image, afin d’obtenir une image améliorée ou d’en extraire des informations utiles. C’est un type de traitement dans lequel l’entrée est une image et la sortie peut être une image ou des caractéristiques associées à cette image. Il constitue également un domaine de recherche dans les disciplines de l’ingénierie et de l’informatique [57].

#### 2.2.2 Historique

Le traitement d’image et vision sont des disciplines relativement jeunes (années 60) et qui évoluent rapidement. Elles sont en plein essor et donnent lieu chaque année à une multitude de travaux, académiques, technologiques, industriels. Cette expansion s’explique par la difficulté du domaine :

- Complexité algorithmique due aux énormes volumes de données.
- Caractère mal posé des problèmes.
- Difficultés à formaliser une faculté biologique.

D’autre part, l’engouement pour ces disciplines s’explique par la multiplication permanente d’applications et d’enjeux industriels dans des domaines aussi variés que la médecine, télécommunications, météorologie, art … etc [26].

### 2.2.3 L'image

Une image est une représentation visuelle, voire mentale de quelque chose (objet, être vivant ou concept). Elle peut être naturelle (capturée par caméra, microscope...) ou artificielle (par synthèse d'images, réalité virtuelle), visuelle ou non, tangible ou conceptuelle, elle peut entretenir un rapport de ressemblance directe avec son modèle ou contraire y être liée par un rapport plus symbolique [26]. On distingue deux types d'images :

- **Image analogique** : est un signal 2D ( $x, y$ ) qui représente une matrice de nombres.
- **L'image numérique** : est une image  $f(x, y)$  dont à la fois le domaine spatial et la luminosité ont été discrétisés. Elle est représentée par une matrice d'entiers.

### 2.2.4 La numérisation

La numérisation est l'opération qui transforme les signaux analogiques capturés par une caméra en valeurs numériques et lui donner une représentation. Cette représentation sera la plupart du temps matricielle, c'est-à-dire une matrice où chaque point sera représenté par une couleur. Pour numériser une image, il faut l'échantillonner puis la quantifier, donc une image numérique est une image échantillonnée et quantifiée [72].

#### 2.2.4.1 Échantillonnage

L'échantillonnage est le procédé de discrétilisation spatiale d'une image consistant à associer à chaque zone rectangulaire  $R(x, y)$  d'une image continue une unique valeur  $I(x, y)$  [13].

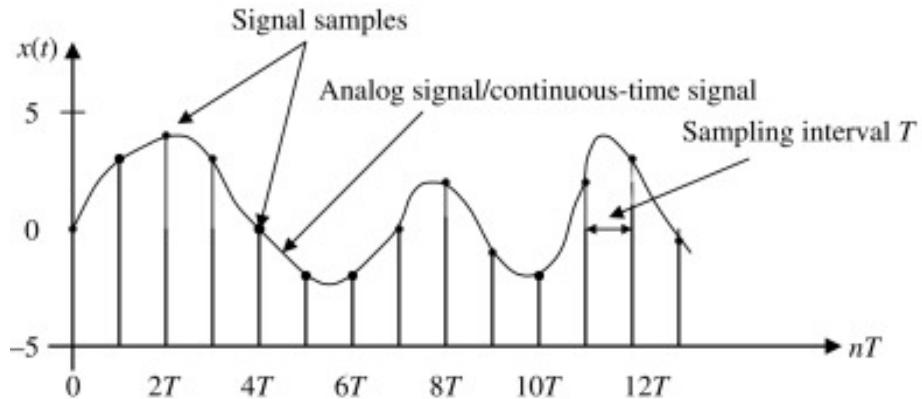


FIGURE 2.1 – Exemple d'une opération d'échantillonnage [115]

#### 2.2.4.2 Quantification

La quantification désigne la limitation du nombre de valeurs différentes que peut prendre  $I(x, y)$ , nombre déterminé en pratique par le nombre d'octets sur lequel on code la valeur numérique en question [13].

### 2.2.5 Histogramme

L'histogramme représente la répartition des pixels en fonction de leur niveau de gris. Les histogrammes sont fréquemment utilisés pour l'amélioration d'images qui consiste à modifier les caractéristiques visuelles de l'image de manière à en faciliter son interprétation par l'œil

humain. Il peut s'agir de rehausser les contrastes, d'accentuer certaines intensités pour mettre en valeur une région. Il est également utilisé pour la segmentation [72].

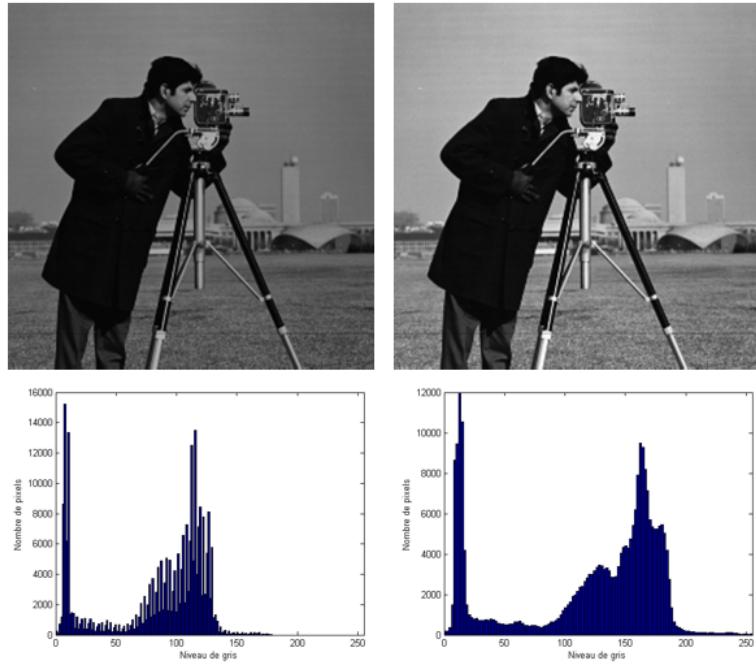


FIGURE 2.2 – Deux images avec leurs histogrammes [119]

## 2.2.6 Opérations arithmétiques sur les images

### 2.2.6.1 Addition d'images

L'addition de deux images  $f$  et  $g$  de même taille est une nouvelle image  $h$  de même taille dont chaque pixel correspond à la somme des valeurs des pixels des images originales [72], on peut la définir par :

$$r(x, y) = \min(f(x, y) + g(x, y), 2^N - 1) \text{ avec } N = \text{Nombres de bits de codage} \quad (2.1)$$

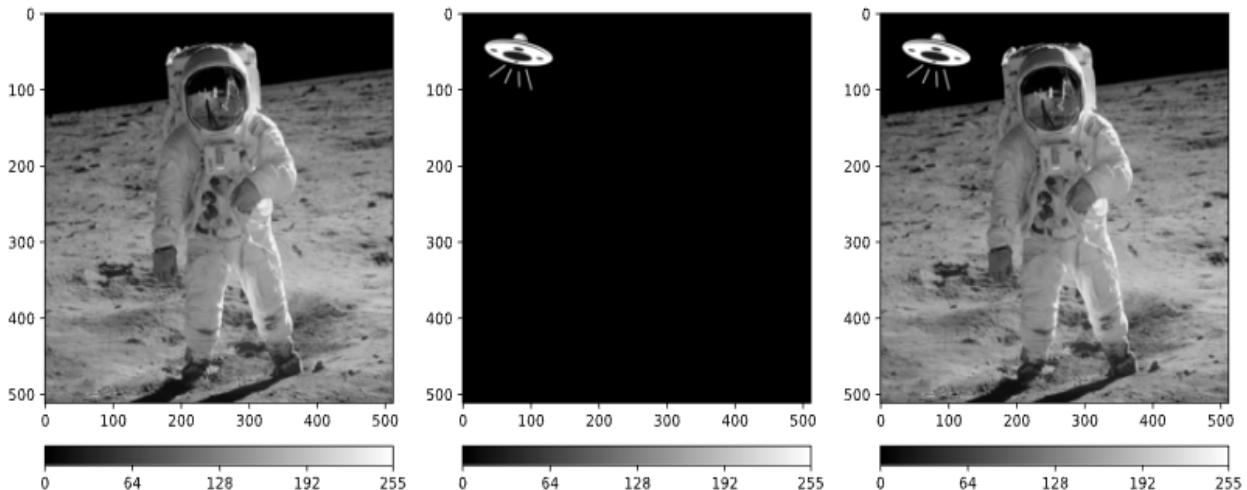


FIGURE 2.3 – Addition de deux images [91]

### 2.2.6.2 Soustraction d'images

La soustraction d'images est utilisée dans certaines applications pour mettre en évidence certains détails caractéristiques [72]. On peut définir la soustraction S pixel à pixel de deux images f et g par :

$$s = \max(f(x, y) - g(x, y), 0) \quad (2.2)$$

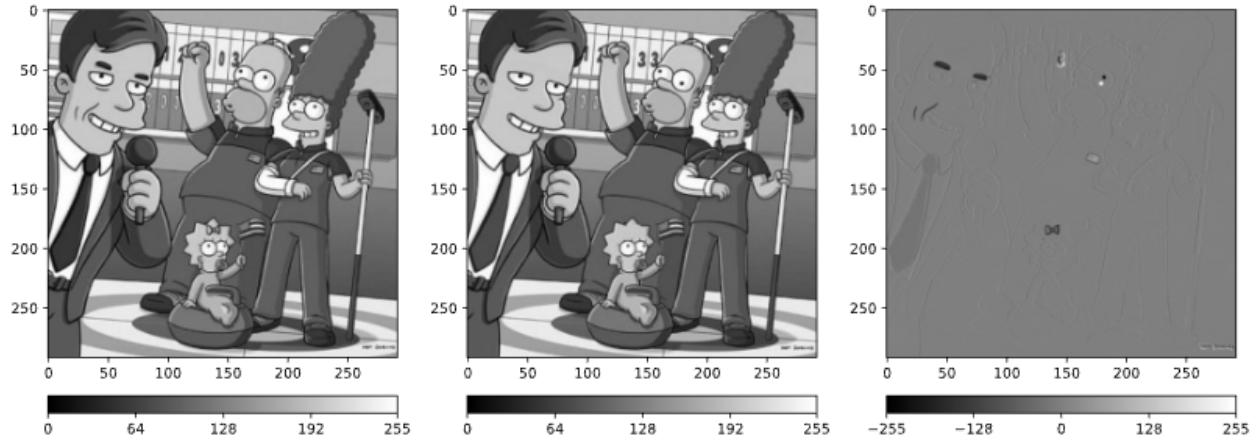


FIGURE 2.4 – Soustraction de deux images [91]

### 2.2.6.3 Division d'images

La division de deux images permet notamment de corriger une illumination non homogène [91], on peut la définir par :

$$\forall x, y \ d(x, y) = \frac{f(x, y)}{g(x, y)} \quad (2.3)$$

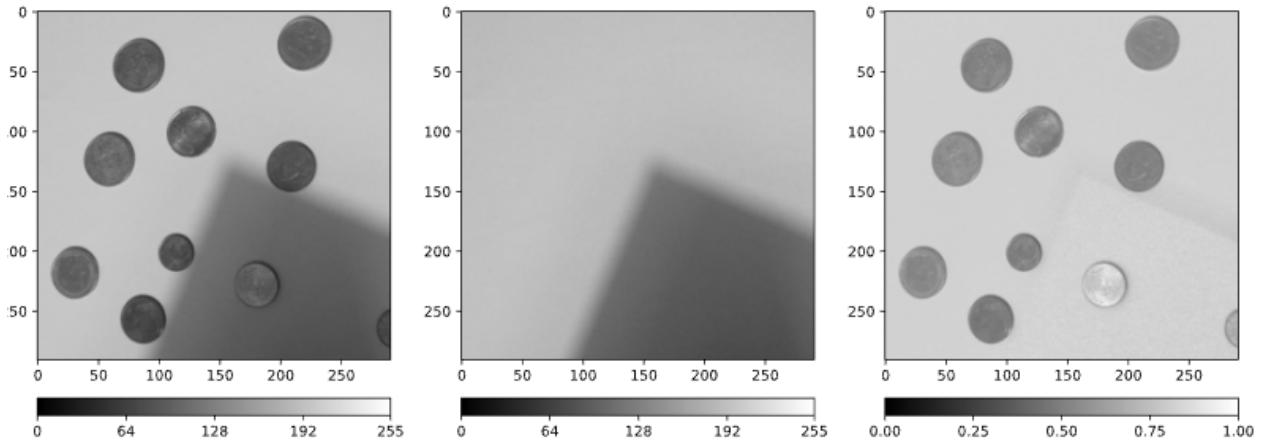


FIGURE 2.5 – Division de deux images [91]

### 2.2.6.4 Multiplication d'images

La multiplication se présente sous deux formes principales. La première forme prend deux images d'entrée et produit une image de sortie dans laquelle les valeurs des pixels sont simplement celles de la première image, multipliées par les valeurs des pixels correspondants dans la seconde image. La seconde forme prend une seule image en entrée et produit une image

en sortie dans laquelle chaque valeur de pixel est multipliée par une constante spécifiée. Cette dernière forme est probablement la plus largement utilisée et est généralement appelée mise à l'échelle au niveau de gris [112].

La multiplication de deux images est effectuée de manière évidente en un seul passage à l'aide de la formule :

$$m(x, y) = f(x, y) \times g(x, y) \quad (2.4)$$

La mise à l'échelle par une constante est effectuée en utilisant :

$$m(x, y) = f(x, y) \times c \quad (2.5)$$

Notez que la constante est souvent un nombre réel, et peut être inférieure à 1, ce qui réduira l'intensité des images. Elle peut même être négative si le format de l'image le permet.

### 2.2.7 Représentation des couleurs

Toute couleur peut être représentée comme une combinaison linéaire de trois primaires  $c_1, c_2, c_3$  qui définissent un espace des couleurs, on peut citer :

#### 2.2.7.1 L'espace RGB

L'espace RGB est l'espace vectoriel engendré par les 3 composantes primaires (Rouge, Vert, Bleu). C'est un modèle idéal pour expliquer l'addition de la luminosité dans les mélanges de couleurs, encore appelé la synthèse additive [72]. La représentation des couleurs dans cet espace donne un cube appelé cube de Maxwell.

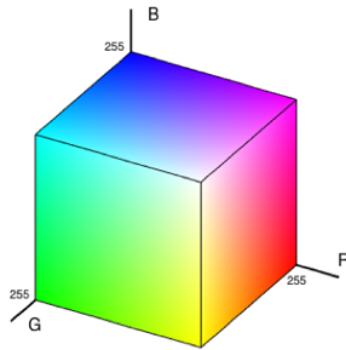


FIGURE 2.6 – Cube de Maxwell [120]

#### 2.2.7.2 L'espace CIE XYZ

Pour éviter les poids négatifs, la CIE a mis en oeuvre un système dérivé de RGB, appelé XYZ, où tous les poids sont positifs. Il est défini à partir d'une transformation linéaire sur l'espace l'espace RGB telle que toutes les couleurs couleurs du spectre visible soient contenues dans le triangle xyz. Les coordonnées XYZ des couleurs naturelles ont ainsi toujours des valeurs positives [72].

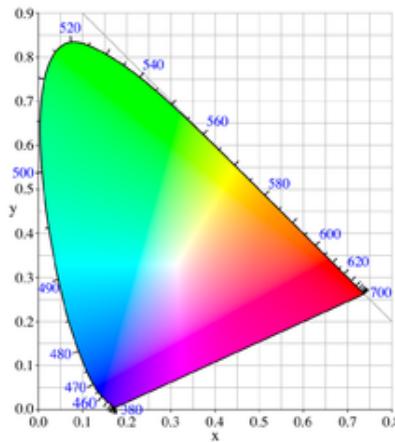


FIGURE 2.7 – L'espace XYZ [95]

### 2.2.7.3 L'espace HSV

Le principe de l'espace HSV (Hue, Saturation, Value), est de caractériser les couleurs de façon plus intuitive, conformément à la perception naturelle des couleurs [72], en termes de :

- **Hue** : Couleur.
- **Saturation** : Pureté.
- **Value** : Luminosité.

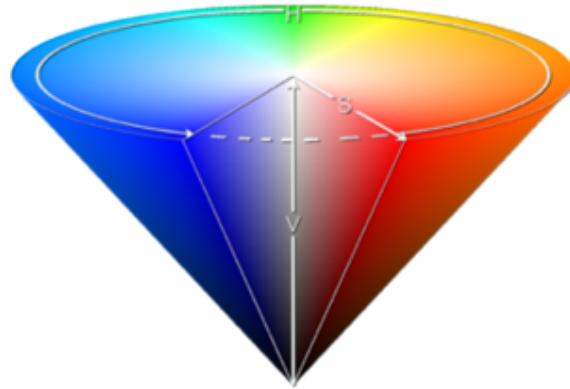


FIGURE 2.8 – L'espace HSV [120]

### 2.2.8 Les unités Hounsfield (HU)

Les unités Hounsfield (**Hounsfield units HU**) sont une unité sans dimension universellement utilisée en tomographie assistée par ordinateur (CT) pour exprimer les chiffres du scan sous une forme normalisée et pratique. Les unités Hounsfield sont obtenues à partir d'une transformation linéaire des coefficients d'atténuation mesurés. Il en résulte une échelle allant de -1000 HU pour l'air à + 2000 HU pour les os très denses et plus de 3000 pour les métaux [23].

Substance	HU
Air	-1000
Lung	-500
Fat	-100 to -50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (dense bone)

FIGURE 2.9 – L'échelle de Hounsfield [10]

## 2.2.9 Filtrage

Le filtrage en traitement d'images est une notion très précise qui se distingue des filtres proposés par certaines applications ou logiciels. Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine [111].

### 2.2.9.1 Les filtres passe-bas

Ils consistent à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Les coefficients du masque sont positifs et leur somme est égale à 1.

#### 2.2.9.1.1 Les filtres linéaires

Ils utilisent une combinaison linéaire des valeurs du voisinage pour déterminer le niveau de gris d'un pixel en utilisant des tailles (3x3, 5x5, ...). L'effet de flou est d'autant plus marqué que la taille est grande [72]. Nous pouvons citer :

- **Filtre moyenneur** : Le filtrage moyen est une méthode simple, intuitive et facile à mettre en œuvre pour lisser les images, c'est-à-dire réduire la quantité de variation d'intensité entre un pixel et le suivant. Il est souvent utilisé pour réduire le bruit dans les images . Il est mis en œuvre par une opération de calcul de la moyenne locale où la valeur de chaque pixel est remplacée par la moyenne de toutes les valeurs dans le voisinage [106].



FIGURE 2.10 – Exemple du filtre moyenneur [106]

- **Filtre gaussien :** L'opérateur de lissage gaussien est un opérateur de convolution 2D qui est utilisé pour "flouter" les images et supprimer les détails et le bruit. Il utilise un noyau qui représente la forme gaussienne (en forme de cloche) [72]. La largeur du filtre est donnée par son écart-type  $\sigma$  :
  - Si  $\sigma$  est plus petit qu'un pixel le lissage n'a presque pas d'effet.
  - Plus  $\sigma$  est grand, plus on réduit, mais plus l'image filtrée est floue.
  - Si  $\sigma$  est choisi trop grand, tous les détails de l'image sont perdus.

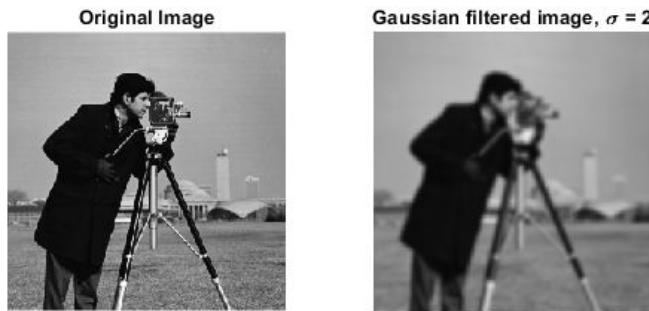


FIGURE 2.11 – Exemple du filtre Gaussien [54]

### 2.2.9.1.2 Les filtres non linéaires

Les filtres linéaires ont la propriété que leur sortie est une combinaison linéaire de leur entrée. Les filtres qui ne satisfont pas cette propriété sont non linéaires. L'idée générale dans le filtrage d'image non linéaire est qu'au lieu d'utiliser le masque spatial dans un processus de convolution, le masque est utilisé pour obtenir les valeurs de pixels voisins, puis les mécanismes de commande produisent le pixel de sortie [109]. Nous pouvons citer :

- **Filtre médian :** Le filtre médian est généralement utilisé pour réduire le bruit poivre et sel dans une image, un peu comme le filtre moyen. Cependant, il fait souvent un meilleur travail que le filtre moyen pour préserver les détails utiles de l'image car il ne dépend pas des valeurs qui sont significativement différentes des valeurs typiques du voisinage. L'idée principale est de classer les pixels voisins par ordre croissant des valeurs de niveaux de gris, puis affecter la valeur médiane au pixel central [72].

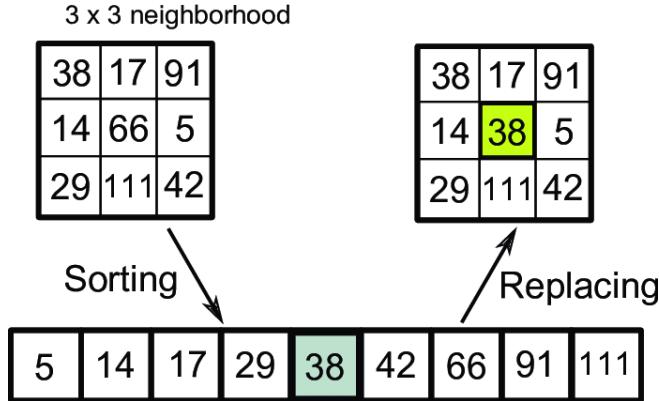


FIGURE 2.12 – Méthode du filtre médian

- **Filtre morphologique** : Contrairement aux filtres définis précédemment, les filtres morphologiques sont définis par des propriétés algébriques : un filtre est un opérateur croissant et idempotent [92]. La valeur de chaque pixel dans l'image de sortie est basée sur une comparaison du pixel correspondant dans l'image d'entrée avec ses voisins. Il sont basés sur deux opérations :
  - **Dilatation** : effectue un « ou » logique entre les voisins d'un pixel (augmente l'épaisseur d'un contour).
  - **L'érosion** : effectue un « et » logique entre les voisins d'un pixel (diminue le contour d'un pixel).
- En combinant ces deux opérations élémentaires, autres opérations peuvent être créées, notamment :
  - **Ouverture(opening)** : érosion suivie d'une dilatation sur le même noyau.

$$A \circ B = (A \ominus B) \oplus B \quad (2.6)$$

- **Fermeture (closing)** : dilatation suivie d'une érosion sur le même noyau.

$$A \bullet B = (A \oplus B) \ominus B \quad (2.7)$$

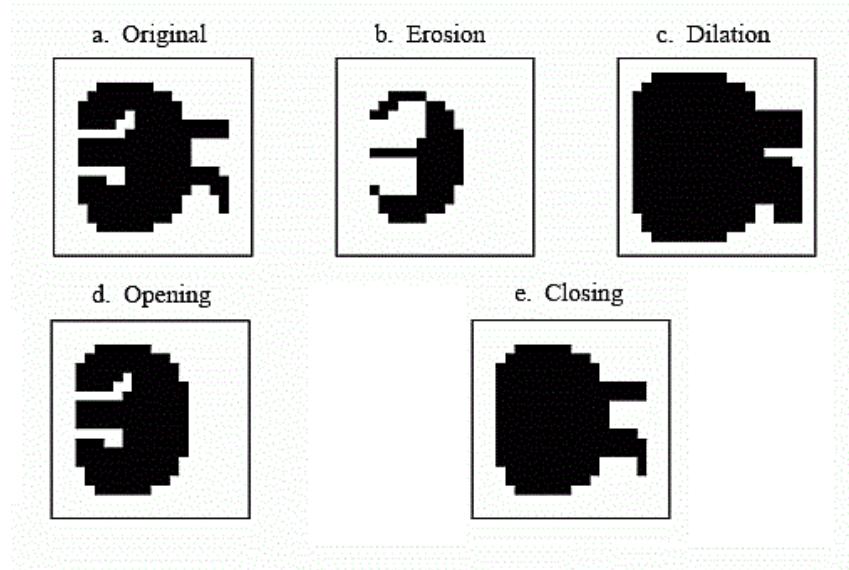


FIGURE 2.13 – Exemple du différents filtres morphologiques.

### 2.2.9.2 Les filtres passe-haut

A l'inverse des filtres passe-bas, ils atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails. Les coefficients du masque peuvent être de signe quelconque mais leur somme est égale à 0 [72].

#### 2.2.9.2.1 Le gradient

C'est la dérivée première de la fonction de niveau de gris, le contour correspond à la norme du gradient supérieure à un seuil donné . Les contours correspondent aux maxima de la première dérivée [72].

#### 2.2.9.2.2 Laplacien

C'est la dérivée seconde de la fonction de niveau de gris, le contour correspond au passage par zéro du Laplacien. Les contours correspondent aux passages par zéros de la deuxième dérivée [72].

### 2.2.10 Segmentation

La segmentation d'image est une étape clé en traitement d'images qui a pour but de regrouper des pixels semblables entre eux suivant des critères : couleur, texture, niveaux de gris et mouvement. Les pixels sont regroupés en régions, qui constituent une partition de l'image. Chaque partition(région) est supposée correspondre à un "objet" de l'image. L'union de ces régions doit redonner l'image initiale [72].

Dans l'imagerie médicale, la segmentation ce concentre sur la discrimination entre les structures anatomiques saines et les tissus pathologiques tel que les nodules.

#### 2.2.10.1 Approche basée région

##### 2.2.10.1.1 Méthode de seuillage

La segmentation par seuillage est l'opération qui associe à chaque point de l'image une classe de luminance après comparaison de ses attributs "niveaux de gris" à un ou plusieurs seuils. Une étiquette est alors affectée à chacun des points en fonction de la classe d'appartenance [1].

##### 2.2.10.1.2 Méthode de croissance de région (Region growing)

Appelée aussi croissance en volume en 3D,est une approche ascendante.Nous commençons avec un pixel(ou bien un groupe de pixels) germe, de départ, puis vérifions les pixels adjacents. Si les pixels adjacents respectent les règles prédéfinies(seuillage de préférence), alors ce pixel est ajouté à la région du pixel de départ et le processus suivant se poursuit jusqu'à ce qu'il ne reste plus de similitude [76].

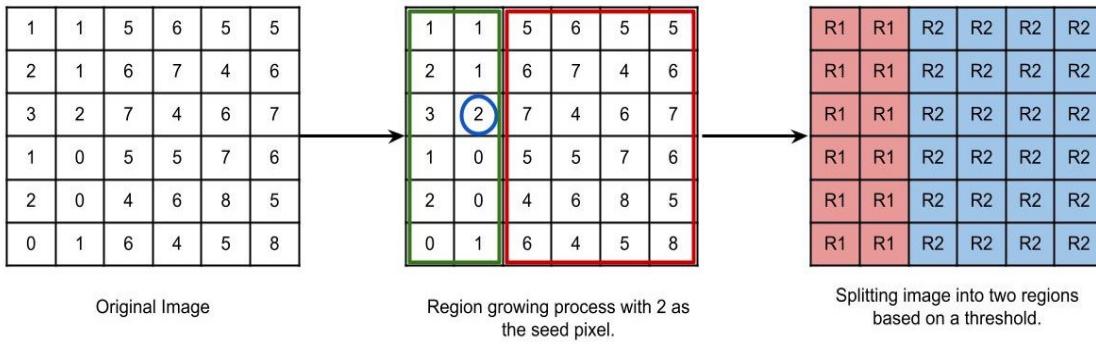


FIGURE 2.14 – Exemple de la méthode Region growing.

#### 2.2.10.1.3 Méthode de division et fusion

C'est une approche descendante. Le processus consiste à décomposer l'image de manière récursive jusqu'à ce que toutes les régions satisfassent à un critère d'homogénéité, puis toutes les régions adjacentes satisfaisant à un deuxième critère d'homogénéité sont fusionnées [76].

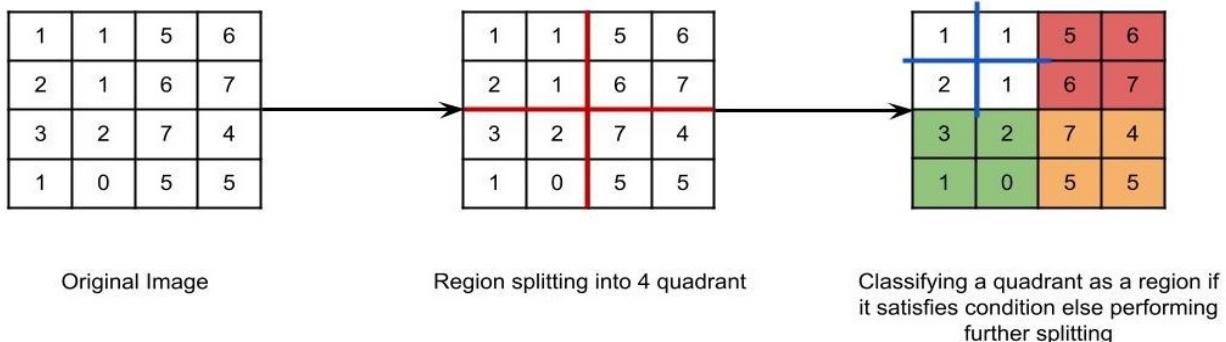


FIGURE 2.15 – Exemple de la méthode division et fusion.

#### 2.2.10.2 Approche basée countour

##### 2.2.10.2.1 Méthode de Sobel

Le filtre Sobel calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords [46].

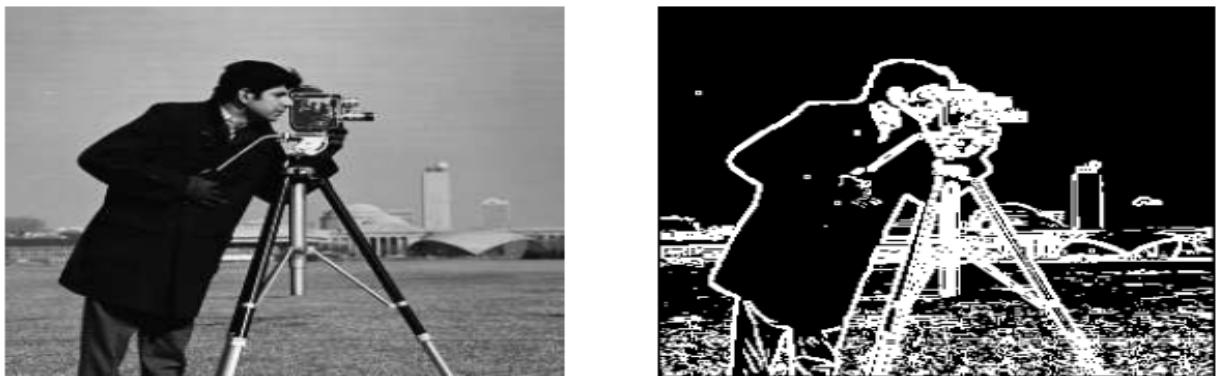


FIGURE 2.16 – Exemple d'application du filtre sobel [18]

### 2.2.10.2.2 Méthode de Canny

La méthode de Canny commence d'abord par le lissage de l'image avec un filtre gaussien, puis applique l'opérateur de gradient de la première différence pour calculer la magnitude et la direction du contour, suivi d'une suppression non-maximale de la magnitude du gradient et en fin un seuillage à l'image obtenue après la suppression [16].

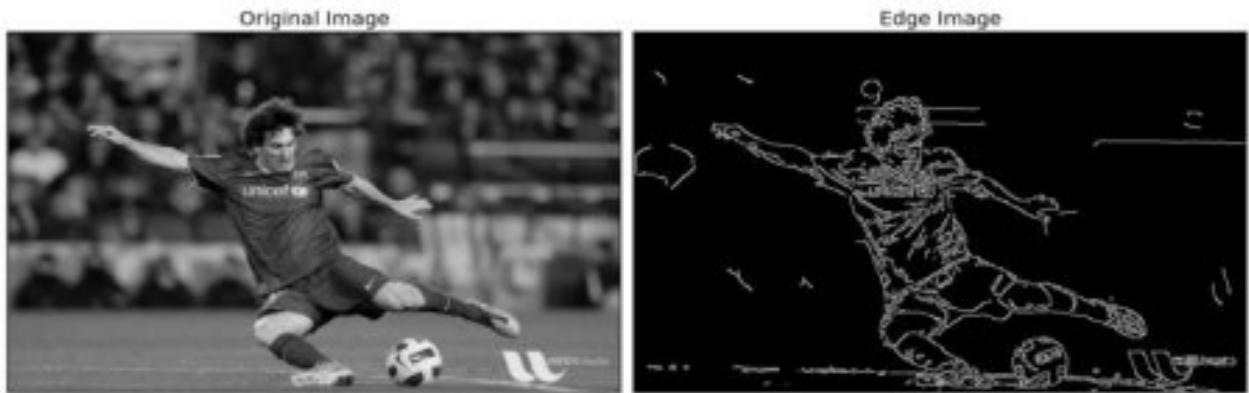


FIGURE 2.17 – Exemple d'application du filtre canny [110]

### 2.2.10.3 Approche hybride

#### 2.2.10.3.1 Ligne des partages des eaux (Water-shed)

Un outil de segmentation issu de la morphologie mathématique, qui permet de partitionner une image en plusieurs régions selon une approche topographique : on recherche les régions séparées par les lignes de crêtes de l'image. Différents types d'algorithmes pour que ça marche,citons : inondation,immersion... etc [94].

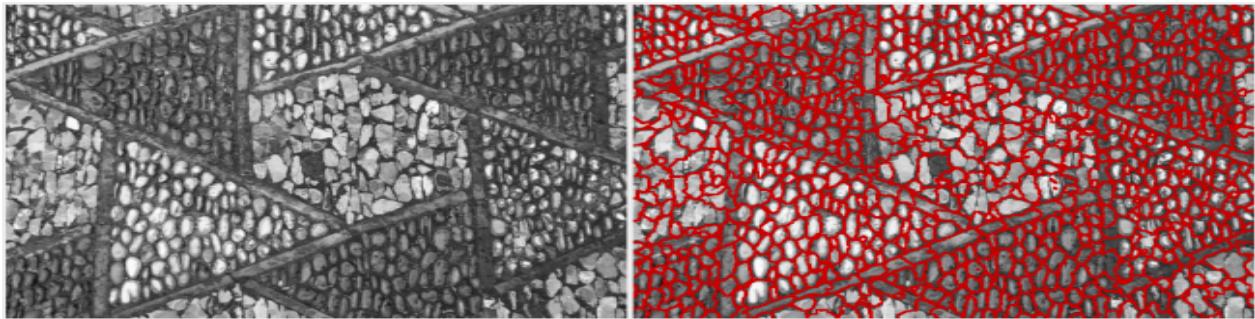


FIGURE 2.18 – Exemple de la méthode Water-shed

#### 2.2.10.3.2 Contour actif (Snake)

Un modèle de contour actif est formé d'un enchaînement de points mobiles et répartis sur une courbe en deux dimensions. La courbe qui peut être fermée, est placée dans la zone d'intérêt de l'image ou autour d'un objet [35].



FIGURE 2.19 – Exemple de la méthode Snake [6]

## 2.3 Apprentissage profond

### 2.3.1 Définition

L'apprentissage profond (**en anglais Deep learning**) est un type d'intelligence artificielle dérivé de l'apprentissage automatique (**machine learning**) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles pré-déterminées. Dans ce dernier, un algorithme informatique apprend à effectuer des tâches de classification directement sur des données complexes sous forme d'images, de textes ou de sons. Ces algorithmes peuvent atteindre une précision de pointe, et même parfois dépasser les performances humaines. Ils sont entraînés à l'aide d'un vaste ensemble de données étiquetées et d'architectures de réseaux neuronaux comprenant de nombreuses couches [55].

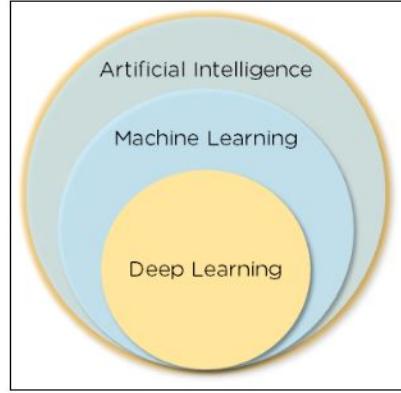


FIGURE 2.20 – La relation entre l'apprentissage profond, apprentissage automatique et l'intelligence artificielle [86]

### 2.3.2 Réseaux de neurones artificielles (ANN)

Un réseau de neurones artificielle (**en anglais Artificial Neural Network ANN**) se compose d'une couche d'entrée de neurones (ou nœuds, unités), d'une ou deux (voire trois) couches intermédiaires de neurones et d'une couche finale de neurones de sortie. Chaque nœud, ou neurone artificiel, se connecte à un autre et possède un poids et un seuil associés. Si la sortie d'un nœud individuel est supérieure à la valeur seuil spécifiée, ce nœud est activé en utilisant une fonction d'activation, envoyant des données à la couche suivante du réseau. Dans le cas contraire, aucune donnée n'est transmise à la couche suivante du réseau. Ces derniers reposent sur des données d'apprentissage pour apprendre et améliorer leur précision au fil du temps. Cependant, une fois que ces algorithmes d'apprentissage sont réglés avec précision, ils constituent des outils puissants en informatique et en intelligence artificielle, nous permettant de classer et de regrouper des données à une vitesse élevée [5].

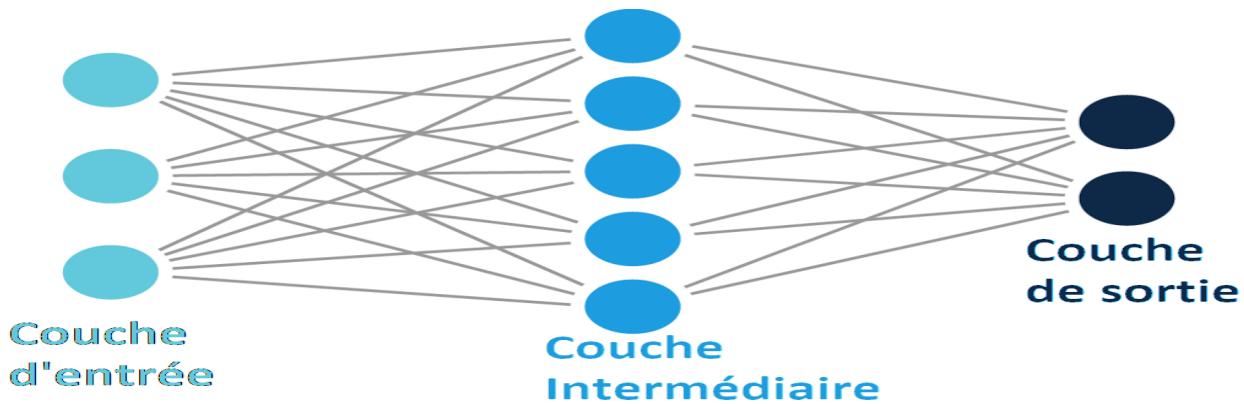


FIGURE 2.21 – Exemple d'un simple réseau de neurones

### 2.3.3 Réseaux de neurones convolutifs (CNN)

#### 2.3.3.1 Définition

Les réseaux de neurones convolutifs (**CNN**) sont analogues aux réseaux ANN traditionnels dans la mesure où ils sont composés de neurones qui s'auto-optimisent par apprentissage. Chaque neurone recevra toujours une entrée et effectuera une opération (comme un produit scalaire suivi d'une fonction non linéaire) - la base des innombrables ANN. Depuis les vecteurs

d'image bruts en entrée jusqu'à la sortie finale du score de classe, l'ensemble du réseau exprimera toujours une seule fonction de score perceptif (le poids). La dernière couche contiendra les fonctions de perte associées aux classes, et tous les conseils et astuces développés pour les ANN traditionnels restent applicables. La seule différence notable entre les CNN et les ANN traditionnels est que les CNN sont principalement utilisés dans le domaine de la reconnaissance des formes dans les images. Cela va permettre d'encoder des caractéristiques spécifiques à l'image dans l'architecture, ce qui rend le réseau plus adapté aux tâches axées sur l'image - tout en réduisant davantage les paramètres requis pour configurer le modèle [27].

### 2.3.3.2 Structure générale

Les CNN sont composés de trois types de couches en plus de la couche d'entrée et de sortie. Ce sont : les couches convolutionnelles, les couches de pooling et les couches entièrement connectées. Lorsque ces couches sont empilées, une architecture CNN est formée. Nous détaillons chaque couche :

- **La couche d'entrée :** Contiendra les valeurs des pixels de l'image.
- **Les couches convolutionnelles :** Va déterminer la sortie des neurones qui sont connectés à des régions locales de l'entrée par le calcul du produit scalaire entre leurs poids et la région connectée au volume d'entrée. L'unité linéaire rectifiée (Connue sous le nom de ReLu) a pour but d'appliquer une fonction d'activation "par éléments", telle que la sigmoïde, à la sortie de l'activation produite par la couche précédente.

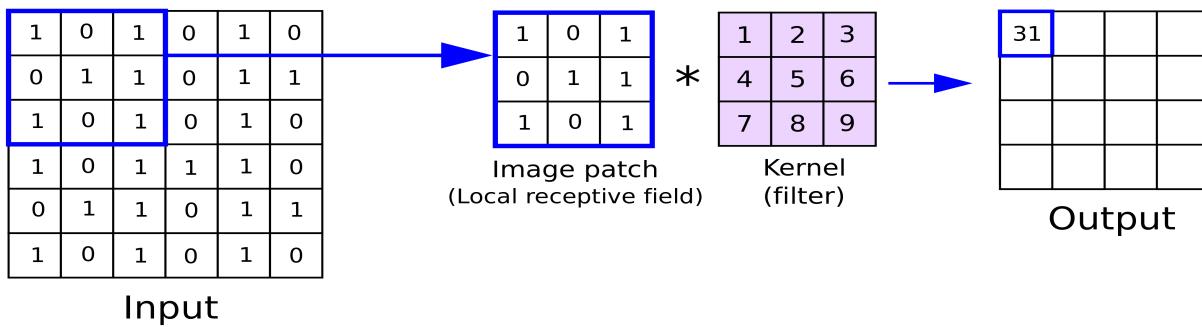


FIGURE 2.22 – Exemple d'une opération de convolution [41]

- **Les couches pooling :** Vont simplement effectuer un sous-échantillonnage le long de la dimension spatiale de l'entrée donnée, réduisant encore le nombre de paramètres dans cette activation.

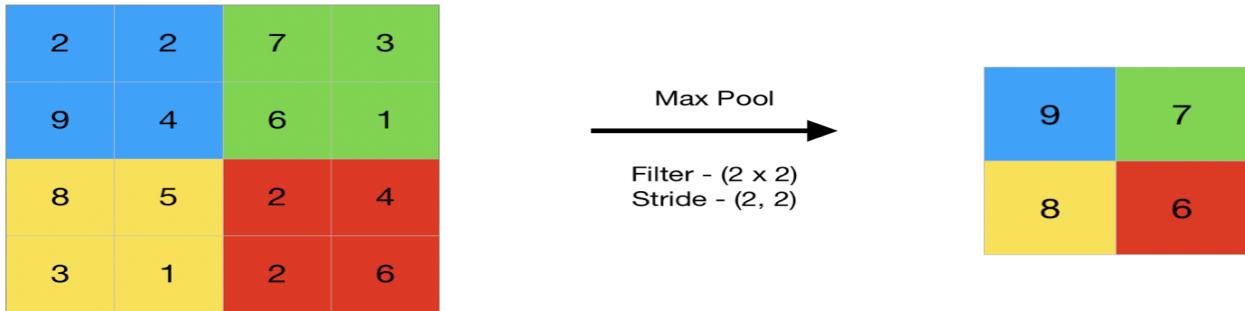


FIGURE 2.23 – Exemple d'un max pooling [78]

- **Les couches entièrement connectées** : effectueront ensuite les mêmes tâches que les ANN standard et tenteront de produire des scores de classe à partir des activations, à utiliser pour la classification. Il est également suggéré que ReLu puisse être utilisé entre ces couches, afin d'améliorer les performances.
- **La couche de sortie** : Permet de conserver le résultat final de l'image traitée.

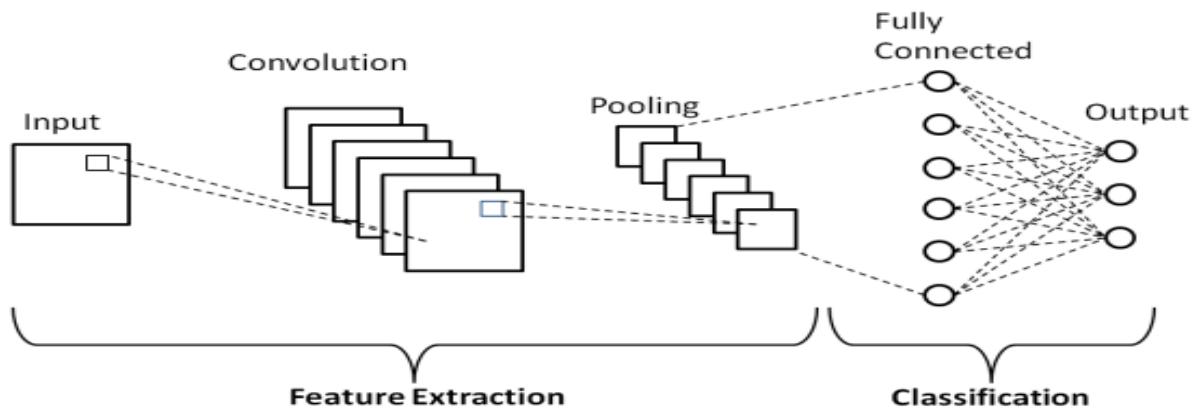


FIGURE 2.24 – Une architecture simple d'un CNN [61]

### 2.3.3.3 Les fonctions d'activations

Une fonction d'activation décide si un neurone doit être activé ou non. Cela signifie qu'elle décidera si l'entrée du neurone dans le réseau est importante ou non dans le processus de prédiction en utilisant des opérations mathématiques plus simples. Il existe 10 fonctions d'activation : fonction à pas binaire, linéaire, Sigmoid, Tanh, ReLU, Leaky ReLU, ReLU paramétré, ELU, Swish, SoftMax [65].

#### 2.3.3.3.1 Fonction à pas binaire

La fonction de pas binaire est la fonction d'activation la plus simple qui existe et elle peut être implémentée avec de simples instructions if-else en Python. Lors de la création d'un classificateur binaire, les fonctions d'activation binaires sont généralement utilisées. Mais, la fonction d'activation binaire ne peut pas être utilisée dans le cas d'une classification multi-classes dans une carrière cible. De plus, le gradient de la fonction binaire est égal à zéro, ce qui peut gêner l'étape de rétropropagation, c'est-à-dire que si nous calculons la dérivée de  $f(x)$  par rapport à  $x$ , elle est égale à zéro [65]. Mathématiquement, la fonction échelon binaire

peut être définie comme suit :

$$\begin{aligned} f(x) &= 1, \text{ si } x \geq 0 \\ f(x) &= 0, \text{ si } x < 0 \end{aligned} \quad (2.8)$$

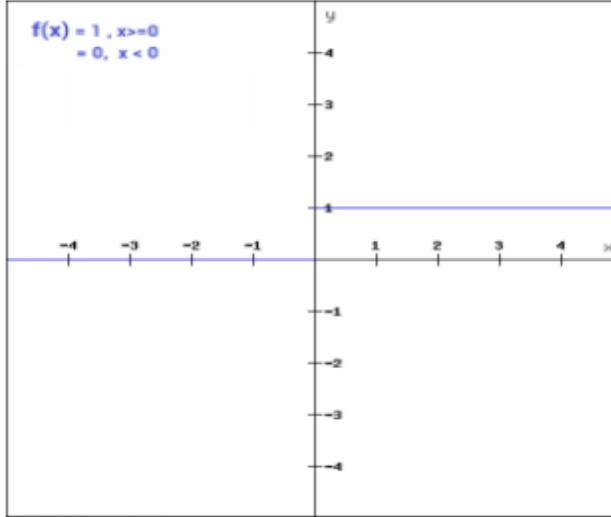


FIGURE 2.25 – La fonction à pas binaire

### 2.3.3.3.2 Linéaire

La fonction d'activation linéaire est directement proportionnelle à l'entrée. Le principal inconvénient de la fonction binaire est qu'elle a un gradient nul car il n'y a pas de composante en x dans la fonction binaire. Afin de supprimer cet inconvénient, une fonction linéaire peut être utilisée [65]. Elle est définie comme suit :

$$f(x) = ax \quad (2.9)$$

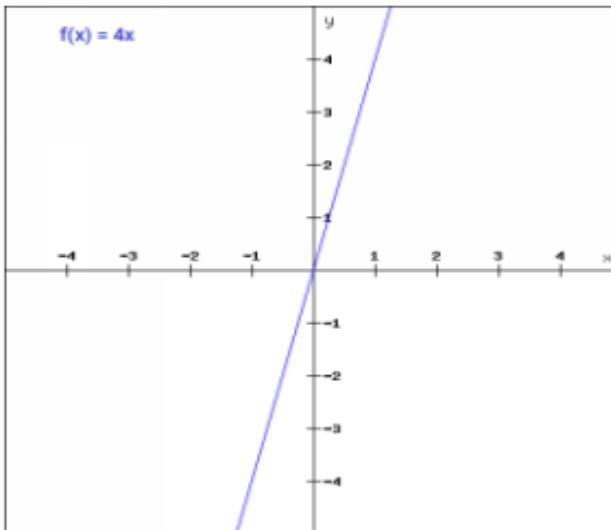


FIGURE 2.26 – La fonction linéaire

### 2.3.3.3.3 Sigmoid

La fonction Sigmoid donne une valeur de probabilité entre 0 et 1. Elle donc très utilisée pour les classifications binaires , lorsque un modèle doit déterminer entre deux labels [65]. Mathématiquement, elle peut être représentée comme suit :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

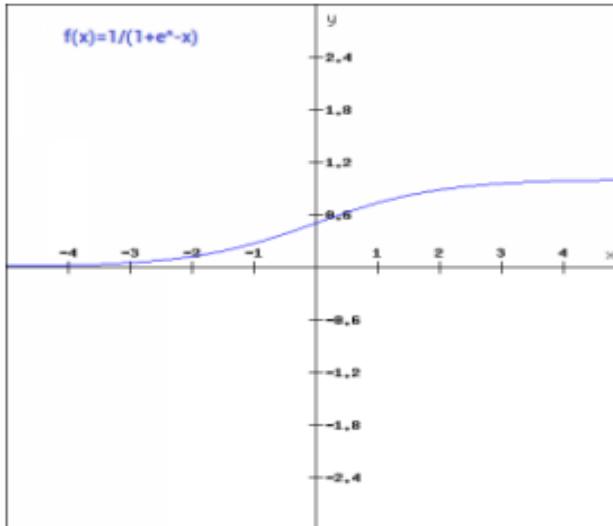


FIGURE 2.27 – La fonction Sigmoid

### 2.3.3.3.4 Tangente hyperbolique (Tanh)

La fonction Tanh est très similaire à la fonction d'activation sigmoïde, et à la même forme en "S", avec la différence que la plage de sortie va de -1 à 1. Dans la fonction Tanh, plus l'entrée est grande (plus positive), plus la valeur de sortie sera proche de 1,0, tandis que plus l'entrée est petite (plus négative), plus la sortie sera proche de -1,0 [65]. Mathématiquement, il peut être représenté comme suit :

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.11)$$

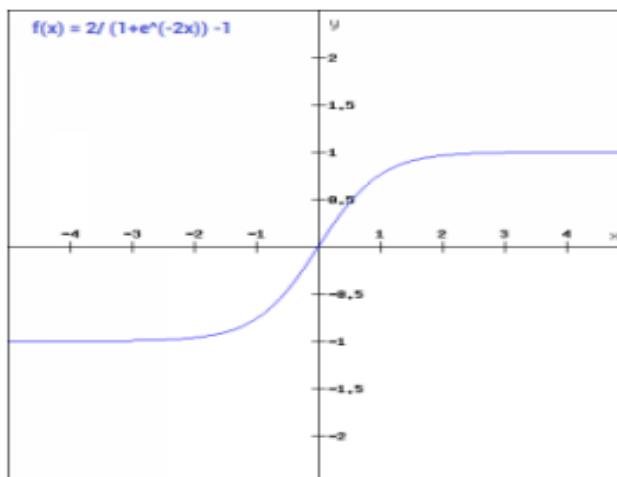


FIGURE 2.28 – La fonction Tanh

### 2.3.3.3.5 Unités linéaires rectifiées (ReLU)

La fonction ReLU est la fonction d'activation la plus simple et la plus utilisée. Elle donne  $x$  si  $x$  est supérieur à 0 , sinon 0. Autrement dit, c'est le maximum entre  $x$  et 0 [65]. Mathématiquement, il peut être représenté comme suit :

$$f(x) = \max(0, x) \quad (2.12)$$

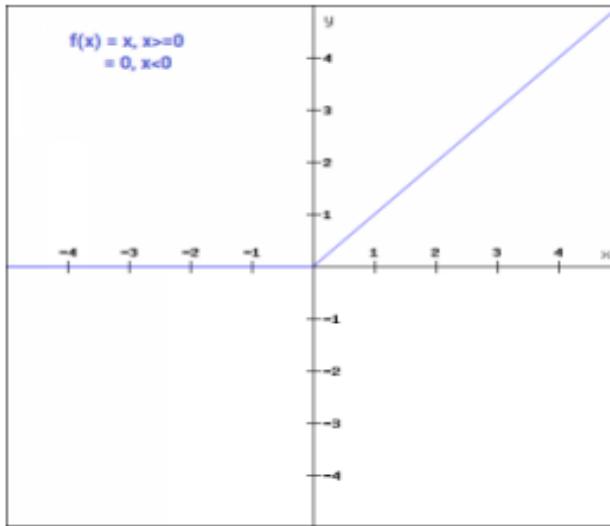


FIGURE 2.29 – La fonction ReLU

### 2.3.3.3.6 Leaky ReLU

Leaky ReLU, est un type de fonction d'activation basé sur un ReLU, mais il a une petite pente pour les valeurs négatives au lieu d'une pente plate. Le coefficient de pente est déterminé avant la formation, c'est-à-dire qu'il n'est pas appris pendant l'entraînement. Ce type de fonction d'activation est populaire dans les tâches où nous pouvons souffrir de gradients éparses, par exemple la formation de réseaux adversaires génératifs [65]. Mathématiquement, elle peut être représentée comme suit :

$$f(x) = \max(0.1x, x) \quad (2.13)$$

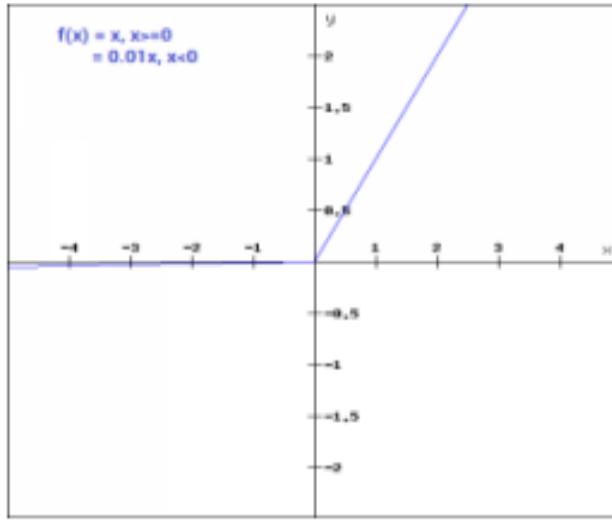


FIGURE 2.30 – La fonction Leaky ReLU

#### 2.3.3.3.7 ReLU paramétré

Il s'agit également d'une variante de l'Unité Linéaire Rectifiée avec de meilleures performances et une légère variation. Elle résout le problème du gradient de la ReLU qui devient nul pour des valeurs négatives de  $x$  en introduisant un nouveau paramètre de la partie négative de la fonction, c'est-à-dire la pente [65]. Elle est exprimée comme suit :

$$\begin{aligned} f(x) &= x, \text{ si } x \geq 0 \\ f(x) &= ax, \text{ si } x < 0 \end{aligned} \tag{2.14}$$

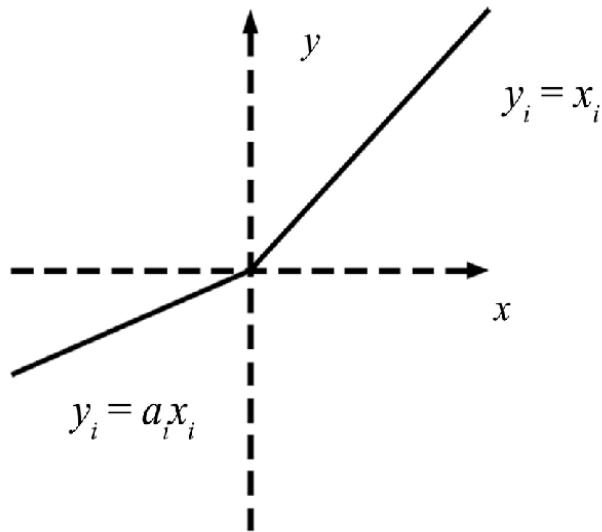


FIGURE 2.31 – La fonction ReLU paramétré

#### 2.3.3.3.8 Unité linéaire exponentielle (ELU)

L'unité linéaire exponentielle ou ELU est aussi une variante de l'unité linéaire rectifiée. ELU introduit un paramètre pente pour les valeurs négatives de  $x$ . Elle utilise une courbe

logarithmique pour définir les valeurs négatives [65] :

$$\begin{aligned} f(x) &= x, \text{ si } x \geq 0 \\ f(x) &= \alpha(e^x - 1), \text{ si } x < 0 \end{aligned} \quad (2.15)$$

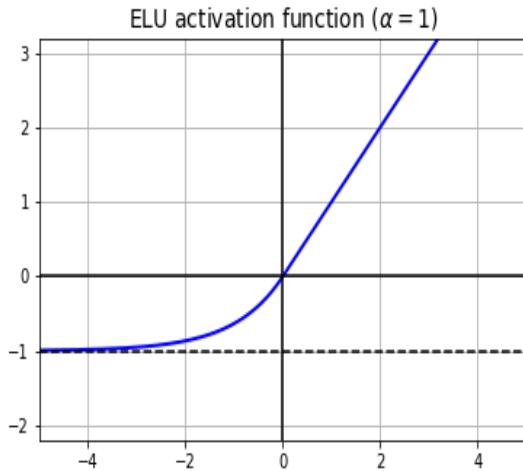


FIGURE 2.32 – La fonction ELU

#### 2.3.3.3.9 Swish

La fonction Swish est une fonction d’activation relativement nouvelle qui a été découverte par des chercheurs de Google. La caractéristique distinctive de la fonction Swish est qu’elle est monotone, ce qui signifie que la valeur de la fonction peut diminuer même si les valeurs des entrées augmentent. Dans certains cas, Swish surpassé même la fonction ReLU. Elle s’exprime mathématiquement comme suit [65] :

$$\begin{aligned} f(x) &= x \times \text{sigmoid}(x) \\ &= \frac{x}{1 - e^{-x}} \end{aligned} \quad (2.16)$$

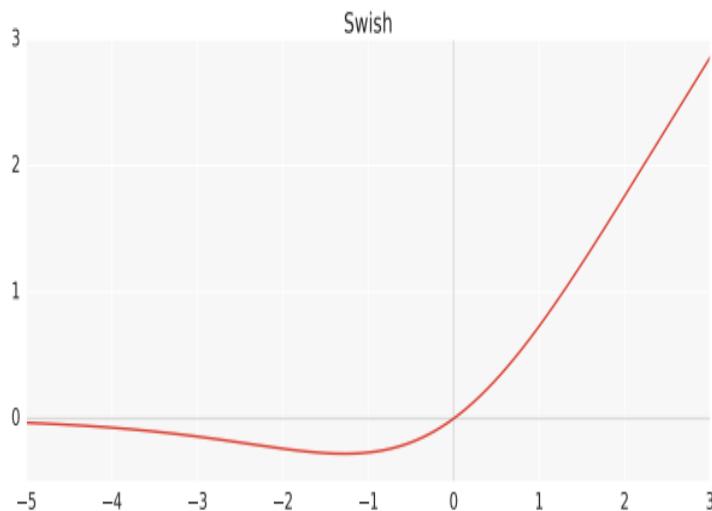


FIGURE 2.33 – La fonction Swish

### 2.3.3.3.10 SoftMax

La fonction softmax permet de transformer un vecteur réel en vecteur de probabilité. On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multiclass. Dans la fonction softmax, chaque vecteur est traité indépendamment. L'argument axis définit l'axe d'entrée sur lequel la fonction est appliquée [65]. Voici la formule :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ pour } i = 1, \dots, K \text{ et } z = (z_1, \dots, z_K) \in \mathbb{R}^K \quad (2.17)$$

### 2.3.3.4 Technique de régularisation

Pour les modèles CNN, le over-fitting représente le problème central associé à l'obtention d'une généralisation bien conduite. Le modèle est qualifié de over-fitted dans les cas où le modèle fonctionne particulièrement bien sur les données d'entraînement et ne réussit pas sur les données de test (données non vues). Un modèle under-fitted est le contraire ; ce cas se produit lorsque le modèle n'apprend pas suffisamment à partir des données d'apprentissage. Le modèle est dit "juste ajusté" s'il fonctionne bien à la fois sur les données d'apprentissage et de test. Divers concepts intuitifs sont utilisés pour aider la régularisation à éviter le over-fitting et under-fitting [70].

#### 2.3.3.4.1 Dropout

Il s'agit d'une technique largement utilisée pour la généralisation. Pendant chaque période d'entraînement, les neurones sont éliminés de manière aléatoire. Ainsi, le pouvoir de sélection des caractéristiques est réparti de manière égale sur l'ensemble du groupe de neurones, ce qui oblige le modèle à apprendre différentes caractéristiques indépendantes. Au cours du processus d'entraînement, le neurone abandonné ne fera pas partie de la micropropagation inverse ou de la micropropagation directe. En revanche, le réseau complet est utilisé pour effectuer des prédictions pendant le processus de test [70].

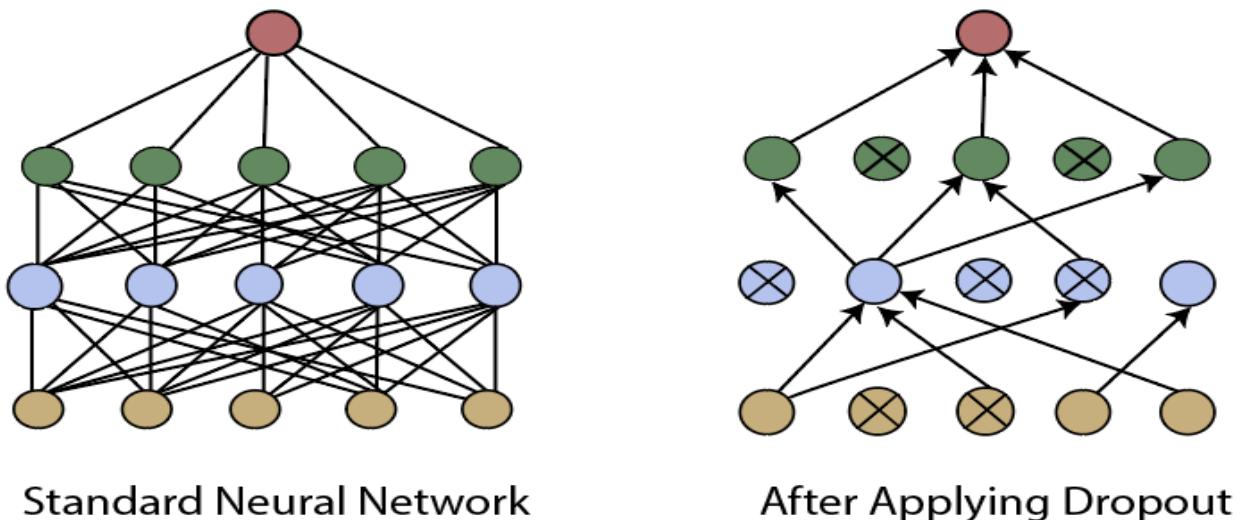


FIGURE 2.34 – Exemple d'application du Dropout [80]

### 2.3.3.4.2 Drop-Weights

Cette méthode est très similaire au dropout. À chaque époque d'apprentissage, les connexions entre les neurones (poids) sont supprimées au lieu de supprimer les neurones ; c'est la seule différence entre le drop-weights et le dropout [70].

### 2.3.3.4.3 Augmentation des données

L'entraînement du modèle sur une quantité importante de données est le moyen le plus simple d'éviter l'over-fitting. Pour y parvenir, on utilise l'augmentation des données. Plusieurs techniques sont utilisées pour augmenter artificiellement la taille de l'ensemble de données d'entraînement [70]. Voici quelques exemples de techniques utilisées pour augmenter les données :

- Ajouter du bruit
- Recadrage (Cropping)
- inversement (Flipping)
- Rotation
- Mise à l'échelle
- Translation
- Luminosité
- Contraste
- Augmentation de la couleur
- Saturation

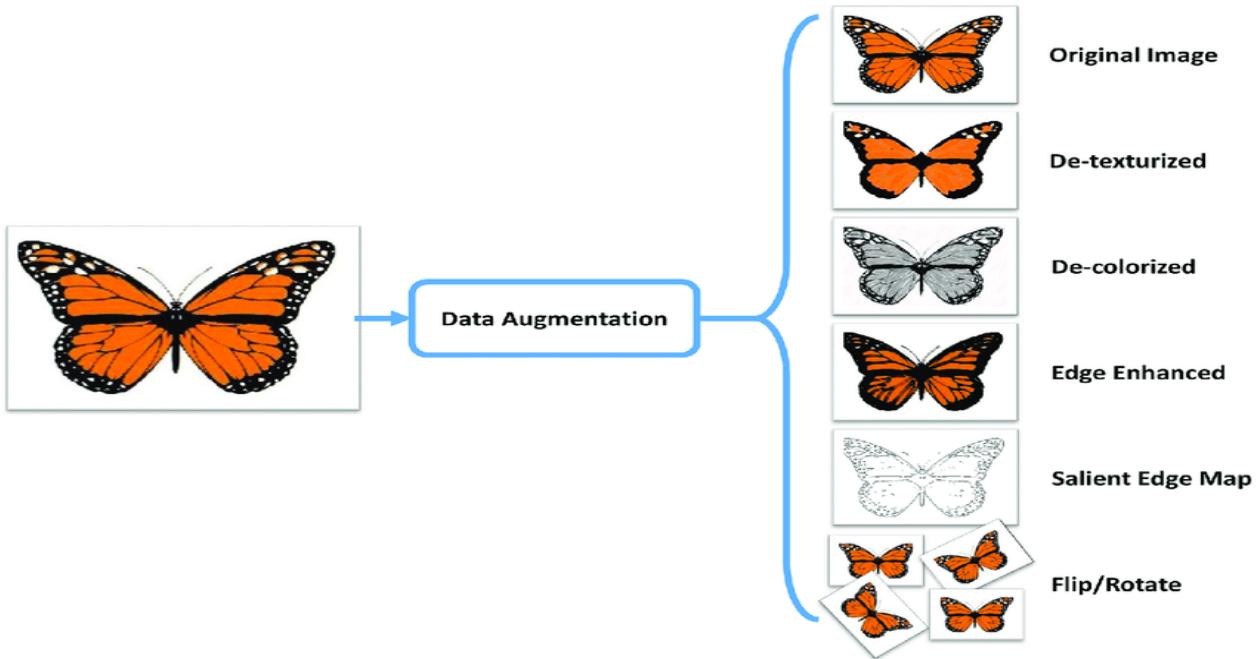


FIGURE 2.35 – Exemple d'application de l'augmentation de données [64]

### 2.3.3.4.4 Batch Normalization

Cette méthode garantit la performance des activations de sortie. Cette performance suit une distribution gaussienne unitaire. En soustrayant la moyenne et en divisant par l'écart-type, on normalise la sortie à chaque couche. S'il est possible de la considérer comme une tâche de prétraitement à chaque couche du réseau, il est également possible de la différencier

et de l'intégrer à d'autres réseaux. En outre, elle est employée pour réduire le "décalage de covariance interne" des couches d'activation. Dans chaque couche, la variation de la distribution de l'activation définit le décalage de covariance interne. Ce décalage devient très élevé en raison de la mise à jour continue des poids au cours de l'apprentissage, ce qui peut se produire si les échantillons des données d'apprentissage proviennent de nombreuses sources différentes (par exemple, des images de jour et de nuit). Ainsi, le modèle consommera un temps supplémentaire pour la convergence, et à son tour, le temps requis pour la formation augmentera également. Pour résoudre ce problème, une couche représentant l'opération de normalisation par lots est appliquée dans l'architecture CNN [70].

### 2.3.3.5 Les architectures du CNN

Au cours des 10 dernières années, plusieurs architectures CNN ont été présentées. L'architecture des modèles est un facteur critique pour améliorer les performances de différentes applications. De 1989 à aujourd'hui, diverses modifications ont été apportées à l'architecture CNN. Ces modifications comprennent la reformulation structurelle, la régularisation, l'optimisation des paramètres, etc. Inversement, il convient de noter que l'amélioration essentielle des performances du CNN est due en grande partie à la réorganisation des unités de traitement, ainsi qu'au développement de nouveaux blocs. En particulier, les développements les plus novateurs dans les architectures CNN ont été réalisés sur l'utilisation de la profondeur du réseau [70]. Nous allons mentionner certaines des architectures que nous avons utilisées dans notre système.

#### 2.3.3.5.1 U-Net

Une architecture apparue en 2015 par des chercheurs allemands, elle est principalement utilisée pour la segmentation sémantique qui consiste essentiellement à segmenter et à classer en même temps (classification à classe unique ou multi-classes). 'U-Net est considérée comme l'une des meilleures solutions pour la segmentation des images biomédicales, il peut être entraînée de bout en bout à partir de très peu d'images et surpassé la meilleure méthode antérieure, ce qui est le cas dans le domaine médical, car il n'y a pas toujours beaucoup de données disponibles et, en même temps, la précision est nécessaire.

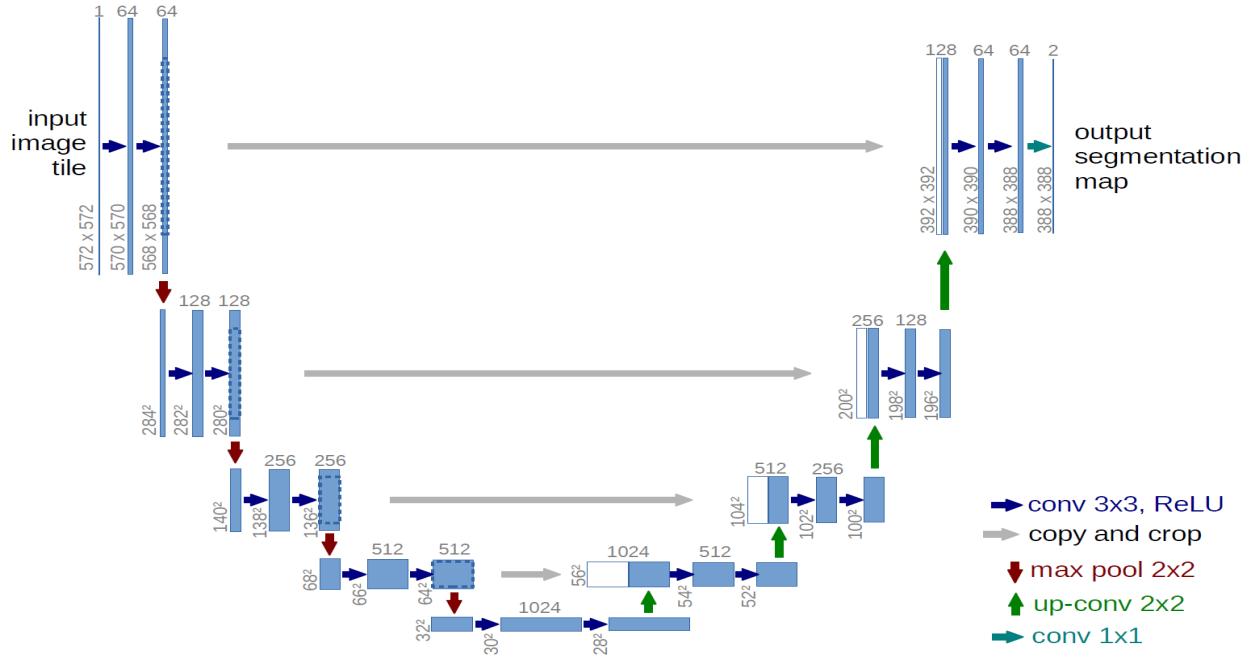


FIGURE 2.36 – Architecture U-net [28]

L'architecture du réseau U-NET présenté dans la figure 2.36 se compose d'un chemin contractant (encodeur) et d'un chemin expansif (décodeur). La voie de contraction, où le modèle apprend le "quoi", suit l'architecture typique d'un réseau convolutif. Il consiste en l'application répétée de deux convolutions  $3 \times 3$ , chacune suivie d'une unité linéaire rectifiée (ReLU) et d'une opération de pooling  $2 \times 2$  max avec un stride 2 pour le sous-échantillonnage. À chaque étape de sous-échantillonnage, le nombre de canaux caractéristiques est doublé. Le chemin expansif que le modèle élabore sur le "Où" ou l'information spatiale dans l'image ,consiste en un sur-échantillonnage de la carte de caractéristiques suivi d'une convolution  $2 \times 2$  qui divise par deux le nombre de canaux de caractéristiques, une concaténation avec la carte de caractéristiques recadrée correspondante du chemin de contraction, et deux convolutions  $3 \times 3$ , chacune suivie d'une ReLU. Le recadrage est nécessaire en raison de la perte de pixels de bordure dans chaque convolution. Dans la dernière couche, une convolution  $1 \times 1$  est utilisée pour mettre en correspondance chaque vecteur de caractéristiques à 64 composantes avec le nombre de classes souhaité. Au total, le réseau comporte 23 couches convolutionnelles. L'architecture u-net obtient de très bonnes performances sur des applications de segmentation biomédicale très différentes. Grâce à l'augmentation des données par des déformations élastiques, elle ne nécessite que très peu d'images annotées et a un temps d'apprentissage très raisonnable de seulement 10 heures sur un GPU NVidia Titan (6 Go) [28].

### 2.3.3.5.2 ResNet

He et al [25] ont développé ResNet (Residual Network), qui a été le gagnant de l'ILSVRC 2015. Leur objectif était de concevoir un réseau ultra-profond exempt du problème du gradient de fuite, par rapport aux réseaux précédents. Plusieurs types de ResNet ont été développés en fonction du nombre de couches (en commençant par 34 couches et en allant jusqu'à 1202 couches). Le type le plus courant était ResNet50, qui comprenait 49 couches convolutionnelles plus une seule couche FC. Le nombre total de poids du réseau était de 25,5 M, tandis que le nombre total de MAC était de 3,9 M. L'idée nouvelle de ResNet est son utilisation du concept de voie de contournement, qui a été employé dans Highway Nets pour résoudre le

problème de la formation d'un réseau plus profond en 2015. Ce concept qui contient le schéma fonctionnel fondamental de ResNet. Il s'agit d'un réseau feedforward classique auquel s'ajoute une connexion résiduelle. La sortie de la couche résiduelle peut être identifiée comme la  $(l - 1)$ ème sortie, qui provient de la couche précédente  $(x_{l-1})$ . Après avoir exécuté différentes opérations (comme la convolution à l'aide de filtres de taille variable, ou la normalisation par lots, avant d'appliquer une fonction d'activation comme ReLU sur  $(x_{l-1})$ ), la sortie est  $F(x_{l-1})$ . La sortie résiduelle finale est  $x_l$ , qui peut être représentée mathématiquement par :

$$x_l = F(x_{l-1}) + x_{l-1} \quad (2.18)$$

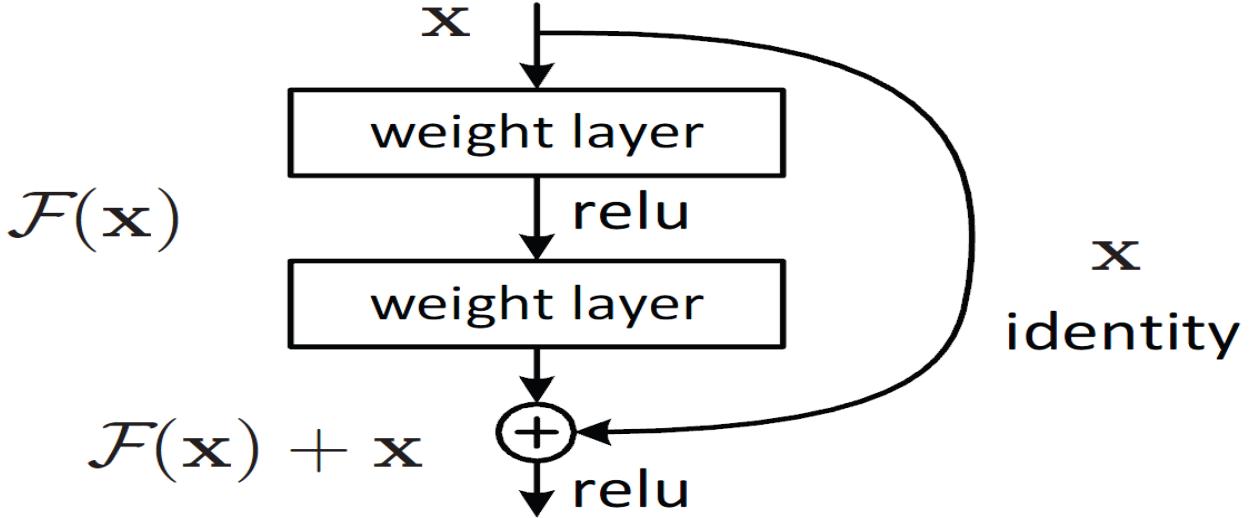


FIGURE 2.37 – Le schéma fonctionnel de ResNet [25]

Il existe de nombreux blocs résiduels de base inclus dans le réseau résiduel. Selon le type d'architecture du réseau résiduel, les opérations dans le bloc résiduel sont également modifiées.

Par rapport au réseau highway, ResNet présente les connexions de raccourcis à l'intérieur des couches pour permettre la connectivité entre les couches, qui sont sans paramètres et indépendantes des données. Notez que les couches caractérisent les fonctions non résiduelles lorsqu'un raccourci est fermé dans le réseau autoroutier. En revanche, les raccourcis d'individualité ne sont jamais fermés, tandis que les informations résiduelles sont transmises en permanence dans ResNet. En outre, ResNet a le potentiel d'éviter les problèmes de diminution du gradient, car les connexions de raccourci (liens résiduels) accélèrent la convergence du réseau profond. ResNet a été le vainqueur du championnat 2015-ILSVRC avec 152 couches de profondeur ; cela représente 8 fois la profondeur de VGG et 20 fois la profondeur d'AlexNet. En comparaison avec le VGG, il a une complexité de calcul plus faible, même avec une profondeur élargie [25].

### 2.3.3.5.3 DenseNet

Pour résoudre le problème de la disparition du gradient, DenseNet a été présenté, en suivant la même direction que ResNet. L'un des inconvénients de ResNet est qu'il conserve clairement l'information au moyen de transformations d'individualité préservatrices, car plusieurs couches contribuent à une information extrêmement faible ou nulle. En outre, ResNet possède un grand nombre de poids, puisque chaque couche possède un groupe de poids isolé.

DenseNet a utilisé la connectivité entre couches dans une approche améliorée pour résoudre ce problème. Il a connecté chaque couche à toutes les couches du réseau en utilisant une approche de type feed-forward. Par conséquent, les cartes de caractéristiques de chaque couche précédente ont été utilisées pour alimenter toutes les couches suivantes. Dans les CNN traditionnels, il y a  $l$  connexions entre la couche précédente et la couche actuelle, alors que dans DenseNet, il y a  $\frac{l(l+1)}{2}$  connexions directes. DenseNet démontre l'influence des sages-convolutions de profondeur entre les couches. Ainsi, le réseau gagne la capacité de discriminer clairement entre l'information ajoutée et l'information préservée, puisque DenseNet concatène les caractéristiques des couches précédentes plutôt que de les ajouter. Cependant, en raison de sa structure étroite de couches, DenseNet devient paramétriquement coûteux en plus du nombre accru de cartes de caractéristiques. L'admission directe de toutes les couches aux gradients via la fonction de perte améliore le flux d'informations dans tout le réseau. En outre, cela inclut un impact de régularisation, qui minimise le surajustement sur les tâches avec des ensembles de formation mineurs [32].

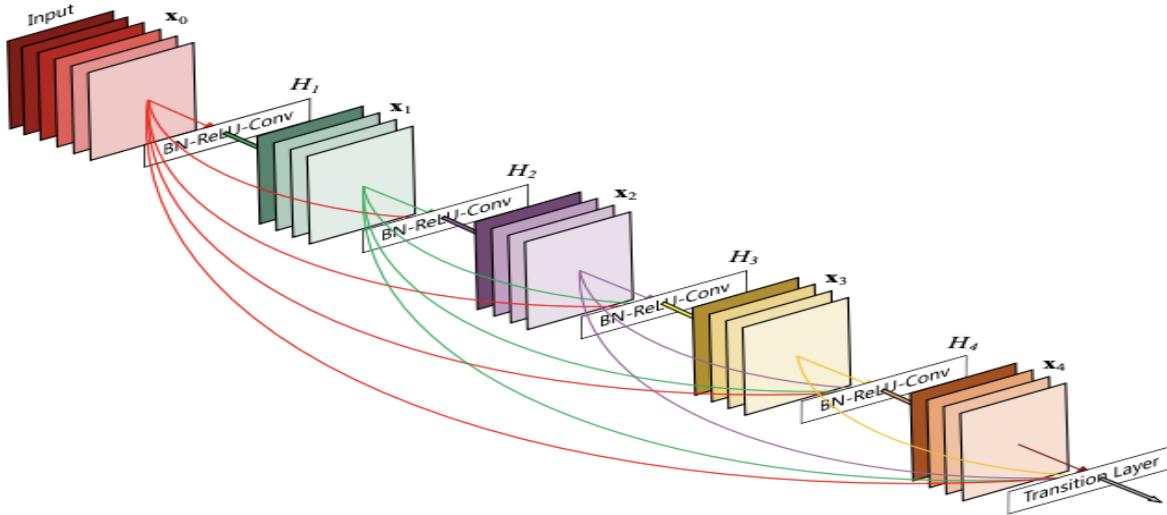


FIGURE 2.38 – L'architecture de DenseNet [32]

#### 2.3.3.5.4 Inception :ResNet and Inception-V3/4

Szegedy et al [34] [29] [22], ont proposé Inception-ResNet et Inception-V3/4 comme types améliorés d'Inception-V1/2. Le concept derrière Inception-V3 était de minimiser le coût de calcul sans affecter la généralisation du réseau en profondeur. Ainsi, Szegedy et al. ont utilisé des filtres asymétriques de petite taille ( $1 \times 5$  et  $1 \times 7$ ) plutôt que des filtres de grande taille ( $7 \times 7$  et  $5 \times 5$ ) ; de plus, ils ont utilisé un bottleneck de convolution  $1 \times 1$  avant les filtres de grande taille. Ces modifications rendent le fonctionnement de la convolution traditionnelle très similaire à la corrélation intercanaux. Auparavant, Lin et al ont utilisé le potentiel du filtre  $1 \times 1$  dans l'architecture NIN. Par la suite, ils ont utilisé la même idée de manière intelligente. En utilisant l'opération convolutive  $1 \times 1$  dans Inception-V3, les données d'entrée sont mappées dans trois ou quatre espaces isolés, qui sont plus petits que les espaces d'entrée initiaux. Ensuite, toutes ces corrélations sont représentées dans ces espaces plus petits par des convolutions communes  $5 \times 5$  ou  $3 \times 3$ . En revanche, dans Inception-ResNet, Szegedy et al. réunissent le bloc d'inception et la puissance d'apprentissage résiduelle en remplaçant la concaténation des filtres par la connexion résiduelle. Szegedy et al. ont démontré empirique-

ment que Inception-ResNet (Inception-4 avec des connexions résiduelles) peut atteindre un pouvoir de généralisation similaire à Inception-V4 avec une largeur et une profondeur élargies et sans connexions résiduelles. Ainsi, il est clairement illustré que l'utilisation des connexions résiduelles dans la formation accélère considérablement la formation du réseau Inception [34] [29].

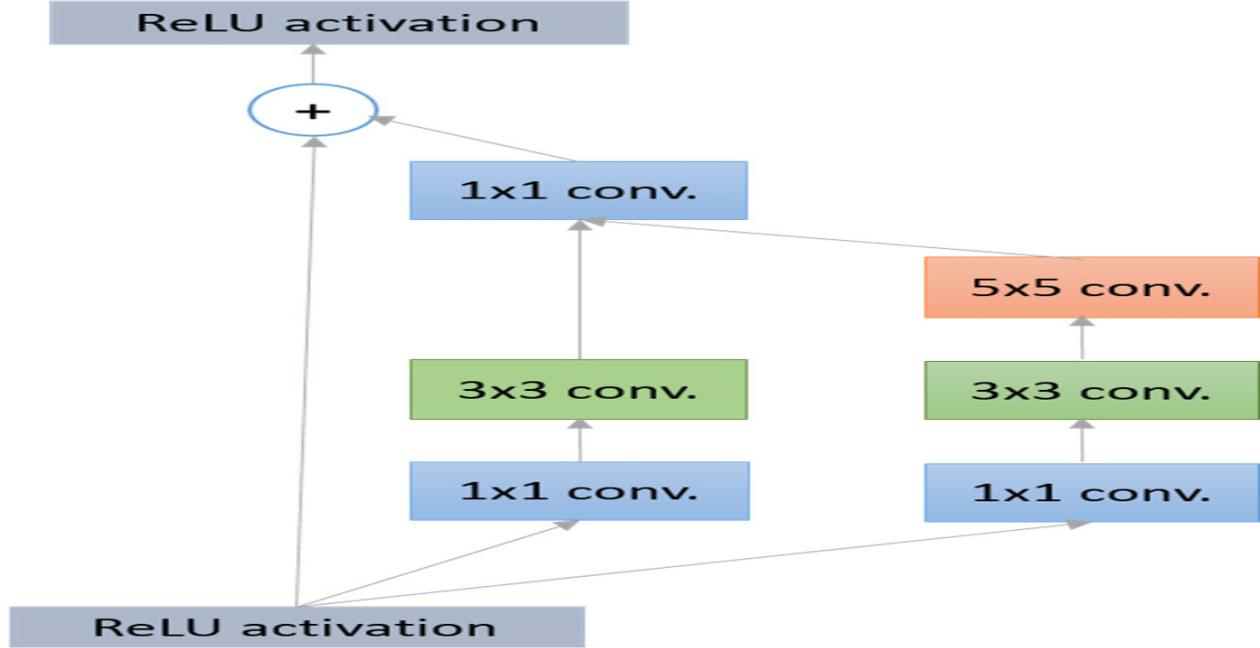


FIGURE 2.39 – Le schéma fonctionnel de base de l'unité Inception Residual [32]

## 2.4 Théorie des jeux

### 2.4.1 Définition

La théorie des jeux se propose d'étudier des situations (jeux) où des individus (les joueurs) prennent des décisions, chacun étant conscient que le résultat de son propre choix (ses gains) dépend de celui des autres. C'est pourquoi on dit parfois de la théorie des jeux qu'elle est une "théorie de la décision en interaction". Les décisions ayant pour but un gain maximum (la rationalité).

La théorie des jeux s'intéresse à des modèles d'un type particulier, les jeux , qui sont constitués de trois éléments : les joueurs, leurs ensembles de stratégies (un par joueur) et les règles du jeu (qui portent notamment sur les gains et l'information de chacun). Après avoir caractérisé chacun de ces éléments, on s'intéressera aux divers types de solution proposés par les théoriciens des jeux pour leurs modèles, dans une perspective "coopérative" puis "non coopérative" [101].

### 2.4.2 Historique

Depuis que John Von Neumann et Oskar Morgenstern ont publié en 1944 leur fameux livre s'intitulant : " Theory of Games and Economic Behavior ", la théorie des jeux a gagné une place incontournable en économie. Cependant elle trouve aujourd'hui de nombreuses applications notamment en intelligence artificielle et la théorie des multi-agents, biologie

et beaucoup plus récemment en informatique de manière générale (Vision par ordinateur, cryptographie, théorie algorithmique des jeux, vérification de preuves etc...) [52].

### 2.4.3 Composants d'un jeu

Un jeu stratégique se caractérise par un ensemble de règles de jeu spécifiant :

- **Une stratégie** : Un plan d'action complet spécifiant ce que le joueur fera à chaque étape de décision et face à chacune des situations pouvant survenir au cours du jeu [8].
- **Joueur** : c'est un agent qui prend des décisions dans un jeu [8].
- **Rationalité** : les joueurs cherchent à maximiser leur gain, compte tenu de l'information dont ils disposent et ce fait est connaissance commune (chacun sait que les autres sont rationnels, qu'ils savent qu'il sait, etc...) [3].

### 2.4.4 Types d'un jeu

#### 2.4.4.1 Un jeu coopératif

La partie de la théorie des jeux qui s'occupe de la détermination des éléments socialement préférables (au niveau du groupe plutôt que de l'individu seul en d'autres termes) de l'ensemble des issues I est souvent dite "coopérative" ou "coalitionnelle". Elle nécessite que les différentes parties puissent communiquer entre elles et... qu'elles soient rationnelles [89].

#### 2.4.4.2 Un jeu non coopératif

La partie dite, au contraire, "non coopérative" ou "stratégique" ne s'intéresse pas à la mise en oeuvre des solutions préconisées par la théorie des jeux coopératifs qui ont force de loi. Elle suppose que les différentes parties en communiquent pas entre elles ou ne sont pas rationnelles [89].

### 2.4.5 Les formes d'un jeu

#### 2.4.5.1 La forme normale

Un jeu sous forme normale représente une interaction entre un nombre fini de joueurs rationnels, chaque joueur choisit une action et dispose d'un ensemble fini d'actions. Le choix étant simultané pour tous les joueurs. Tous les joueurs connaissent le jeu et l'objectif de chacun est de maximiser son gain. Si le jeu est à un seul joueur il devient un problème de maximisation [52]. Le tableau ci-dessous présente un exemple d'un jeu en forme normale (dilemme du prisonnier) :

	Avoue	N'avoue pas
Avoue	(-5,-5)	(0,10)
N'avoue pas	(-10,0)	(-1,-1)

TABLE 2.1 – Table du jeu dilemme du prisonnier

### 2.4.5.2 La forme extensive

La modélisation sous forme extensive est un des moyens les plus simples de représenter un jeu. Il s'agit d'un modèle où les joueurs choisissent séquentiellement leurs actions, jusqu'au moment où le jeu est déclaré fini.

Tout jeu sous forme extensive peut être représenté par un arbre (graphe connexe sans cycle) où à chaque noeud terminal correspond un résultat du jeu à chaque noeud non terminal est associé un joueur : arrivé à ce point du jeu, c'est à son tour de jouer. chaque arc représente chacune des actions que ce joueur peut prendre à ce point du jeu [30].

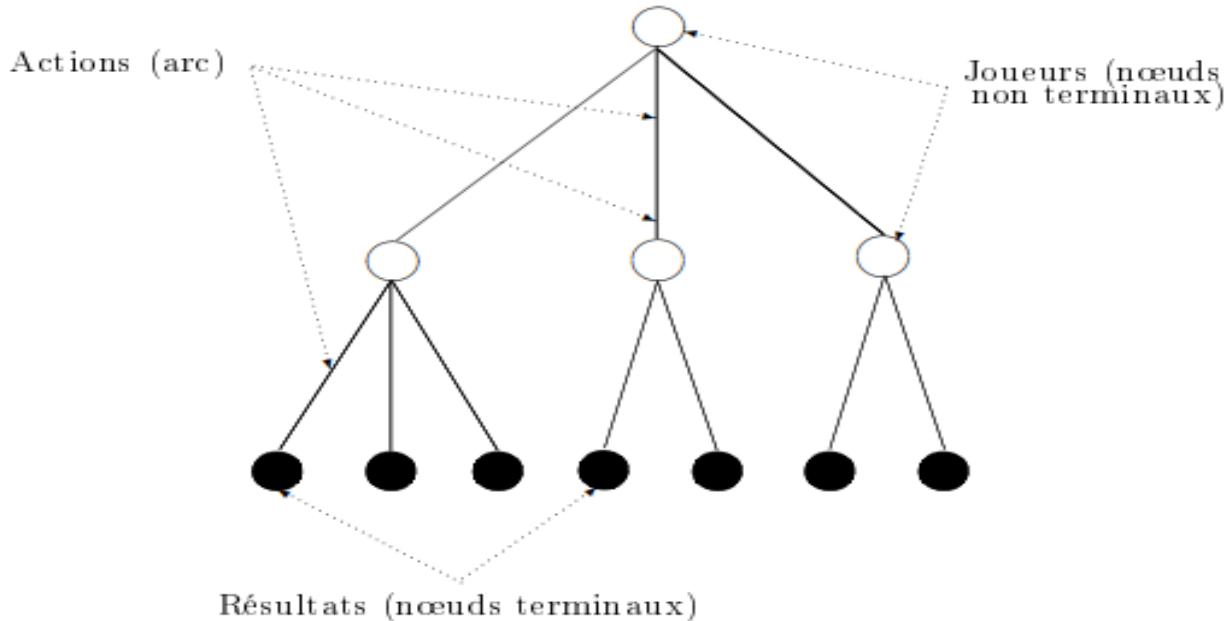


FIGURE 2.40 – Représentation graphique de la forme extensive d'un jeu

### 2.4.6 Les stratégies dominantes

Une stratégie dominante pour un joueur est une stratégie qui lui donne toujours un gain supérieur ou égal au gain qu'il peut attendre de toutes ses autres stratégies (quelles que soient les stratégies des autres joueurs). [8].

$$\forall s_i \in S_i - \{s'_i\} \text{ nous avons } u_i(s'_i, s_{-i}) > u_i(s_i, s_{-i}) \quad (2.19)$$

### 2.4.7 Les stratégies dominées

Une stratégie dominée pour un joueur est une stratégie qui lui donne toujours un gain inférieur à celui d'au moins une des autres stratégies à sa disposition (quelles que soient les stratégies des autres joueurs) [8].

$$\forall s_i \in S_i - \{s'_i\} \text{ nous avons } u_i(s'_i, s_{-i}) < u_i(s_i, s_{-i}) \quad (2.20)$$

### 2.4.8 L'équilibre de Nash

L'équilibre de Nash est un concept de la théorie des jeux selon lequel l'issue optimale d'un jeu est celle où il n'y a aucune incitation à dévier de la stratégie initiale. Plus précisément,

l'équilibre de Nash est un concept de la théorie des jeux où le résultat optimal d'un jeu est celui où aucun joueur n'est incité à dévier de la stratégie choisie après avoir pris en compte le choix de son adversaire [107].

Globalement, un individu ne peut recevoir aucun avantage supplémentaire en changeant d'action, en supposant que les autres joueurs conservent des stratégies constantes. Un jeu peut avoir plusieurs équilibres de Nash ou n'en avoir aucun.

Un profil de stratégie  $s^* = (s_1^*, s_2^*, \dots, s_i^*, \dots, s_n^*)$  est un équilibre de Nash si et seulement si l'inégalité suivante est satisfaite pour chaque joueur  $i = 1, \dots, n$  [52] :

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \text{ pour tout } s_i \in S_i \quad (2.21)$$

### 2.4.9 Jeu à somme nulle

Un jeu à somme nulle est un triplet  $\Omega = \{A, B, f\}$  [8] :

- **A** : Est un ensemble non vide appelé ensemble d'actions (ou de stratégies) du joueur  $J_1$ .
- **B** : Est un ensemble non vide d'actions du joueur  $J_2$
- **f** :  $A \times B \Rightarrow R$  est une fonction bornée qu'on appelle fonction de paiement ou bien d'utilité ou encore de gain. Le joueur  $J_1$  cherche à maximiser et le joueur  $J_2$  à minimiser.
- La somme des gains et des pertes de tous les joueurs est égale à 0.

### 2.4.10 MaxMin et MinMax

#### 2.4.10.1 MaxMin

Le Maximin est le Maximum du Minimum de satisfaction au cas où le pire arriverait. C'est la tactique du moindre mal : la prudence absolue [50].

Le MaxMin représente le paiement maximal que le joueur 1 peut assurer quelle que soit l'action du joueur 2.

Le MaxMin en stratégie pure est noté  $\underline{v}$  est la quantité :

$$\underline{v} = \text{Max}_{a \in A} (\text{Min}_{b \in B} f(a, b)) \quad (2.22)$$

Tel que :  $a$  est la stratégie du joueur 1,  $b$  est la stratégie du joueur 2,  $f$  est la fonction d'utilité.

#### 2.4.10.2 MinMax

Le Minimax c'est rendre Minimum le Maximum de ce que l'autre peut faire. On joue empathique et paranoïaque, en anticipant les coups bas de l'autre [50].

Le MinMax représente le paiement minimal que le joueur 2 peut assurer quelle que soit l'action de joueur 1.

Le MaxMin en stratégie pure est noté  $\bar{v}$  est la quantité :

$$\bar{v} = \text{Min}_{b \in B} (\text{Max}_{a \in A} f(a, b)) \quad (2.23)$$

### 2.4.11 Stratégies prudentes

#### 2.4.11.1 Stratégie prudente en pure

Une stratégie pure du joueur  $i$  est dite prudente (ou MaxMin) si elle garantit son MaxMin.

### 2.4.11.2 Stratégie prudente en mixte

Une stratégie mixte  $\sigma_1$  du joueur 1 est une stratégie prudente ou MaxMin si seulement si :

$$\sigma_1 \in \arg \max_{\sigma_1 \in \sum^i} \min_{s_2 \in S_2} f(\sigma_1, \sigma_2) \quad (2.24)$$

### 2.4.12 Valeur du jeu

#### 2.4.12.1 Valeur du jeu en pure

On dit qu'un jeu admet une valeur  $v$  en stratégie pure si et seulement si  $\text{MinMax} = \text{MaxMin}$  [90].

#### 2.4.12.2 Valeur du jeu en mixte

Considérons un jeu fini à somme nulle alors un profil  $(\sigma_1^*, \sigma_2^*)$  est un profil d'équilibre de Nash si seulement c'est un profil de stratégies prudentes [90].

### 2.4.13 Exemples d'applications de la théorie des jeux

La théorie des jeux a de nombreuses utilisations et se retrouve partout dans des cas réels, nous en mentionnerons quelques-uns :

#### 2.4.13.1 Décisions relatives à la fixation des prix des produits

Les applications de la théorie des jeux sont également largement utilisées pour décider des stratégies de prix des consommateurs et des détaillants. Les détaillants sont en concurrence les uns avec les autres pour gagner des parts de marché et pour cela, ils optent pour différents jeux ou stratégies, comme l'offre de remises intéressantes sur des produits spécifiques afin d'augmenter les ventes de produits complémentaires. Par exemple, pendant la basse saison, c'est-à-dire la basse saison d'été ou la basse saison d'hiver, les vendeurs de vêtements ou les détaillants proposent des ventes attrayantes sur un certain stock de vêtements, en adoptant des stratégies de prix optimales pour attirer un maximum de clients. Pour ce faire, les détaillants utilisent l'approche de la théorie des jeux où les détaillants et les consommateurs sont les principaux acteurs. Les détaillants se concentrent sur l'utilisation de la meilleure stratégie de prix tandis que la préférence des consommateurs est de choisir la meilleure offre en termes de rabais et de variété [67].

#### 2.4.13.2 Stratégies de guerre

La riposte musclée de l'Inde à la perfidie du Pakistan post-Uri, post-Pulwama a des fondements dans la théorie des jeux. Après l'attaque jihadiste contre le camp militaire d'Uri en 2016, l'Inde a lancé sa "frappe chirurgicale". Cette année, après qu'un kamikaze a éperonné un convoi de la Central Reserve Police Force et tué 40 jawans, l'armée de l'air indienne a frappé un camp terroriste à Balakot au fin fond du Pakistan, dans la province de Khyber Pakhtunkhwa. Ces stratégies de guerre et ces décisions militaires sont des exemples de la théorie des jeux. En général, le chef militaire ou le commandant choisit le plan d'action qui offre la promesse de succès la plus significative au vu des capacités de l'ennemi à s'opposer à lui [67].

#### 2.4.13.3 Utilisation de la théorie des jeux dans la Segmentation d'image

La théorie des jeux en sa qualité est un outil fort d'étude des situations et d'analyse des comportements des joueurs dans le but de prendre des décisions stratégiques, C'est pourquoi les chercheurs ont pensé de tirer profit de la théorie des jeux dans la segmentation d'images par fusion de résultats [24].

#### 2.4.13.4 Utilisation de la théorie des jeux dans la conception de jeux vidéo

L'IA stratégique de la machine (jeux contre l'ordinateur) est souvent primitive, Le concepteur du jeu passe son temps à « bluffer » le joueur pour qu'il ne découvre pas trop vite les mécanismes. Cette difficulté doit être contrôlée à l'aide de la théorie des jeux [4].

### 2.5 Conclusion

Dans ce chapitre, nous avons présenté les outils de conception du système d'aide au diagnostic automatique du cancer du poumon, nous avons commencé par quelques notions de base du traitement d'image, Ensuite nous avons parlé de l'apprentissage profond en détaillant les CNN et en mentionnant certaines des architectures les plus utilisées. Enfin, nous avons défini la théorie des jeux et ses concepts de base et des exemples d'application de cette dernière. Le chapitre suivant sera consacré à la conception de notre système.

# Chapitre 3

## Conception du système

### 3.1 Introduction

La détection précoce des nodules pulmonaires peut contribuer à la détection précoce du cancer du poumon, ce qui pourrait améliorer le pronostic des patients atteints de cancer du poumon et réduire les coûts médicaux [84].

Cependant, la détection des nodules pulmonaires est une tâche difficile car une image CT contient non seulement le poumon mais aussi d'autres tissus, et certains d'entre eux peuvent avoir des formes sphériques et ressembler à des nodules.

Par conséquent, notre plan de travail consiste à construire un système de diagnostic assisté par ordinateur basé sur l'apprentissage profond et la théorie des jeux pour aider les cliniciens à détecter et à classer les nodules pulmonaires bénins et malins.

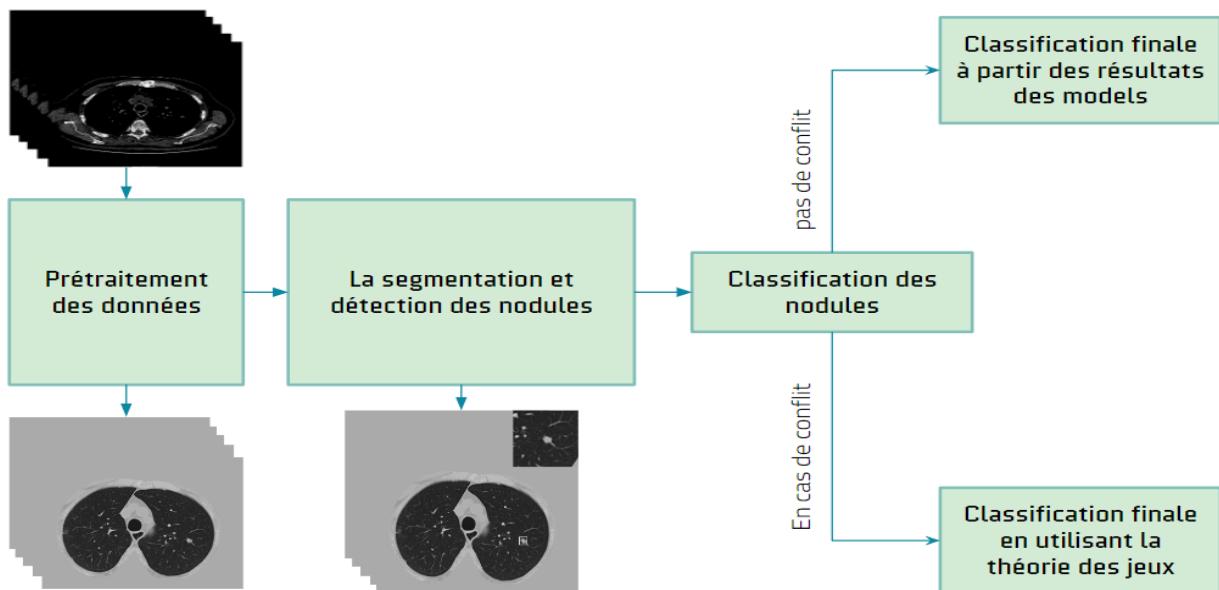


FIGURE 3.1 – L'architecture de notre système de détection et de classification des nodules pulmonaires

Dans ce chapitre, nous allons d'abord expliquer toutes les étapes du pré-traitement des données, puis nous mentionnerons tous les concepts utilisés pour la segmentation et la classification des nodules pulmonaires.

## 3.2 Pré-traitement des données

Le pré-traitement des données est une étape très importante, surtout pour l'apprentissage automatique.

Dans notre cas, le pré-traitement d'une image CT consiste à supprimer tous les autres tissus et à ne laisser que les informations nécessaires sur les poumons et l'amélioration de la qualité d'image en réduisant le bruit et en ajustant le contraste de l'image d'entrée.

### 3.2.1 Chargement et conversion des scans en HU

Dans nos données, il existe deux types de fichiers '.mhd' et '.dcm'. Chacun de ces fichiers contient des informations réelles sur le patient et du CT scan , ainsi qu'un chemin pour les données de scan 3d .

Toutes les données brutes du scan sont d'abord converties en unité **Hounsfield (HU)**, qui est une échelle quantitative standard pour décrire la radio-densité. Chaque tissu a sa propre gamme HU spécifique, et cette gamme est la même pour différentes personnes.



FIGURE 3.2 – Un exemple de coupe de CT-scan après le chargement et la conversion des données en HU

### 3.2.2 L'extraction du masque des poumons

les fichiers '.mhd' ont aussi un chemin pour le masque pulmonaire segmenté mais dans les fichiers '.dcm' nous devons extraire le masque manuellement en suivant ces étapes :

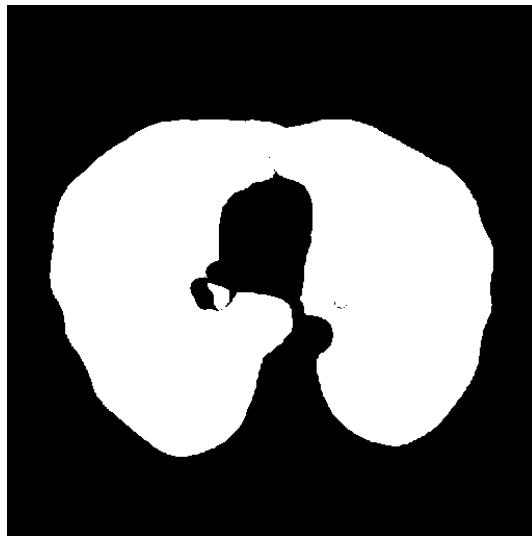


FIGURE 3.3 – Un exemple de masque de poumons lié au fichier '.mhd'

### 3.2.2.1 Binarisation des images par seuillage

Dans cette étape, nous allons extraire le masque de poumon et ignorer tous les autres tissus. Chaque tranche 2d du scan est filtrée avec un filtre gaussien (standard écart = 1 pixel) puis binarisé en utilisant -600 comme seuil, et toutes les composantes connectées 2D inférieures à  $30 \text{ mm}^2$  ou dont l'excentricité est supérieure à 0,99 (qui correspondent à un bruit d'imagerie radiale à forte luminance) sont supprimées et ce processus renvoie comme résultat une matrice binaire 3d.

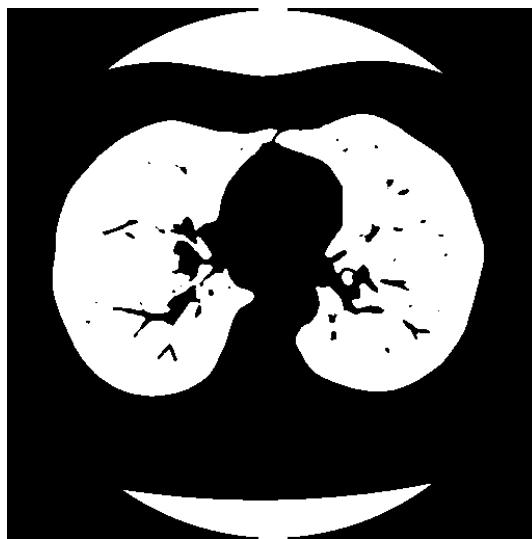


FIGURE 3.4 – Un exemple d'une tranche binarisée du scan.

### 3.2.2.2 La sélection du domaine connecté correspondant aux poumons

Ensuite, toutes les composantes connectées 3D de la matrice binaire 3D résultante sont calculées, et seules celles qui ne touchent pas le coin de la matrice et dont le volume est compris entre 0,68 L et 7,5 L sont conservées.

Après cela, il ne reste généralement qu'une seule composante binaire correspondant au poumon, mais il y a parfois aussi des composantes interférentes. Par rapport à ces composantes in-

terférentes, la composante pulmonaire est toujours au centre de l'image. Pour chaque tranche d'une composante, nous calculons la distance minimale entre celle-ci et le centre de l'image (MinDist) et sa surface. Ensuite, nous sélectionnons toutes les tranches ayant une surface  $> 6000 \text{ mm}^2$  dans la composante, et nous calculons la MinDist moyenne de ces tranches. Si la MinDist moyenne est supérieure à 62 mm, cette composante est supprimée.

Les composants restants sont alors combinés, représentant le masque du poumon. Dans certains cas, le poumon est relié au monde extérieur sur les tranches supérieures, ce qui empêche la procédure décrite ci-dessus de séparer le poumon de l'espace du monde extérieur. Par conséquent, ces tranches doivent d'abord être retirées pour que le traitement ci-dessus fonctionne.



FIGURE 3.5 – Une tranche du masque final extrait du 3d scan

### 3.2.3 L'enveloppe convexe et dilatation

Il y a quelques nodules attachés à la paroi externe du poumon. Ils ne sont pas inclus dans le masque obtenu à l'étape précédente, ce qui est indésirable. Pour les garder à l'intérieur du masque, un moyen pratique consiste à calculer L'enveloppe convexe (**convex hull**) du masque. Toutefois, le calcul direct de L'enveloppe convexe du masque inclut trop de tissus non liés (comme le cœur et la colonne vertébrale). Ainsi, le masque pulmonaire est d'abord séparé en deux parties (correspondant approximativement aux poumons gauche et droit) avant le calcul de L'enveloppe convexe en utilisant l'approche suivante :

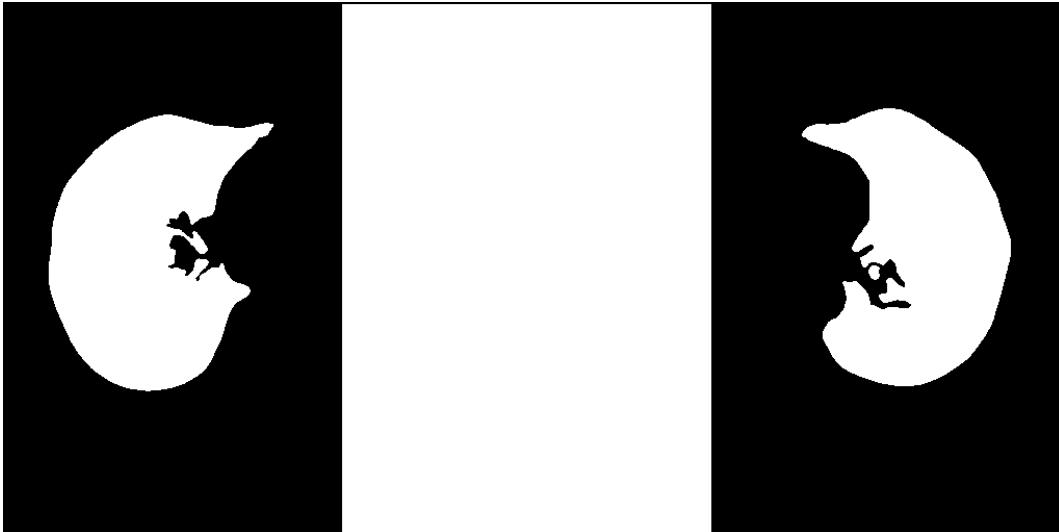


FIGURE 3.6 – Un exemple des masques de poumons séparés afin d'appliquer le L'enveloppe convexe

Le masque est érodé de manière itérative jusqu'à ce qu'il soit divisé en deux composants (leurs volumes seraient similaires), qui sont les parties centrales des poumons gauche et droit. Ensuite, les deux composants sont dilatés à leurs tailles d'origine. Leurs intersections avec le masque brut sont maintenant des masques pour les deux poumons séparément.

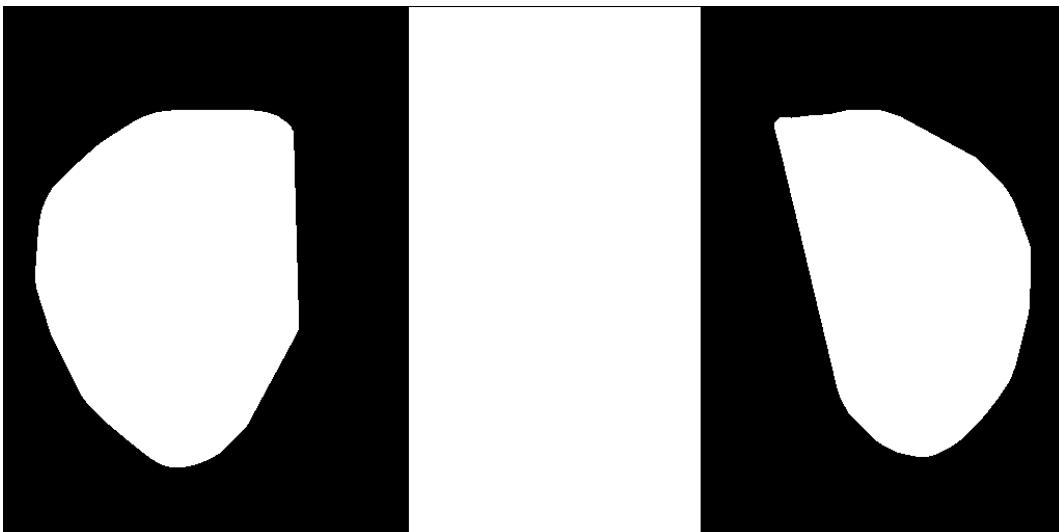


FIGURE 3.7 – Un exemple des masques des poumons gauche et droit après application de L'enveloppe convexe

Les masques résultants sont encore dilatés de 10 voxels pour inclure un peu d'espace environnant. Un masque complet est obtenu en réunissant les masques des deux poumons. Cependant, certaines tranches 2D de la partie inférieure du poumon ont des formes en croissant. Leurs enveloppes convexes peuvent contenir trop de tissus indésirables. Ainsi, si la surface de l'enveloppe convexe d'un masque 2D est supérieure à 1,5 fois celle du masque lui-même, le masque d'origine est conservé.

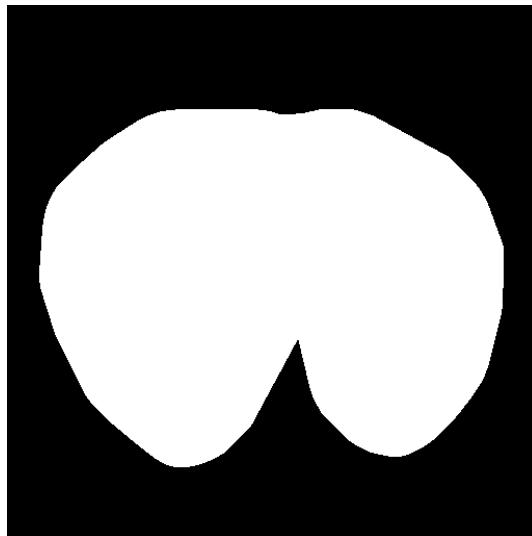


FIGURE 3.8 – Un exemple de masque pulmonaire complet regroupé après dilatation

### 3.2.4 Normalisation de l'intensité

Pour préparer les données pour les réseaux profonds, nous transformons l'image de HU en UINT8. La matrice de données brutes est d'abord coupée entre [-1200, 600] et transformée linéairement en [0, 255]. Il est ensuite multiplié par le masque complet obtenu ci-dessus, et tout ce qui se trouve à l'extérieur du masque est rempli de 170, qui est la luminance des tissus communs.

De plus, pour l'espace généré par dilatation à l'étape précédente, toutes les valeurs supérieures à 210 sont également remplacées par 170. Parce que la zone environnante contient des os (les tissus à haute luminance), ils sont facilement classés à tort comme calcifiés nodules (également des tissus à haute luminance). Nous choisissons de remplir les os avec 170 afin qu'ils ressemblent à des tissus normaux. L'image est recadrée dans les 3 dimensions afin que la marge à chaque côté soit de 10 pixels.



FIGURE 3.9 – Une coupe du CT-scan après la phase de pré-traitement

### 3.2.5 Préparation des informations des nodules

Dans nos données, il existe également un fichier qui contient les coordonnées et les diamètres des nodules de tous les scans, ces coordonnées sont représentées sur le système de coordonnées du monde réel.

Par conséquent, nous devons convertir ces coordonnées dans le système de coordonnées voxel et après cela nous créons pour chaque scan une liste qui contient toutes les coordonnées et les diamètres des nodules. Cette liste est utilisée pour afficher les nodules et aussi pour la création des masques binaires.

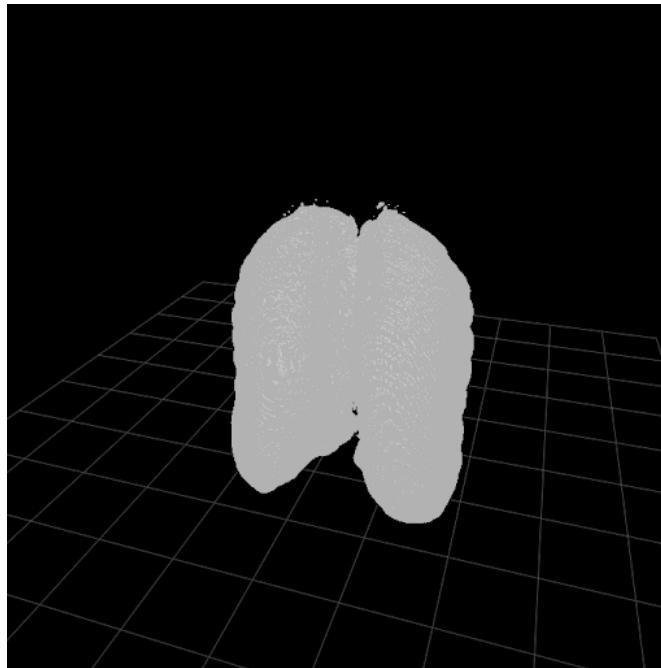


FIGURE 3.10 – Représentation 3D du CT-scan après le pré-traitement des données

## 3.3 La Segmentation des nodules

Dans les systèmes de diagnostic assisté par ordinateur pour le cancer du poumon, la segmentation des nodules pulmonaires est importante pour analyser les caractéristiques des nodules pulmonaires sur les images de tomographie par ordinateur (CT) et distinguer les nodules malins des nodules bénins.

Nous proposons donc deux modèles pour la segmentation des nodules pulmonaires, une architecture U-Net 2d et une autre 3D.

### 3.3.1 Segmentation 2d des nodules pulmonaires par tranches

#### 3.3.1.1 Pré-traitement des données d'entrée

Pour chaque nodule en utilisant les coordonnées 3d ( $x,y,z$ ) du centre du nodule et son diamètre( $d$ ),nous prenons toutes les tranches du nodule dans l'intervalle  $[z-d//2 , z+d//2]$ .Aussi un masque 2d est créé pour chaque tranche en utilisant le diamètre ( $d$ ) et les coordonnées ( $x,y$ ). Pour réduire le temps d'apprentissage du modèle, toutes les tranches et leurs masques sont redimensionnés en  $256*256$ .

Après , nous divisons ces données en deux parties : partie d'entraînement qui représente 70%

des données, et la partie test et validation qui représente 30% des données. Chaque tranche du scan peut être ajoutée soit aux données d'entraînement, ou aux données de test pour éviter toute ressemblance entre ces données.

L'augmentation des données est utilisée pour atténuer le problème de over-fitting, les données de l'entraînement sont aléatoirement inversées ou tournées de 90° dans une direction aléatoire.

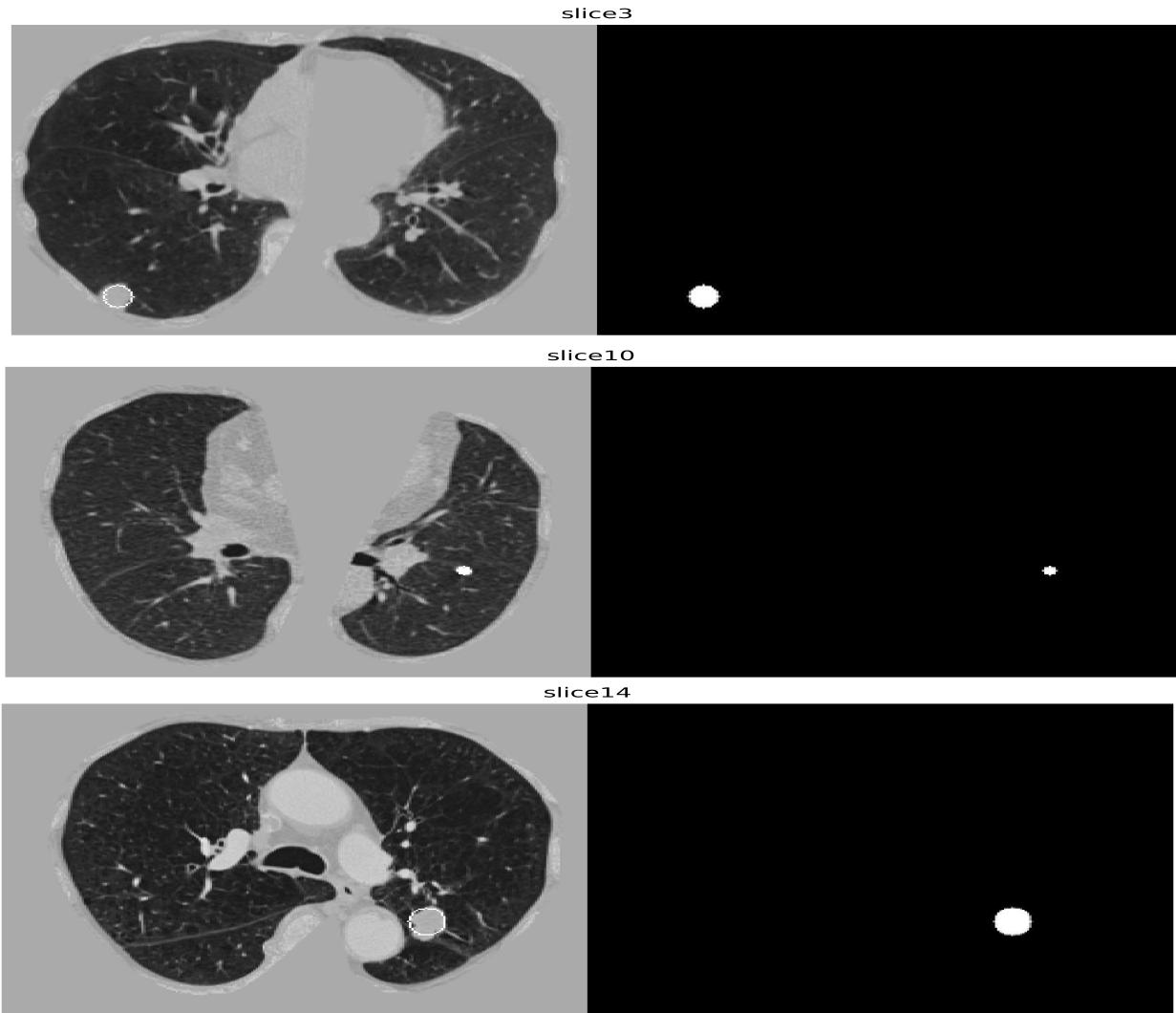


FIGURE 3.11 – Exemples des tranches 2d du scan qui contiennent des nodules pulmonaires avec leurs masques

### 3.3.1.2 Architecture du modèle

Dans cette section, nous présenterons notre architecture U-net utilisée pour la solution de segmentation 2D. La figure ci-dessous représente une architecture globale avec plus de détails à suivre :

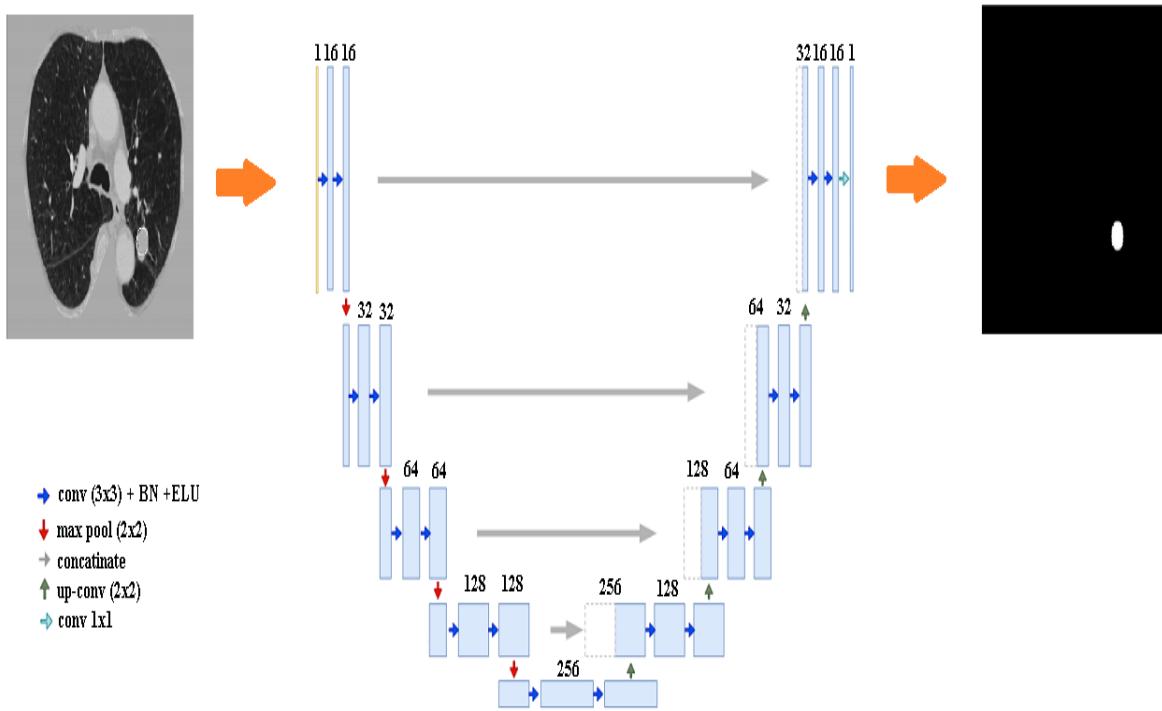


FIGURE 3.12 – L'architecture du modèle U-net 2D

La couche d'entrée du modèle reçoit des vecteurs normalisés de taille 256x256x1 comme entrées. Ensuite, nous faisons passer la couche d'entrée dans le premier bloc du chemin de contraction.

Chaque bloc du chemin de contraction est composé de l'application répétée de deux opérations de convolution 2d avec un nombre de noyaux de 3x3. Le nombre des filtres sur les couches de convolution 2d commence à 16 et se multiplie après chaque fin du bloc, pour atteindre 256 dans la couche finale. Chaque opération de convolution 2d est suivie par une fonction d'activation ELU et 2x2 max pooling ,batch normalisation , drop out régularisation pour éviter le problème du overfitting. L'ensemble du processus du chemin de contraction a le même padding que l'entrée.

Le chemin d'expansion est également composé de 4 blocs. Chacun de ces blocs est composé d'une couche d'up-convolution 2x2 avec un pas de 2. Une application répétée de deux opérations de convolution avec un nombre de noyaux de taille 3x3.commence de 128 et réduit deux fois après chaque fin de bloc et devient 16 dans la deuxième opération de convolution du dernier bloc du chemin d'expansion. Chaque opération de convolution ascendante dans ce chemin est concaténée avec la partie correspondante du chemin de contraction et suivie par une fonction d'activation ELU sa fonction mathématique est :

$$\begin{aligned} f(x) &= x, \text{ si } x \geq 0 \\ f(x) &= \alpha(e^x - 1), \text{ si } x < 0 \end{aligned} \quad (3.1)$$

La couche de sortie est la dernière couche du modèle U-NET. Cette couche est composée d'une opération de convolution avec un noyau de 1x1 suivi d'une fonction d'activation sigmoïde sa formule est comme suit :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

Cette opération produit un masque prédit de sortie de taille 256x256x1 et pour le processus de compilation nous avons choisi l'optimiseur Adam avec un taux d'apprentissage de 0,0003.

### 3.3.2 Segmentation 3d des nodules pulmonaires par blocs

Le poumon est une structure tridimensionnelle, le réseau UNet 2D utilise des opérations de convolution et de mise en commun bidimensionnelles pour extraire les caractéristiques des nodules pulmonaires à partir des images CT du poumon, ce qui entraîne une perte importante d'informations spatiales, une grande partie des informations contextuelles est perdue dans le processus de sous-échantillonnage.

Ces informations ne peuvent pas être entièrement restaurées lors du rééchantillonnage, ce qui conduit à des résultats de suréchantillonnage flous et une insensibilité aux détails de l'image. Combiné avec les problèmes ci-dessus, il faut un réseau tridimensionnel pour une optimisation plus poussée [69].

#### 3.3.2.1 Pré-traitement des données d'entrée

Pour chaque nodule en utilisant les coordonnées 3D (x,y,z) du centre du nodule et son diamètre (d), un bloc 3D de taille 64\*64\*64 qui contient le nodule est créé à partir du scan ,le centre du bloc est le centre du nodule.

Un masque binaire 3d similaire de taille 64\*64\*64 est créé en utilisant les mêmes coordonnées. Les blocs qui contiennent des nodules ne représentent que 70% des données, les 30% restants sont choisis au hasard a partir des scans et peuvent ne pas contenir de nodules (masque noir), et ceci est utilisé afin de réduire les faux positifs.

L'augmentation des données en 3D est plus puissante que l'augmentation des données en 2D. Par exemple, si nous ne considérons que l'inversion et l'échange d'axes, il y a 8 variantes dans le cas 2D mais 48 variantes dans le cas de la 3D. Plus précisément, les méthodes d'augmentation des données suivantes sont utilisées :

- Inversement aléatoire dans 3 directions.
- Rotation selon n'importe quel angle en 3D.

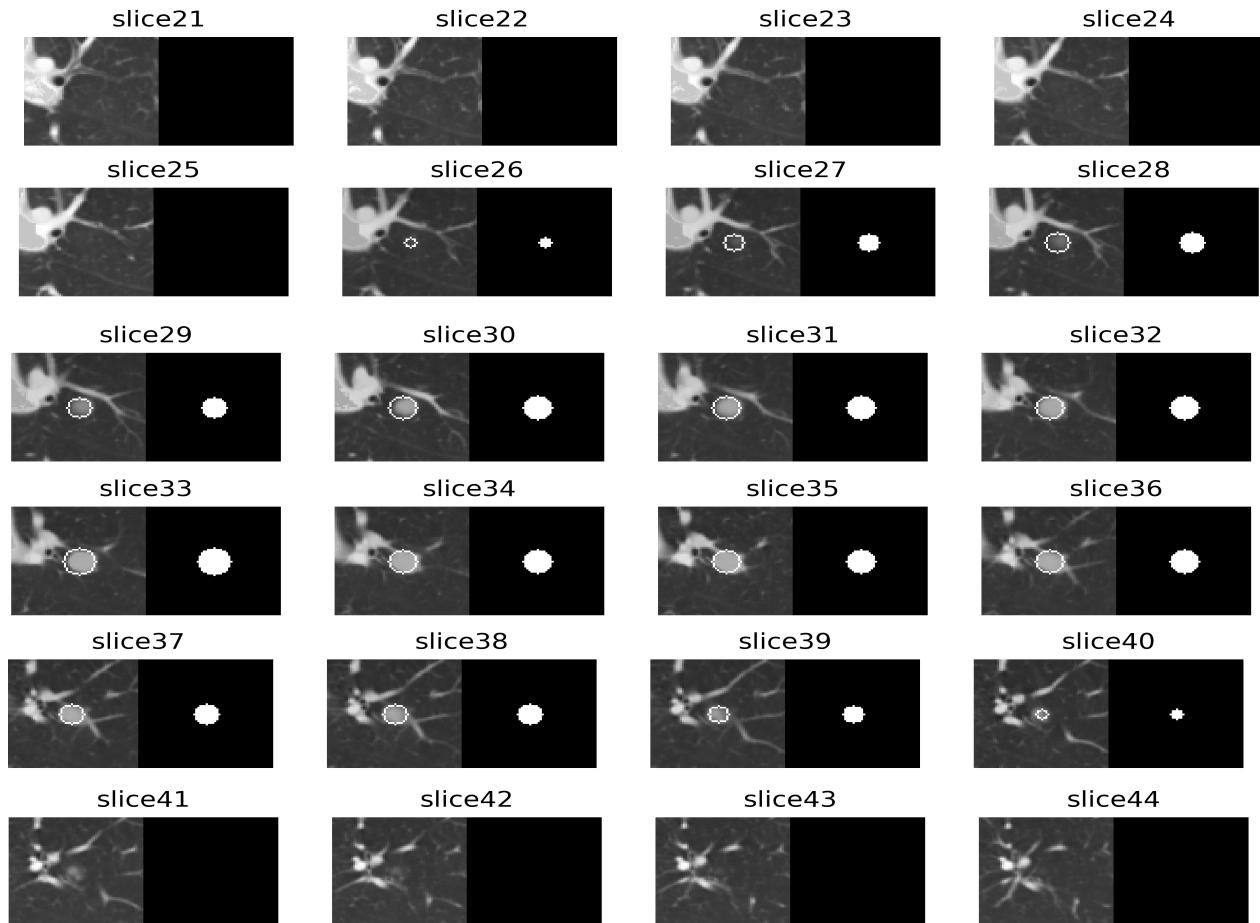


FIGURE 3.13 – Exemple d'un nodule pulmonaire à l'intérieur d'un bloc 64x64x64 du scan

### 3.3.2.2 Architecture du modèle

Semblable au modèle précédent, nous présenterons en détail le modèle utilisé pour la segmentation 3D. La figure ci-dessous représente l'architecture globale du modèle U-net utilisé :

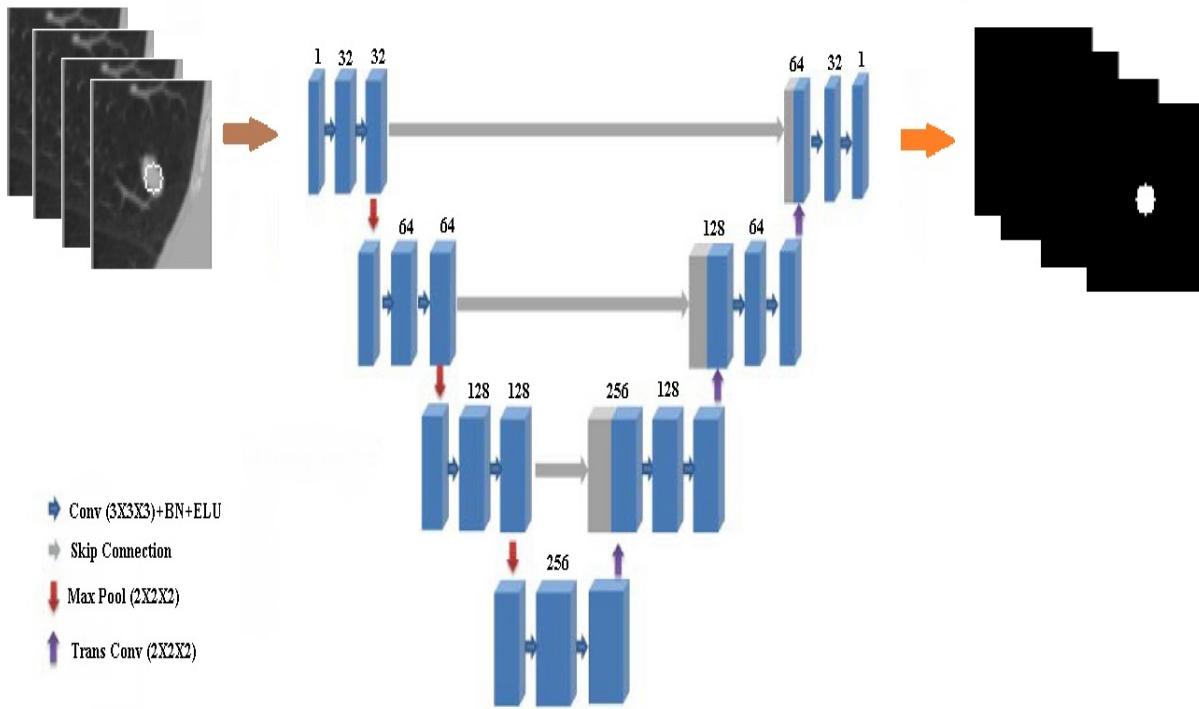


FIGURE 3.14 – L'architecture du modèle U-net 3D

La couche d'entrée reçoit des blocs normalisés de taille 64x64x64x1 comme entrées. Ensuite, nous faisons passer la couche d'entrée dans le premier bloc du chemin de contraction. Chaque bloc du chemin de contraction est composé de l'application répétée de deux opérations de convolution 3d avec un nombre de noyaux de 3x3x3. Le nombre des filtres sur les couches de convolution 3d commence à 32 et se multiplie après chaque fin du bloc, pour atteindre 512 dans la couche finale. Chaque opération de convolution 3d est suivie par une fonction d'activation ELU et 2x2x2 max pooling ,batch normalisation , drop out régularisation pour éviter le problème du over-fitting. L'ensemble du processus du chemin de contraction a le même padding que l'entrée.

Le chemin d'expansion est également composé de 4 blocs. Chacun de ces blocs est composé d'une couche d'up-convolution 2x2x2 avec un pas de 2. Une application répétée de deux opérations de convolution avec un nombre de noyaux de taille 3x3x3.commence de 128 et réduit deux fois après chaque fin de bloc et devient 16 dans la deuxième opération de convolution 3d du dernier bloc du chemin d'expansion.Chaque opération de convolution ascendante dans ce chemin est concaténée avec la partie correspondante du chemin de contraction et suivie par une fonction d'activation ELU.

La couche de sortie est la dernière couche du modèle U-NET. Cette couche est composée d'une opération de convolution avec un noyau de 1x1 suivi d'une fonction d'activation sigmoïde.

Cette opération produit un masque prédit de sortie de taille 64x64x64x1 et pour le processus de compilation nous avons choisi l'optimiseur Adam avec un taux d'apprentissage de 0,0003.

## 3.4 La classification des nodules pulmonaires

La classification des images médicales est une technique clé des systèmes de diagnostic assisté par ordinateur . Les méthodes traditionnelles s'appuient principalement sur les caractéristiques de forme, de couleur et/ou de texture ainsi que sur leurs combinaisons, dont la plupart sont spécifiques au problème et se sont révélées complémentaires dans les images médicales, ce qui conduit à un système qui n'a pas la capacité de représenter les concepts de haut niveau du domaine du problème et qui a une faible capacité de généralisation du modèle.

Les méthodes récentes d'apprentissage profond offrent un moyen efficace de construire des modèles capables de calculer les étiquettes de classification finale avec les pixels bruts des images médicales.

Par conséquent, la méthode que nous proposons pour la classification du cancer du poumon consiste à construire 4 modèles d'apprentissage profond ResNet 50, DenseNet121, InceptionV3, InceptionResNetv2 , chacun d'eux est entraîné et testé séparément sur les mêmes données.

Si au moins 3 des modèles ont donné le même résultat de classification, le nodule est classé comme bénin ou malin avec un certain degré calculé en utilisant la moyenne des résultats des modèles d'apprentissage profond.

Dans le cas contraire, 2 des modèles ont classé le nodule comme bénin et les autres comme malin et ce cas est considéré comme un cas de conflit, nous allons donc introduire le modèle de la théorie des jeux afin de résoudre le conflit en construisant une matrice de jeu à somme nulle, et la valeur du jeu décidera de la classification finale.

### 3.4.1 Les modèles d'apprentissage profond

Nous avons choisi les modèles ResNet50, DenseNet121, InceptionV3, InceptionResNetv2 parce qu'ils utilisent le concept de skip-connections qui résout le problème du vanishing-gradient, ce qui permet d'obtenir de meilleurs résultats que les modèles classiques tels que VGG16. Nous avons également essayé d'autres modèles(MobileNet,VGG19,AlexNet) et nous n'avons gardé que ces 4 modèles car ils donnent les meilleurs résultats.

Chacun des modèles est chargé à partir d'une bibliothèque externe [87] et prend en entrée un bloc 3D de taille 64\*64\*64\*3, et deux couches denses sont ajoutées. La couche de sortie est composée d'une opération de convolution avec un noyau de 1x1 suivi d'une fonction d'activation softmax sa formule est comme suit :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ pour } i = 1, \dots, K \text{ et } z = (z_1, \dots, z_K) \in \mathbb{R}^K \quad (3.3)$$

La fonction softmax retournera un vecteur 1d qui contient les probabilités de chaque classe(Bénin et Malin). Celles-ci sont utilisées aussi dans la fonction d'utilité du modèle de théorie des jeux.

Pour la compilation un optimiseur Adam est choisi avec un taux d'apprentissage de 0,0003.

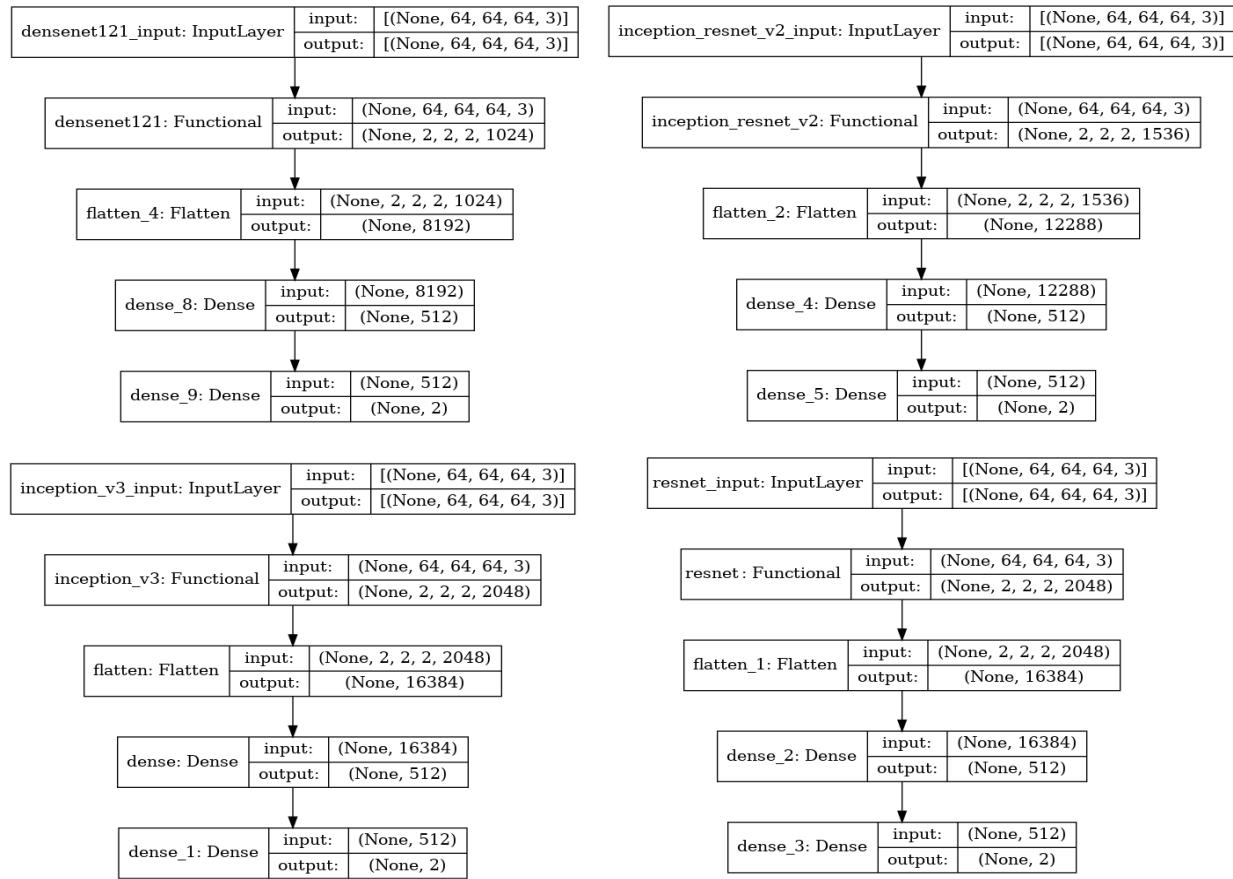


FIGURE 3.15 – Les 4 architectures utilisés

### 3.4.2 Entraînement des modèles

Les modèles sont entraînés en utilisant les blocs 3D des nodules de taille 64\*64\*64\*3 en entrée ainsi que les étiquettes de classification correspondant fournie pour le nodule, 0 pour bénin et 1 pour malin.

Les données sont divisées en 4 ensembles de d'entraînement et de validation/test, chaque modèle est entraîné , testé sur tous les ensembles, ce processus est appelé **K-fold cross-validation** [44].

Les méthodes d'augmentation des données suivantes sont utilisées :

- Inversement aléatoire dans 3 directions.
- Rotation selon n'importe quel angle en 3D.



FIGURE 3.16 – Les étapes du K-fold cross-validation [47]

### 3.4.3 Modèle de la théorie des jeux

Le but principal pour la création de ce modèle est d'avoir une classification finale unique avec le moindre taux d'erreur, les différents modèles d'apprentissage profond s'unissent pas toujours sur la même classification, ce qui entraîne un conflit de décision et d'où l'utilité de développer un système d'aide à la décision.

La solution était de modéliser ce problème de classification comme étant un jeu à deux joueurs (Malin, Non-Malin) et à somme nulle. Contrairement au modèle de segmentation des régions de la peau [62], dans notre travail nous avons proposé une fonction d'utilité qui s'appuie sur l'attribut probabiliste de la classification de chacune des architectures de l'apprentissage profond. Les différents résultats des modèles seront alors considérés comme des stratégies, chacun des joueurs essayera d'imposer sa propre classification du nodule. Le signe de la valeur du jeu en extension mixte représente alors la décision finale de notre système. Si les méthodes de toutes les caractéristiques s'unissent sur la même classification alors le système affiche ce résultat, sinon il y a donc un conflit et on passe les résultats des méthodes précédentes au modèle de la théorie des jeux pour résoudre le problème. Le modèle sera formalisé comme suit :

#### 3.4.3.1 Les joueurs

- L'ensemble des joueurs : {Malin, Non-Malin}.
- Chaque joueur maximise son gain. Donc au final le nodule est classifiée selon le gain.

#### 3.4.3.2 Les stratégies

- L'ensemble total des stratégies : {ResNet50, DenseNet121, InceptionV3, InceptionResNetV2}.
- Seulement les stratégies qui classifient le nodule comme la nature du joueur sont prises.

### 3.4.3.3 Fonction d'utilité

la fonction d'utilité  $U$  pour ce couple de stratégies est calculé selon la formule suivante :

$$U(S_1, S_2) = P_{S_1} - P_{S_2} \quad (3.4)$$

avec :

- $S_1$  : La stratégie du joueur malin.
- $S_2$  : La stratégie du joueur Bénin.
- $P_{S_1}$  : Degré de malignité prédit par le joueur malin.
- $P_{S_2}$  : Degré de bénignité prédit par le joueur bénin.

Voici un exemple d'un jeu :

		<b>Joueur Bénin</b>	
		DenseNet121	InceptionV3
<b>Joueur Malin</b>	ResNet50	( 0.21 , -0.21 )	( 0.38 , -0.38 )
	InceptionResNetv2	( 0.3 , -0.3 )	( 0.47 , -0.47 )

TABLE 3.1 – Matrice d'un jeu du notre system

### 3.4.3.4 Valeur finale du jeu

Après la construction de la matrice de jeu, la décision sur la classification du nodule est faite en cherchant l'équilibre en stratégies pures et s'il n'existe pas, on détermine l'équilibre en stratégies mixtes.

L'algorithme choisi est l'énumération du support [9] qui permet de détecter tous les équilibres possibles du jeu : Après l'application de l'algorithme d'énumération de support, la décision est déterminée selon la valeur mixte du jeu :

---

#### Algorithm 1 Algorithme d'énumération de support

---

Entrée : Matrice de jeu  $(A, B) \in \mathbb{R}^{m \times n^2}$

Début :

//Pour toute taille de support

**for all** ( $1 \leq k_1 \leq m$ ) et ( $1 \leq k_2 \leq n$ ) **do**

//Pour toute paire de support  $(I, J)$  de taille  $(k_1, k_2)$

**for all**  $(I, J)$  **do**

//Résoudre les équations (CLP)

$\sum_{i \in I} \sigma_{r_i} B_{ij} = v$ , pour tout  $j \in J$

$\sum_{j \in J} A_{ij} \sigma_{c_j} = u$ , pour tout  $i \in I$

//Résoudre ces équations pour vérifier les stratégies mixtes

$\sum_{i=1}^m \sigma_{r_i} = 1$  et  $\sigma_{r_i} \geq 0$  pour tout  $i$

$\sum_{j=1}^n \sigma_{c_j} = 1$  et  $\sigma_{c_j} \geq 0$  pour tout  $j$

**end for**

**end for**

Fin.

Sortie : Liste de tous les équilibres de Nash

---

- Si la valeur est négative alors le joueur gagnant est le joueur, (bénin) donc le nodule est classifiée comme bénin, avec un degré calculé par la moyenne des modèles qui ont classé le nodule comme bénin.

- Si la valeur est positive alors le joueur gagnant est le joueur malin, donc le nodule est classifié comme malin, avec un degré calculé par la moyenne des modèles qui ont classé le nodule comme malin.
- Enfin, si la valeur est nulle, donc la décision n'est pas claire , nous avons calculé le degré moyen de malignité de toutes les stratégies du joueur malin et nous faisons de même pour le joueur bénin, et le résultat est décidé en fonction du maximum des deux valeurs.

### 3.5 Conclusion

Nous avons présenté dans ce chapitre la structure de notre système en détail, ensuite nous avons expliqué les méthodes utilisées ainsi que leurs algorithmes correspondants. Enfin, nous avons exposé le modèle de la théorie des jeux utilisé comme méthode de décision finale dans le cas de conflit.

Dans le 4ème et dernier chapitre, nous allons entamer la partie application et résultats. Nous présenterons l'application réalisée et l'ensemble des résultats obtenus qui seront quantifiés afin d'indiquer le degré d'efficacité atteint par notre système.

# Chapitre 4

## Résultats et implémentation

### 4.1 Introduction

Dans le chapitre précédent, nous avons présenté les différentes étapes suivies pour la réalisation de notre système de segmentation et classification des nodules pulmonaires. Dans ce chapitre, nous citerons les outils qui nous ont aidés à implémenter et à présenter notre solution ainsi que les mesures d'évaluation que nous avons choisies pour vérifier l'efficacité des architectures adoptées.

En premier lieu, nous allons spécifier l'environnement matériel et logiciel pour la conception de notre application ainsi que les bases de données utilisées. Ensuite, nous parlerons de l'interface graphique réalisée et de ses différentes fonctionnalités. Nous exposerons par la suite les tests effectués et discuterons à travers des métriques significatives dans le domaine les résultats obtenus.

### 4.2 Environnements de travail

Dans cette partie, nous allons présenter l'environnement de travail (matériel et logiciel) qui nous a permis de réaliser application.

#### 4.2.1 Environnement matériel

Afin d'implémenter et d'entraîner les modèles d'apprentissage profond pour la segmentation et la classification, nous avons utilisé les machines virtuelles de Kaggle afin d'exploiter les ressources GPU et d'exécuter plusieurs modèles avec différents paramètres afin de trouver la meilleure configuration. D'autre part, nous avons utilisé notre machine pour prétraiter les données et mettre en œuvre l'interface graphique. Les tableaux ci-dessous décrivent toutes les spécifications des machines utilisées.

PC	OS	GPU	Mémoire GPU	Frequence	RAM
Kaggle GPU VM	Linux	Nvidia P100	16 GO	1.32GHz	12 GO

TABLE 4.1 – Kaggle GPU specs

PC	OS	CPU	GPU	RAM
Desktop	Windows 10 64bits	Intel core I5 4590S	NVIDIA GTX 1050 TI 4GO	16 GO
Lenovo	Windows 10 64bits	Intel core I5 4210U	/	4 GO

TABLE 4.2 – Spécifications des machines personnels

## 4.2.2 Environnement logiciel

Pour notre implémentation, nous avons utilisé de nombreuses bibliothèques et outils qui facilitent notre travail en raison de leurs diverses fonctionnalités et de leur flexibilité. Dans cette section, nous mentionnerons les outils qui nous semblent les plus importants et nécessaires pour notre travail.

### 4.2.2.1 Python

Python est un langage de programmation de haut niveau, interprété et polyvalent, couramment utilisé pour le développement de sites Web, l'analyse de données et l'automatisation. Python est un langage à usage général, ce qui signifie qu'il est polyvalent et peut être utilisé pour programmer de nombreux types de fonctions différentes. Comme il s'agit d'un langage interprété, il n'est pas nécessaire de compiler le code avant de l'exécuter et comme il s'agit d'un langage de programmation de haut niveau, Python est capable d'abstraire les détails du code. En fait, Python accorde tellement d'attention à l'abstraction que son code peut être compris par la plupart des programmeurs novices.

Python a une syntaxe simple et est connu pour avoir une grande communauté qui contribue activement à une sélection croissante de modules logiciels et de bibliothèques. Le développement initial de Python a été mené par Guido van Rossum à la fin des années 1980. Aujourd'hui, Python est géré par la "Python Software Foundation" [88].

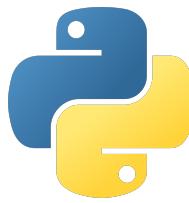


FIGURE 4.1 – Python

### 4.2.2.2 OpenCV

OpenCV est une énorme bibliothèque open-source pour la vision par ordinateur, l'apprentissage automatique et le traitement d'images. Elle joue désormais un rôle majeur dans les opérations en temps réel, ce qui est très important dans les systèmes actuels. En l'utilisant, on peut traiter des images et des vidéos pour identifier des objets, des visages ou même l'écriture d'un être humain. Lorsqu'il est intégré à diverses bibliothèques, telles que NumPy, Python est capable de traiter la structure de tableau OpenCV pour l'analyse. Pour identifier le motif d'une image et ses différentes caractéristiques, nous utilisons l'espace vectoriel et effectuons des opérations mathématiques sur ces caractéristiques [74].



FIGURE 4.2 – OpenCV

#### 4.2.2.3 Pydicom

Pydicom est une bibliothèque purement python pour travailler avec des fichiers DICOM tels que des images médicales, des rapports et des objets de radiothérapie. Pydicom facilite la lecture de ces fichiers complexes en structures python naturelles pour une manipulation aisée. Les ensembles de données modifiés peuvent être réécrits dans des fichiers au format DICOM [100].



FIGURE 4.3 – Pydicom

#### 4.2.2.4 SimpleITK

SimpleITK est une boîte à outils d'analyse d'images comportant un grand nombre de composants permettant d'effectuer des opérations générales de filtrage, de segmentation et d'enregistrement d'images. Il est construit au-dessus de l'Insight Segmentation and Registration Toolkit ITK avec l'intention de fournir une interface simplifiée à ITK. SimpleITK lui-même est écrit en C++ mais est disponible pour un grand nombre de langages de programmation [116].



FIGURE 4.4 – SimpleITK

#### 4.2.2.5 TensorFlow

TensorFlow est une plateforme open source de bout en bout pour l'apprentissage automatique. Elle dispose d'un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires qui permet aux chercheurs de faire progresser l'état de l'art de l'apprentissage automatique et aux développeurs de créer et de déployer facilement des applications basées sur l'apprentissage automatique [118].



FIGURE 4.5 – TensorFlow

#### 4.2.2.6 Keras

Keras est un API d'apprentissage profond de haut niveau développée par Google pour la mise en œuvre de réseaux neuronaux. Elle est écrite en Python et est utilisée pour faciliter la mise en œuvre de réseaux neuronaux. Il prend également en charge le calcul de plusieurs réseaux neuronaux en arrière-plan.

Keras est relativement facile à apprendre et à utiliser parce qu'il fournit un frontal Python avec un haut niveau d'abstraction tout en ayant la possibilité d'utiliser plusieurs backends pour le calcul. Cela rend Keras plus lent que d'autres cadres d'apprentissage profond, mais extrêmement convivial pour les débutants [75].



FIGURE 4.6 – Keras

#### 4.2.2.7 Scikit-learn

Scikit-learn (Sklearn) est la bibliothèque la plus utile et la plus robuste pour l'apprentissage automatique en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité, via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en Python, s'appuie sur NumPy, SciPy et Matplotlib [114].



FIGURE 4.7 – Scikit-learn

#### 4.2.2.8 Nashpy

La bibliothèque de théorie de jeux Nashpy est une bibliothèque open source du langage de programmation python qui est destinée aux calculs des équilibres dans les jeux stratégiques de deux joueurs. Crée par Vince Knight [117], mathématicien à l'université de Cardiff.

#### 4.2.2.9 PyQt

PyQt est une boîte à outils de widgets GUI. Il s'agit d'une interface Python pour Qt, l'une des bibliothèques d'interface graphique multiplateforme les plus puissantes et les plus populaires. PyQt est un mélange du langage de programmation Python et de la bibliothèque Qt. Ce tutoriel d'introduction vous aidera à créer des applications graphiques à l'aide de PyQt [113].



FIGURE 4.8 – PyQt

#### 4.2.2.10 Visual Studio Code

Visual Studio Code est un éditeur de code source léger mais puissant qui fonctionne sur votre bureau et est disponible pour Windows, macOS et Linux. Il offre une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages et moteurs d'exécution (tels que C++, C#, Java, Python, PHP, Go, .NET) [71].

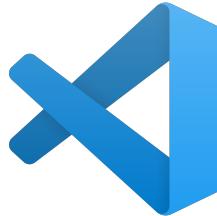


FIGURE 4.9 – Visual Studio Code

### 4.3 Les bases de données utilisées

Les bases de données utilisées sont essentiellement constituées de scanners pulmonaires en 3D. Chaque scan est une pile d'images 2D triées et déposées à chaque profondeur du poumon. Il existe quelques bases de données publiques , dans notre cas nous en avons utilisé deux : LUNA16 [104] et DSB 2017 [98].

La raison pour laquelle nous avons choisi ces deux jeux de données est la disponibilité des annotations pour la segmentation et la classification, contrairement aux autres jeux de données qui sont rarement annotés.

#### 4.3.1 LUNA16

LUNA 16 est l'abréviation de "Lung Nodule Analysis 2016", un concours qui s'est déroulé en 2016. Dans ce concours, les organisateurs fournissent aux concurrents un ensemble de données à utiliser et à évaluer leur travail avec. Cet ensemble de données est considéré comme un sous-ensemble du plus grand ensemble de données publiquement disponible pour

les nodules pulmonaires LIDC/IDRI [14][39].

Le jeu de données LUNA16 supprime les tomodensitogrammes dont l'épaisseur de coupe est supérieure à 3 mm, l'espacement des coupes est incohérent ou des coupes manquent dans le jeu de données LIDC-IDRI. patients annotés par des radiologues [42]. Le jeu de données LUNA16 contient environ 888 scans et un total de 477 nodules.

### 4.3.2 DSB 2017

DSB 17 est l'acronyme de "Data Sciecne Bowl 2017", un autre concours similaire à LUNA avec son jeu de données. Contrairement à LUNA, ce jeu de données annoté pour la classification ne comprend que les annotations binaires par sujet indiquant si celui-ci a été diagnostiqué comme un cancer du poumon.

Le jeu de données DSB comprend 1397, 198, 506 personnes (cas) et contient environ 753 nodules dans son entraînement, sa validation et son évaluation [98].

### 4.3.3 Différence entre DSB et LUNA

Il existe quelques différences significatives entre les nodules LUNA et les nodules DSB. Le jeu de données LUNA comporte de nombreux nodules annotés de très petite taille, qui peuvent être sans rapport avec le cancer. Selon l'expérience des médecins [40], les nodules de moins de 6 mm ne sont généralement pas dangereux. En revanche, le jeu de données DSB comporte de nombreux nodules de très grande taille (plus de 40 mm). Le diamètre moyen des nodules est de 13,68 mm dans le jeu de données DSB et de 8,31 mm dans le jeu de données LUNA. En outre, le jeu de données DSB présente de nombreux nodules sur la branche principale, ce qui est rarement le cas dans le jeu de données LUNA. Si le réseau est entraîné uniquement sur l'ensemble de données LUNA, il sera difficile de détecter les nodules dans l'ensemble de données DSB. L'absence de gros nodules conduirait à des prédictions de cancer incorrectes, car l'existence de gros nodules est une caractéristique des patients atteints de cancer. Pour faire face à ces problèmes, nous retirons les nodules de moins de 6 mm des annotations de LUNA et nous avons cherché une annotation manuelle pour les nodules DSB afin de combiner les deux ensembles de données et de les utiliser pour entraîner notre modèle de segmentation.

Pour la classification, l'ensemble de données LUNA16 ne contient pas d'annotations sur le cancer. Nous avons donc utilisé uniquement l'ensemble de données DSB pour la classification des nodules.

## 4.4 Présentation de l'application

Dans cette partie, nous allons présenter les interfaces de notre application et ses différentes fonctionnalités.

### 4.4.1 Écran d'accueil

La figure 4.10 représente le premier écran de notre application, il est considéré comme une page de bienvenue qui a deux boutons : le bouton "About" qui redirige l'utilisateur vers la page d'informations, le bouton "Get started" qui le redirige vers la page suivante de chargement des scans.

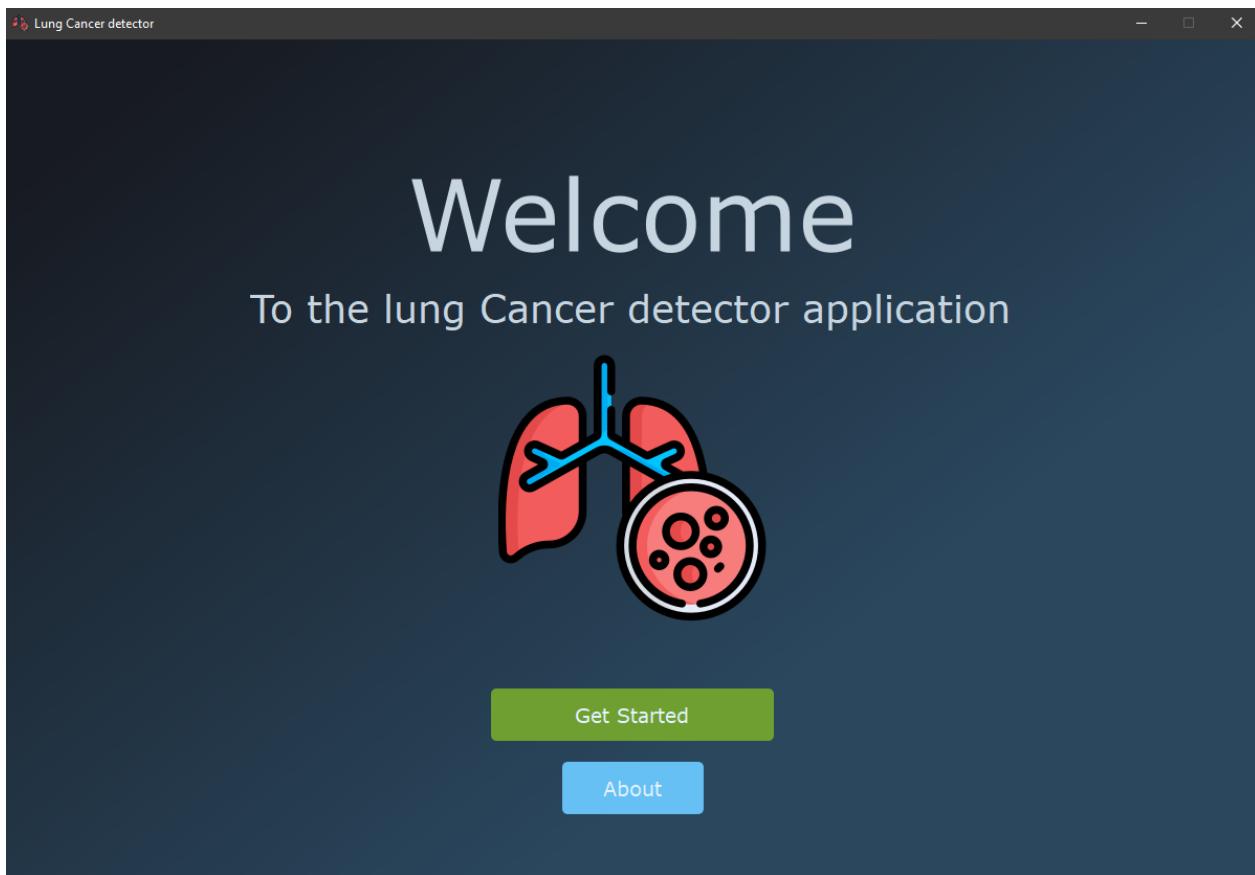


FIGURE 4.10 – Écran d'accueil

### 4.4.2 Écran d'information

Cet écran présente toutes les informations liées à l'application à partir de son objectif, avec des instructions et les conditions requises pour utiliser cette application, ainsi que des informations de contact pour les suggestions et les retours des utilisateurs.

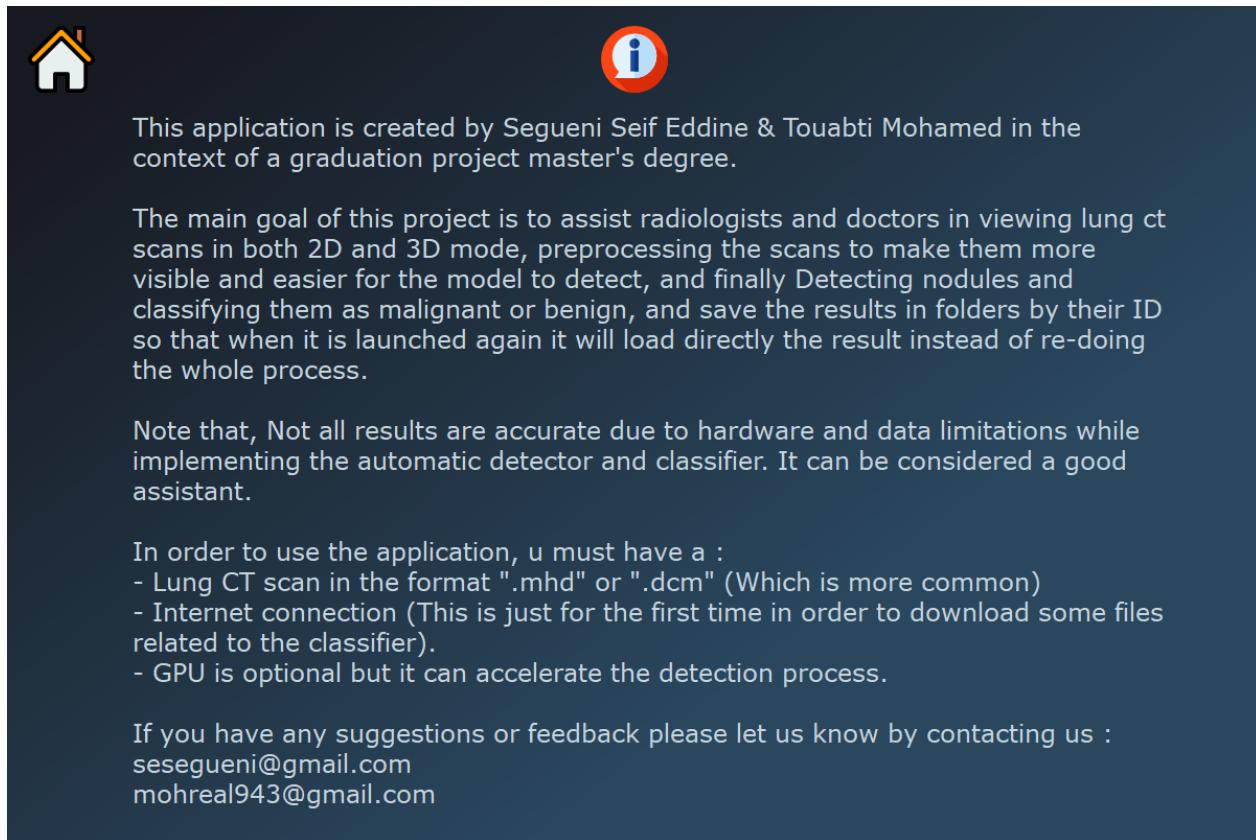


FIGURE 4.11 – Écran d'informations

#### 4.4.3 Écran de chargement des scans

Dans cet écran, l'utilisateur devra choisir le type de fichier à utiliser, soit des fichiers "mhd" ou des fichiers "dicom", chaque type a son propre bouton. Une fois que l'utilisateur a cliqué sur l'un des boutons, une boîte de dialogue s'affiche pour lui permettre de choisir le fichier ou le dossier désiré à partir de son stockage. Un bouton d'écran d'accueil est également ajouté afin de retourner à l'écran d'accueil.

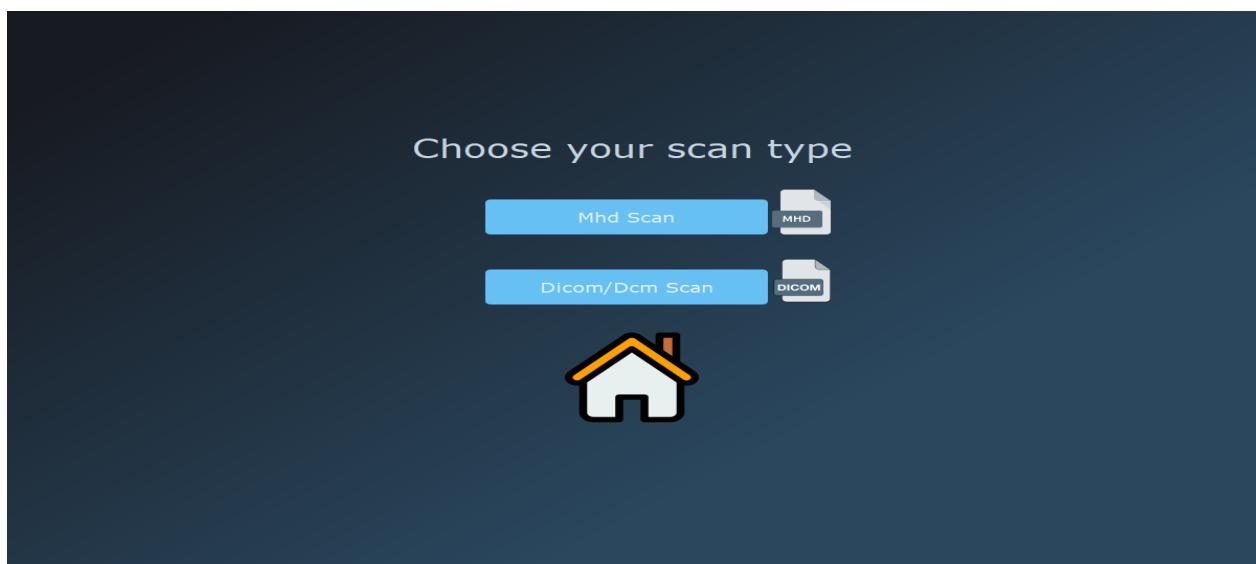


FIGURE 4.12 – Écran de chargement des scans

#### 4.4.4 Écran de visualisation du scan

Une fois que l'utilisateur a choisi les données avec lesquelles il veut travailler, un écran de chargement apparaît, ce qui signifie que notre système est en train de charger le fichier et les métadonnées nécessaires pour le visualiser. L'utilisateur peut interrompre le chargement et revenir à l'écran précédent en appuyant sur le bouton "Annuler" comme le montre la figure 4.13 :

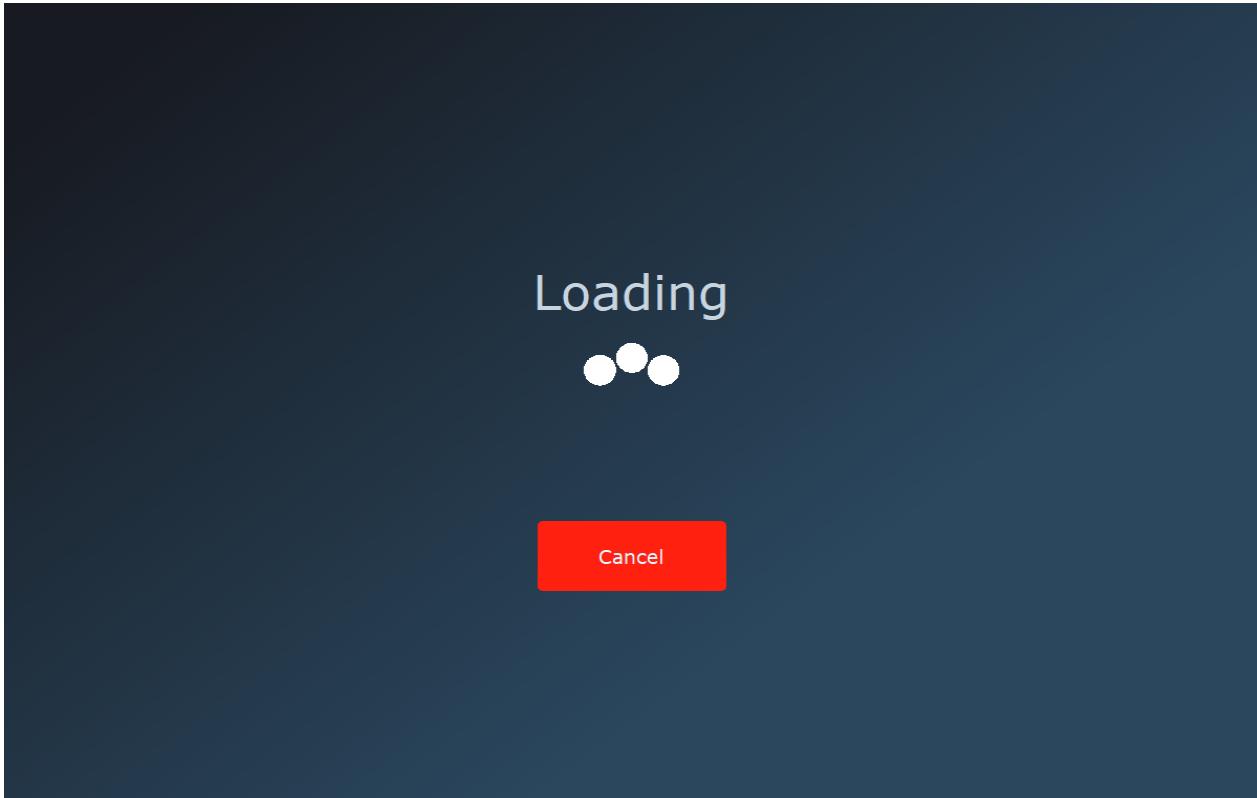


FIGURE 4.13 – Écran de chargement d'un scan

Une fois le chargement terminé, l'écran de visualisation du scan apparaît. Cet écran comporte 4 éléments principaux : La barre d'outils, le visualiseur de scan, le tableau des métadonnées et le bouton de prétraitement. Voici un aperçu global de l'écran :

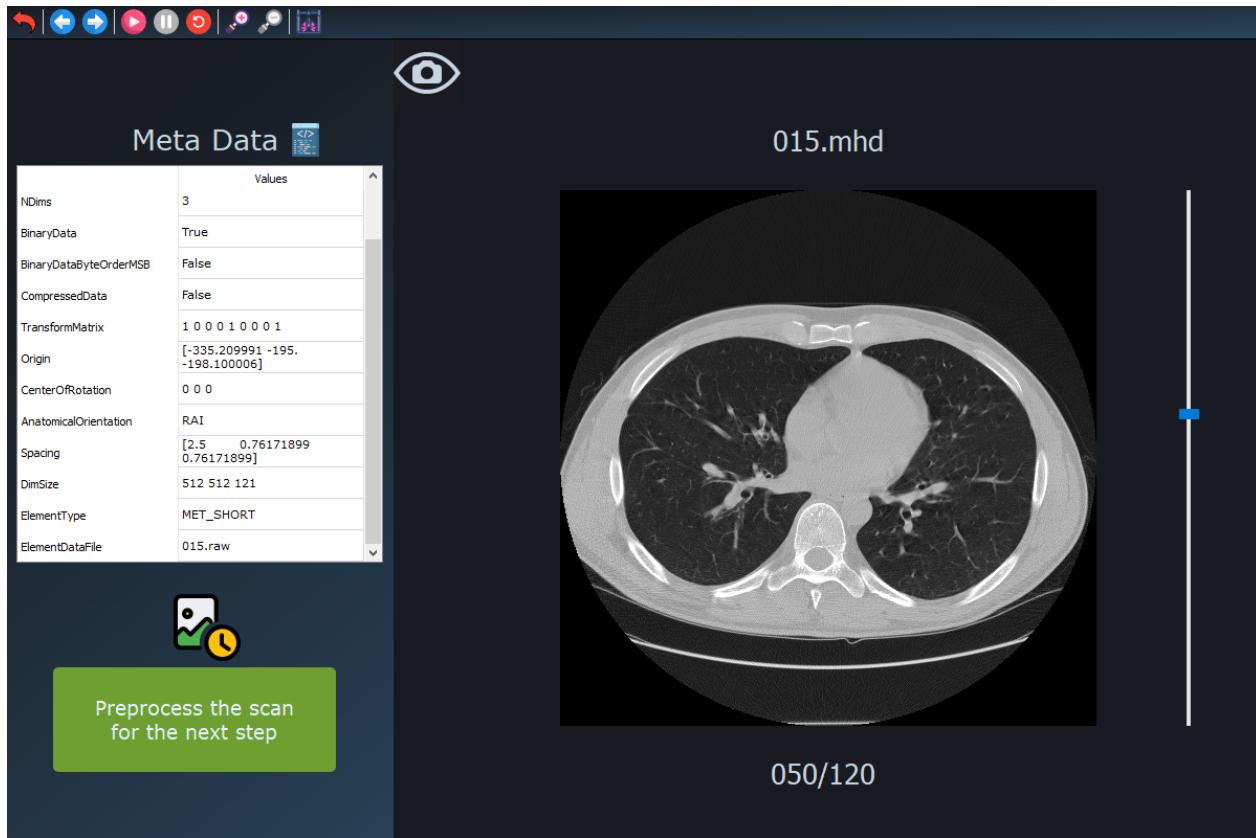


FIGURE 4.14 – Écran de visualisation du scan

#### 4.4.4.1 La barre d'outils

La barre d'outils est située dans la partie supérieure de presque chaque écran de l'application. Elle contient de nombreux boutons d'action, chacun ayant sa propre fonctionnalité. Voici un aperçu de la barre d'outils :

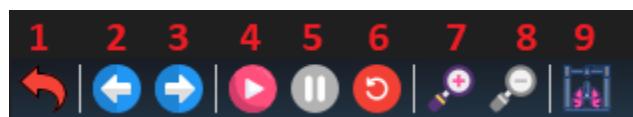


FIGURE 4.15 – La barre d'outils

Comme nous l'avons mentionné, chacun de ces boutons a sa propre fonctionnalité, nous allons détailler chacun d'entre eux :

- Bouton 1 : Un bouton de retour, afin de revenir à l'écran précédent.
- Bouton 2 : Afficher la tranche précédente du scan .
- Bouton 3 : Afficher la tranche suivante du scan.
- Bouton 4 : Démarrer la lecture automatique de toutes les tranches du scan (mode vidéo).
- Bouton 5 : Mettre en pause la lecture automatique.
- Bouton 6 : Remise à la première tranche du scan.
- Bouton 7 : Faites un zoom de 25% sur le point cliqué dans le scan.
- Bouton 8 : Réinitialiser le zoom appliqué (Ré-initialisation de l'échelle).
- Bouton 9 : Ouvrir l'écran de visualisation du modèle 3D.

#### 4.4.4.2 Le tableau des métadonnées

Cette table contient les métadonnées liées au scan. ces données peuvent être trouvées dans le scan lui-même. Elles diffèrent d'un scan à l'autre, et d'un type de fichier à l'autre également, mais en général elles contiennent beaucoup de données importantes qui peuvent être utiles pour analyser le patient ou le scan. voici un exemple de table de métadonnées :

	Values
NDims	3
BinaryData	True
BinaryDataByteOrderMSB	False
CompressedData	False
TransformMatrix	1 0 0 0 1 0 0 0 1
Origin	[-335.209991 -195. -198.100006]
CenterOfRotation	0 0 0
AnatomicalOrientation	RAI
Spacing	[2.5 0.76171899 0.76171899]
DimSize	512 512 121
ElementType	MET_SHORT
ElementDataFile	015.raw

FIGURE 4.16 – Table de métadonnées

#### 4.4.4.3 Le visualiseur de scan

Cette partie de l'écran agit comme une visionneuse d'images qui montre chaque tranche du scan. l'utilisateur peut naviguer en utilisant les boutons gauche et droite du clavier ou les boutons mentionnés dans la barre d'outils ou le curseur vertical situé à droite de l'image. Il montre également en haut le nom du fichier et en bas l'index actuel avec le nombre total de tranches du scan.

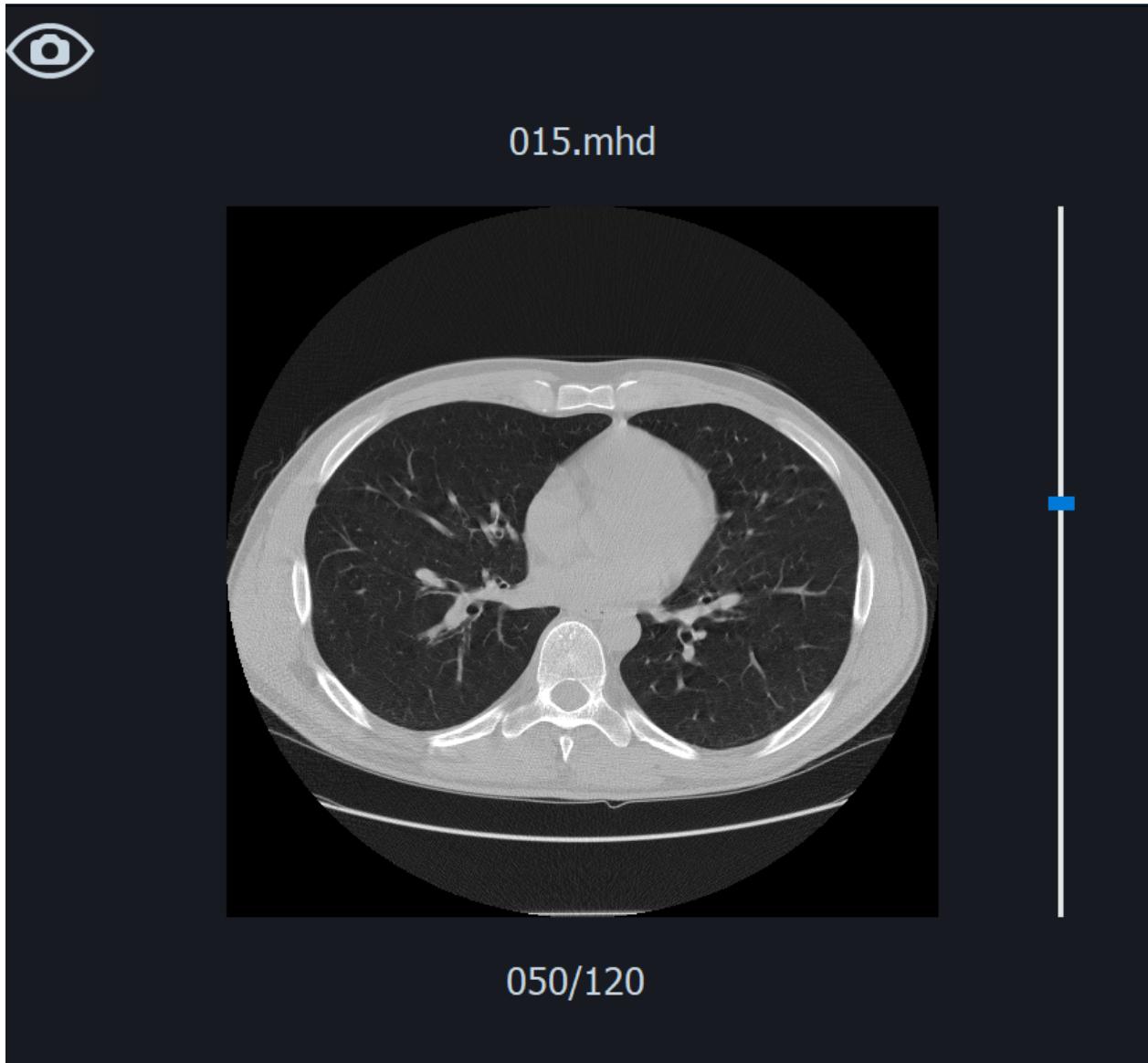


FIGURE 4.17 – Le visualiseur de scan

#### 4.4.4.4 Le bouton de pré-traitement

Ce bouton permet d'initier la phase de prétraitement et de rediriger vers l'écran suivant.

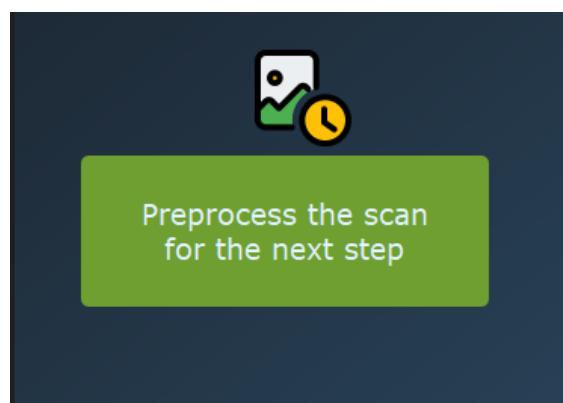


FIGURE 4.18 – Le bouton de pré-traitement

#### 4.4.5 Page de visualisation du scanner pré-traité

Comme dans l'écran précédent, un écran de chargement apparaîtra d'abord pendant que le système pré-traite le scan, ce qui peut prendre un certain temps selon le type de fichier et le nombre de tranches dans le scan, donc un bon CPU et RAM sont nécessaires pour accélérer l'opération.

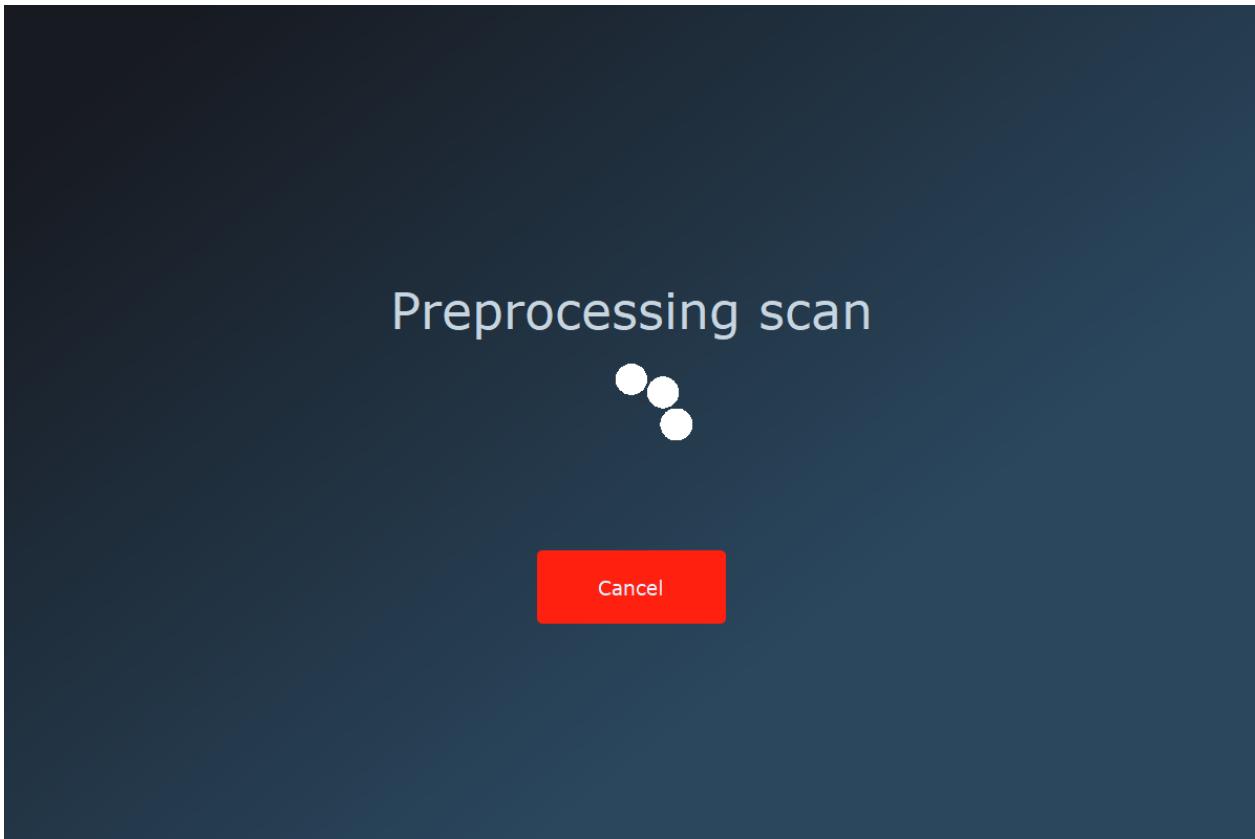


FIGURE 4.19 – Écran de chargement du pré-traitement du scan.

Une fois que c'est fait, un écran semblable au précédent apparaît, mais sans le tableau de métadonnées et avec un bouton différent. Dans la visionneuse d'images, le résultat du préprocesseur sera affiché et vous pourrez naviguer comme dans l'écran précédent. Le bouton "Déetecter le nodule" lance le processus de détection et de classification et redirige vers l'écran suivant.

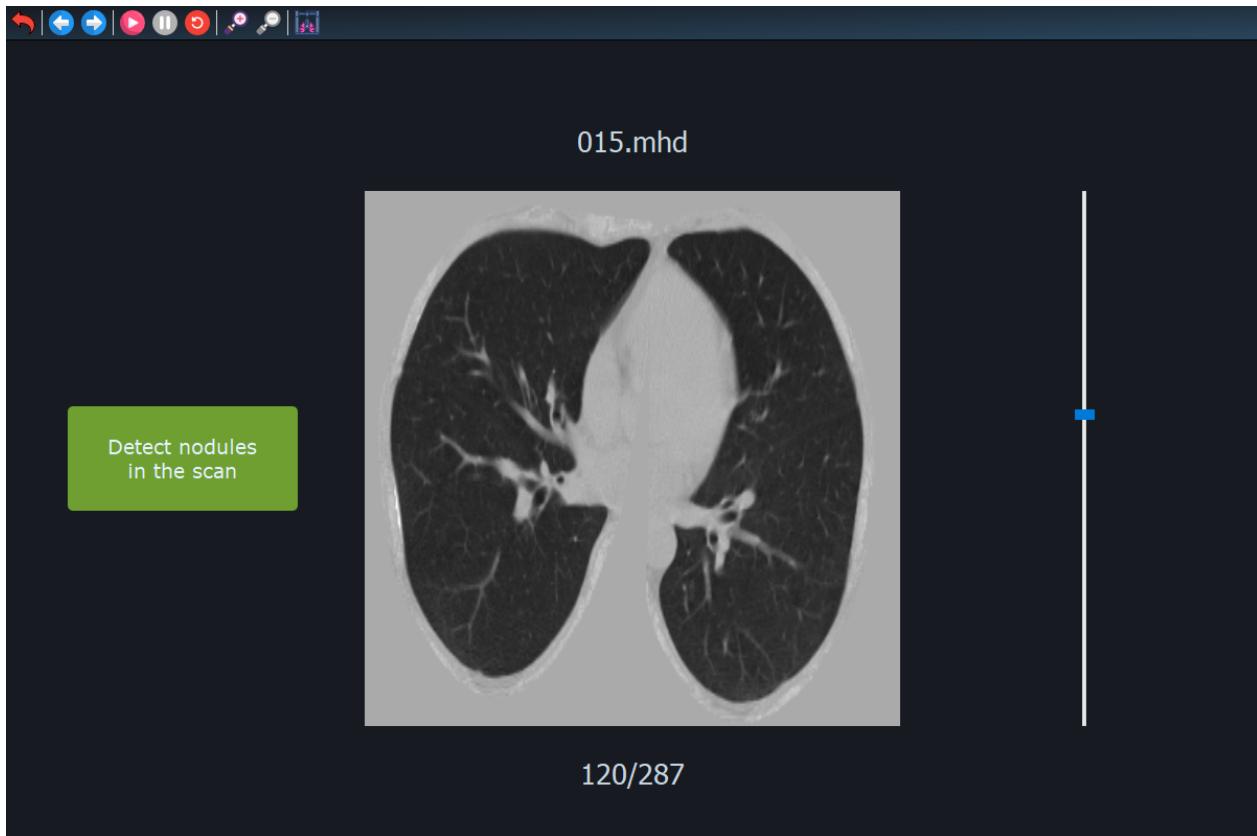


FIGURE 4.20 – Écran de visualisation du scan pré-traité

#### 4.4.6 Page des résultats finaux

Une fois que la phase de détection et de classification commence, un écran de chargement apparaîtra comme les précédents et durera jusqu'à la fin de l'opération, ce qui peut prendre un certain temps selon la profondeur de l'analyse. Cette phase recommande fortement l'utilisation d'un GPU afin de réduire le temps d'attente estimé.

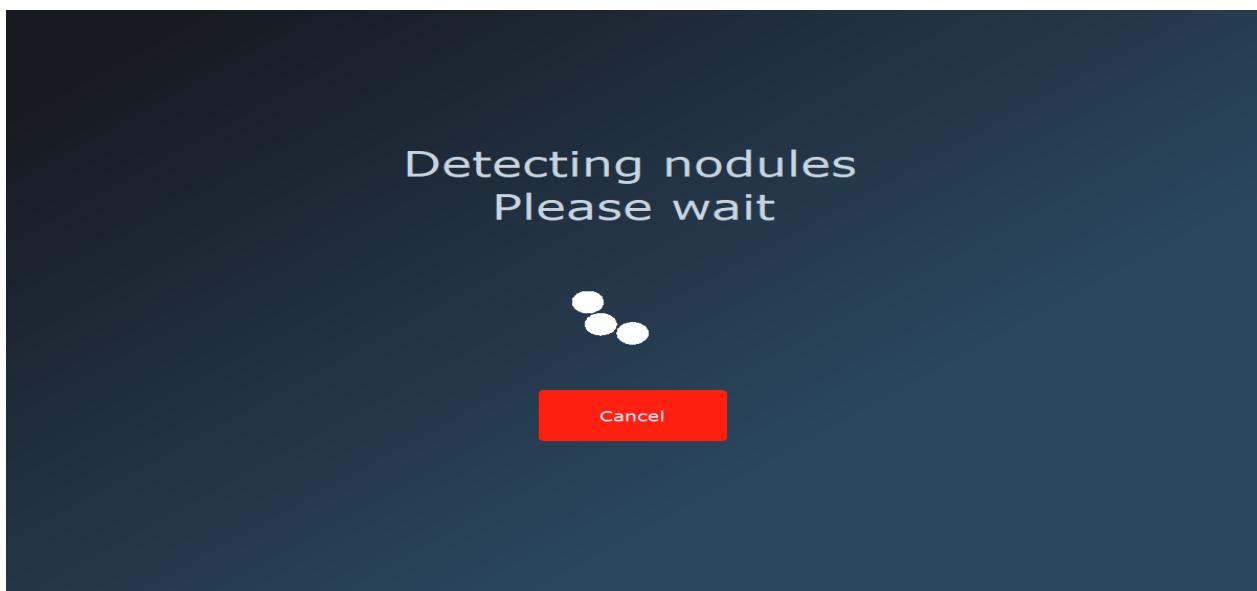


FIGURE 4.21 – Écran de chargement de la phase de détection

Une fois la détection effectuée, un écran apparaîtra avec la même barre d'outils et la même partie de scanner avec les mêmes fonctionnalités mais au lieu de la table de métadonnées, il contient deux nouvelles tables : La table de la liste des nodules et la table des informations sur les nodules, ainsi qu'un petit visualiseur d'images du nodule. La figure ci-dessous montre une vue d'ensemble de l'écran dans le cas où il ya des nodules.

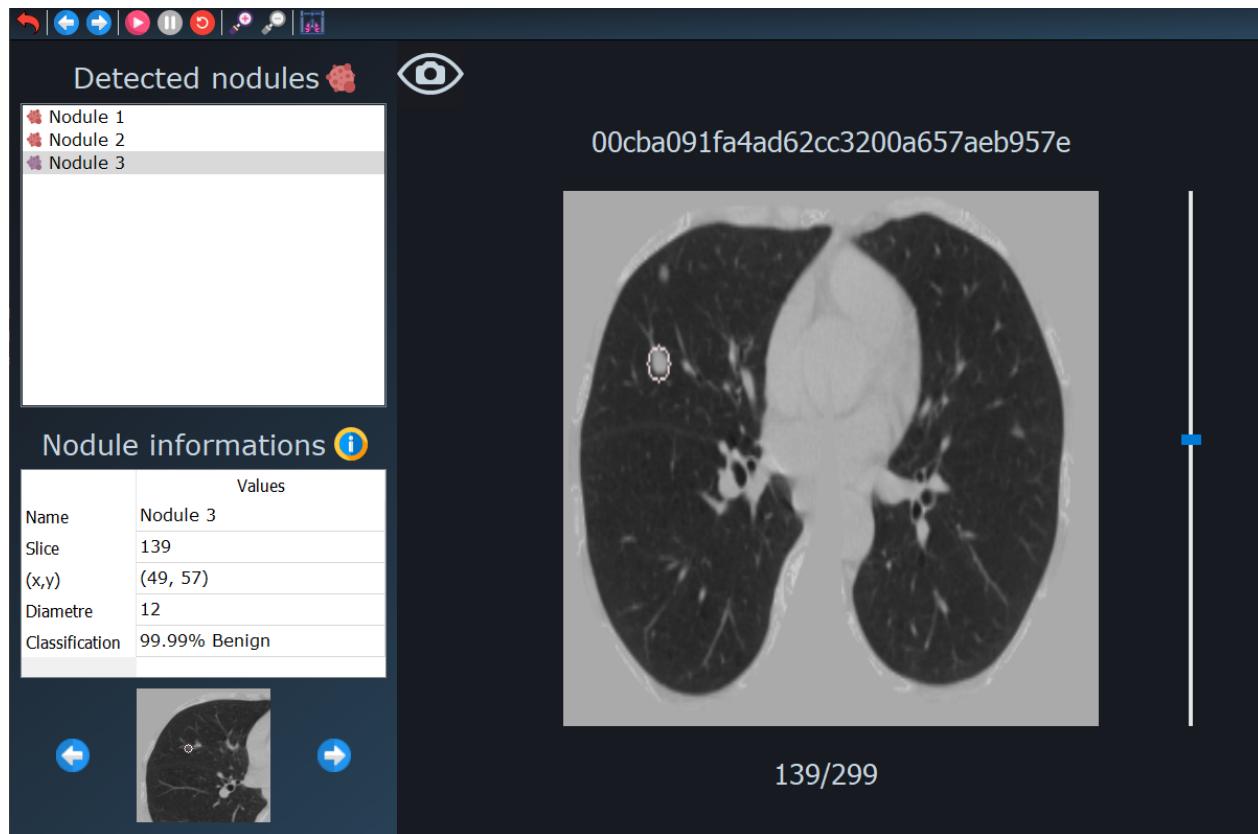


FIGURE 4.22 – Écran de visualisation du résultats de détection

Dans le cas où le système ne détecte aucun nodule, les tableaux et le petit visualiseur d'images n'apparaîtront pas mais il ressemblera à ceci :

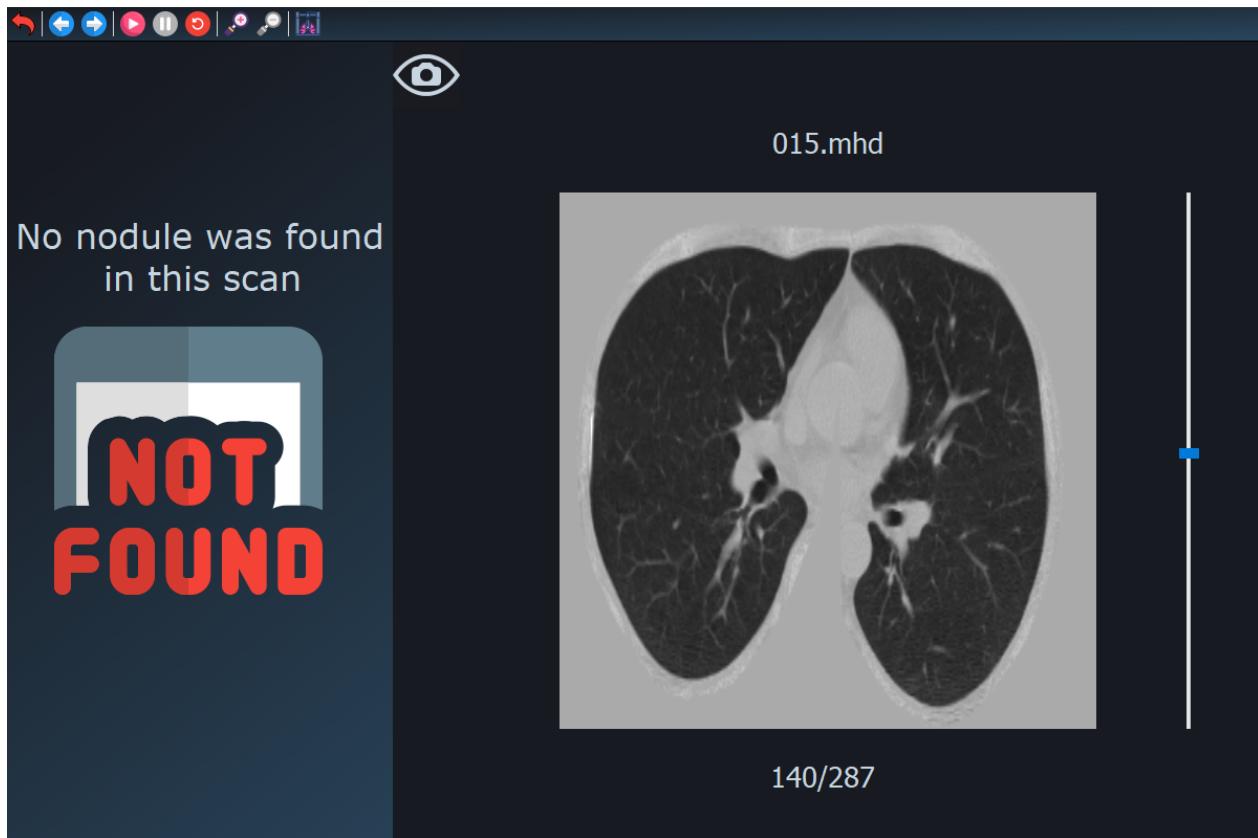


FIGURE 4.23 – Écran de visualisation du résultats de détection lorsque il n'y a pas de nodule détecté

#### 4.4.6.1 Tableau des nodules détectés

Ce tableau contient une liste de tous les nodules détectés. Une fois que l'utilisateur clique sur des nodules, les informations respectives s'afficheront dans le tableau ci-dessous, ainsi que l'image de nodule en bas.

Detected nodules	
	Nodule 1
	Nodule 2
	Nodule 3

FIGURE 4.24 – Tableau des nodules détectés

#### 4.4.6.2 Tableau d'informations sur les nodules

Ce tableau contient les informations sur les nodules sélectionnés telles que la position et la classification de sa malignité et la taille.

Nodule informations 	
	Values
Name	Nodule 3
Slice	139
(x,y)	(49, 57)
Diametre	12
Classification	99.99% Benign

FIGURE 4.25 – Tableau des informations du nodule

#### 4.4.6.3 Visualiseur de blocs de nodules

Il montre une image recadrée du nodule afin qu'il puisse être mieux analysé, et il peut également être navigué à l'aide des boutons gauche et droit à côté de celui-ci.

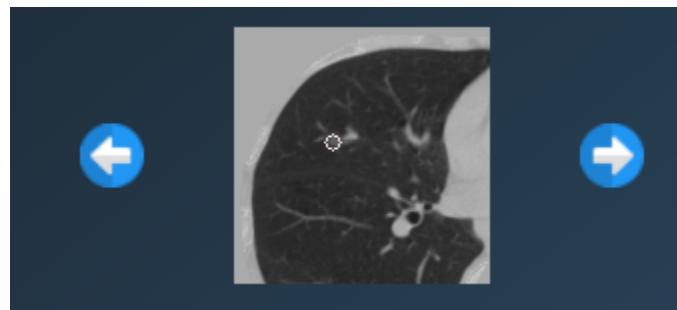


FIGURE 4.26 – Visualiseur de blocs du nodule

#### 4.4.7 Écran de visualisation du modèle 3D

Cet écran apparaît lorsque l'utilisateur appuie sur le bouton du visualiseur de modèle 3D situé dans la barre d'outils. Il ouvrira une nouvelle fenêtre qui affiche le modèle 3D de la numérisation et peut zoomer et dézoomer à l'aide de la molette de la souris.

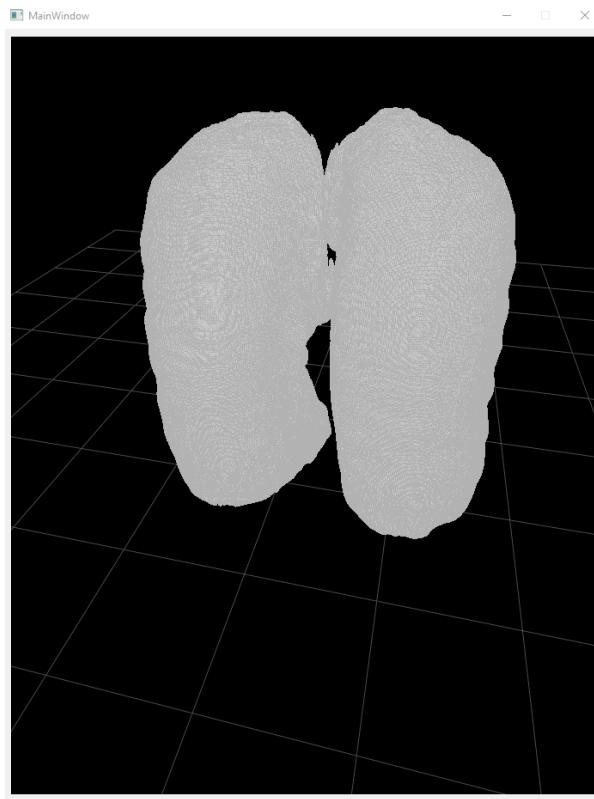


FIGURE 4.27 – Fenêtre de visualisation 3D

### 4.5 Expérimentation et validation

Dans cette partie, nous présenterons les différents résultats obtenus lors du test des modèles de segmentation et de classification des nodules pulmonaires, ainsi que tous les critères d'évaluation utilisés pour obtenir ces résultats, puis nous montrerons quelques exemples de détection et de classification des nodules.

#### 4.5.1 Les critères d'évaluation

Les métriques d'évaluation sont utilisées pour mesurer la qualité du modèle statistique ou d'apprentissage automatique. L'évaluation des modèles ou des algorithmes d'apprentissage automatique est essentielle pour tout projet. Il existe de nombreux types de métriques d'évaluation disponibles pour tester un modèle [53]. Voici les différentes métriques que nous avons utilisées :

#### 4.5.1.1 Accuracy

L'accuracy est une métrique d'évaluation qui vous permet de mesurer le nombre total de prédictions correctes d'un modèle [51]. La formule pour L'accuracy est la suivante :

$$\text{Accuracy} = \frac{\text{Vrais Négatifs} + \text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs} + \text{Vrais Négatifs} + \text{Faux Négatifs}} \quad (4.1)$$

#### 4.5.1.2 Précision

La précision est le rapport entre les vrais positifs et tous les positifs [51] et elle évalue la précision d'un modèle à prédire des étiquettes positives [81].

$$\text{Précision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}} \quad (4.2)$$

#### 4.5.1.3 Rappel (Sensibilité)

Le rappel calcule le taux des vrais positifs qu'un modèle a correctement identifiés [51].

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \quad (4.3)$$

#### 4.5.1.4 Spécificité

La spécificité ou le taux des vrais négatifs est la mesure qui évalue la capacité d'un modèle à prédire les vrais négatifs [58].

$$\text{Spécificité} = \frac{\text{Vrais Négatifs}}{\text{Vrais Négatifs} + \text{Faux Positifs}} \quad (4.4)$$

#### 4.5.1.5 La Matrice de confusion

Une matrice de confusion, également appelée matrice d'erreurs, est un tableau récapitulatif utilisé pour évaluer les performances d'un modèle de classification. Le nombre de prédictions correctes et incorrectes est résumé par des valeurs de comptage et ventilé pour chaque classe [85].

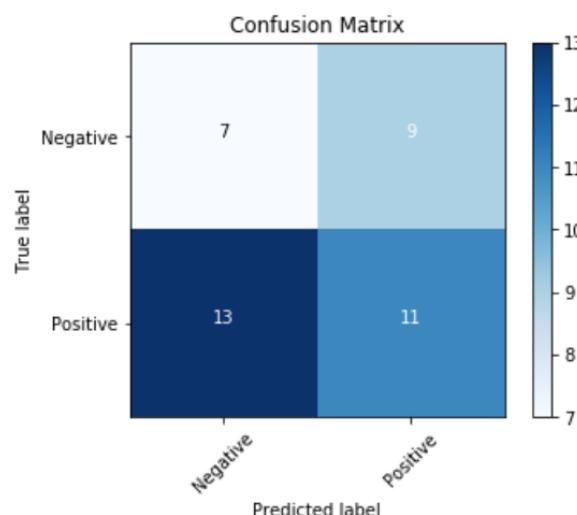


FIGURE 4.28 – Exemple d'une matrice de confusion [51]

Une matrice de confusion affiche le nombre de vrais positifs, de faux positifs, de vrais négatifs et de faux négatifs produits par un modèle. En utilisant une matrice de confusion, nous pouvons obtenir les valeurs nécessaires pour calculer la précision d'un modèle [51].

#### 4.5.1.6 F-Measure

Le F-Mesure , également appelé F1-score, est une mesure de la précision d'un modèle sur un ensemble de données. Il est utilisé pour évaluer les systèmes de classification binaire, qui classent les exemples en "positifs" ou "négatifs".

La formule du F1-score standard est la moyenne harmonique de la précision et du rappel. Un modèle parfait a un F-Mesure de 1.

$$\begin{aligned} \text{F-Measure} &= \frac{2}{\frac{1}{\text{Rappel}} + \frac{1}{\text{Précisions}}} \\ &= 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \end{aligned} \quad (4.5)$$

Et il pourrait être représenté aussi par cette formule :

$$\text{F-Measure} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \frac{1}{2}(\text{Faux Positifs} + \text{Faux Négatifs})} \quad (4.6)$$

#### 4.5.1.7 Courbe ROC et AUROC

L'analyse ROC (**R**eceiver **O**perating **C**harteristic) est une approche graphique permettant d'analyser les performances d'un classificateur. Elle utilise une paire de statistiques - le taux de vrais positifs et le taux de faux positifs - pour caractériser les performances d'un classificateur.

Les statistiques sont reportées sur un graphique à deux dimensions, avec le taux de faux positifs sur l'axe des x et le taux de vrais positifs sur l'axe des y. Le graphique obtenu peut être utilisé pour comparer les performances relatives de différents classificateurs et pour déterminer si un classificateur est plus performant qu'une estimation aléatoire [12].

AUC (**A**rea **U**nder **T**he **C**urve) Courbe ROC. C'est l'une des mesures d'évaluation les plus importantes pour vérifier les performances de tout modèle de classification. Elle s'écrit également AUROC (Area Under the Receiver Operating Characteristics). Plus l'AUC est élevé, plus le modèle est capable de distinguer les classes [73].

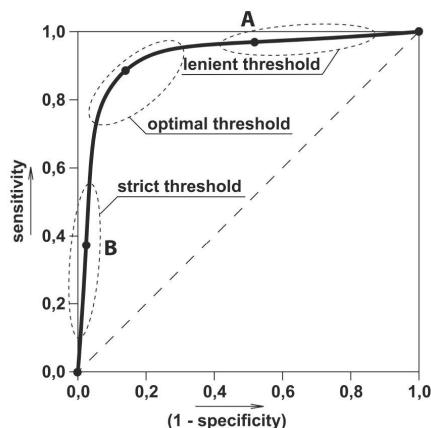


FIGURE 4.29 – Courbe ROC [77]

#### 4.5.1.8 Courbe de précision-rappel

La courbe précision-rappel est construite en calculant et en traçant la précision par rapport au rappel pour un classificateur unique à une variété de seuils [66].

Une AUC élevée représente à la fois un rappel et une précision élevés et un bon classificateur maintiendra à la fois une précision et un rappel élevés sur le graph.

La courbe Precision-Recall est meilleure qu'une courbe ROC pour calibrer le seuil de probabilité dans un scénario de classe déséquilibrée [43].

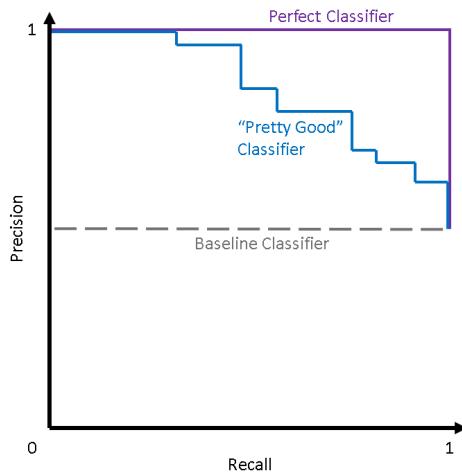


FIGURE 4.30 – Quelques courbes théoriques de précision-rappel [66]

#### 4.5.1.9 Erreur absolue moyenne(MAE)

L'erreur absolue moyenne est une métrique d'évaluation de modèle. L'erreur absolue moyenne d'un modèle par rapport à un ensemble de tests est la moyenne des valeurs absolues des erreurs de prédiction individuelles sur toutes les instances de l'ensemble de tests. Chaque erreur de prédiction est la différence entre la valeur réelle et la valeur prédictive pour l'instance.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \lambda(x_i)|}{n} \quad (4.7)$$

où  $y_i$  est la valeur cible réelle pour l'instance de test  $x_i$ ,  $\lambda(x_i)$  est la valeur cible prédictive pour l'instance de test  $x_i$ , et  $n$  est le nombre d'instances de test [15].

#### 4.5.1.10 Intersection-Over-Union

L'intersection-sur-union, également appelée indice de Jaccard, est une mesure d'évaluation courante pour la segmentation sémantique des images, qui nous permet de quantifier le pourcentage de chevauchement entre le masque cible et notre sortie de prédiction. Pour faire simple, la métrique IoU mesure le nombre de pixels communs entre le masque cible et le masque de prédiction, divisé par le nombre total de pixels présents dans les deux masques

[45].

$$\text{IoU} = \frac{\text{Target} \cap \text{Prediction}}{\text{Target} \cup \text{Prediction}} \quad (4.8)$$

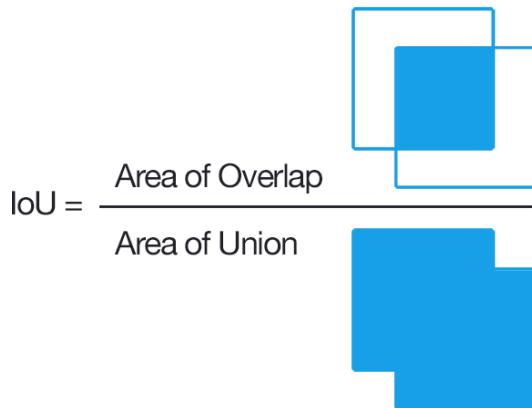


FIGURE 4.31 – Visualisation du calcul de l’IoU [82]

#### 4.5.1.11 Dice Coefficient

En termes simples, le coefficient Dice est égal à  $2 * \text{la zone de chevauchement}$  divisée par le nombre total de pixels dans les deux masques cible et de prédiction [68].

Dice coefficient est la métrique la plus utilisée pour valider les segmentations du volume médical.

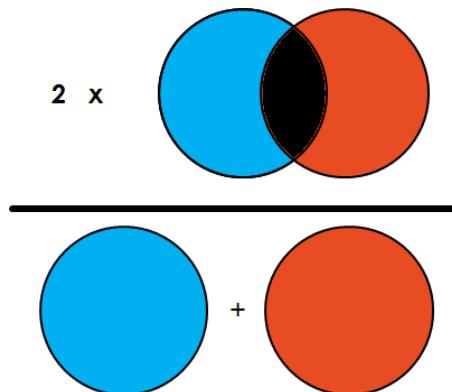


FIGURE 4.32 – Illustration du Dice Coefficient [68]

#### 4.5.2 L'évaluation des modèle de la segmentation

Après l'entraînement des modèles, les poids finaux de chaque modèle sont chargés et utilisés pour l'évaluer sur les ensembles de test, l'évaluation pour chacun des modèles 2d et 3d est représentée sur le tableau suivant :

	Précision	Rappel	F-measure	IoU	Dice Coef
2d- Unet	0.88	0.64	0.74	0.64	0.78
3d- Unet	0.83	0.71	0.76	0.68	0.80

TABLE 4.3 – Tableau récapitulatif des résultats de la segmentation 2d et 3d en utilisant des métriques différentes

#### Discussions

- D'après le tableau, nous remarquons que le modèle 2d-Unet à une précision de 0,88 et 3d u-net de 0,83, ce qui signifie que 2d-Unet est plus efficace pour détecter correctement les nodules et détecter moins de nodules faux positifs que le modèle 3d-Unet, mais 3d-Unet à un meilleur rappel de 0,71, ce qui signifie qu'il a réussi à identifier 71% de tous les vrais nodules. Et le meilleur score F-mesure était de 0,76 pour le modèle 3d-Unet.
- Le modèle 3d-Unet a également obtenu le meilleur IoU score de 0,68 et le meilleur dice coefficient de 0,80.

A partir de ces résultats, nous déduisons que le modèle 3d-Unet est meilleur que 2d-Unet pour la segmentation des nodules pulmonaires.

#### 4.5.2.1 Affichage des Résultats de la segmentation 2d et 3d

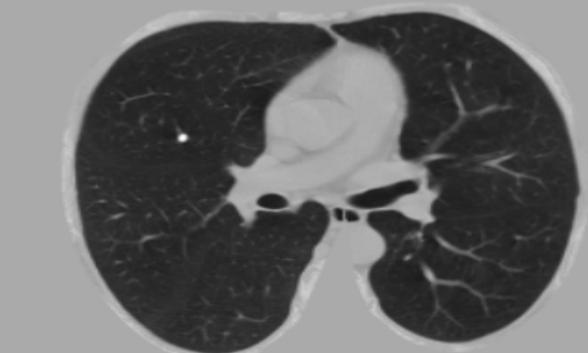
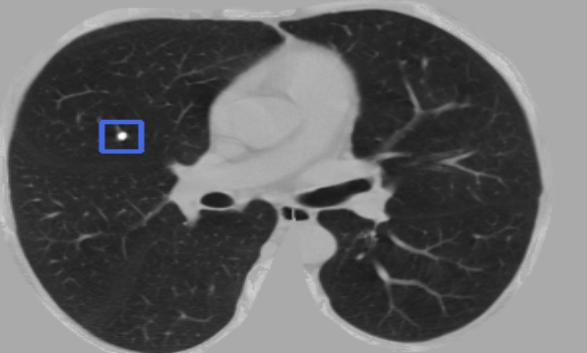
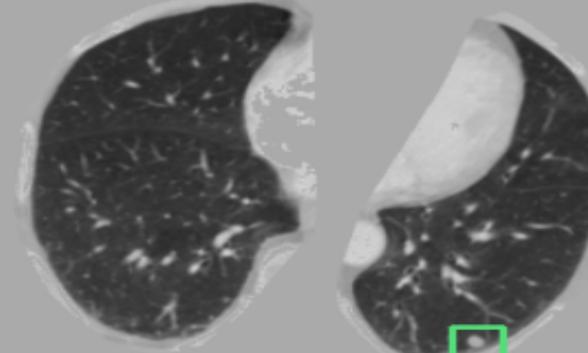
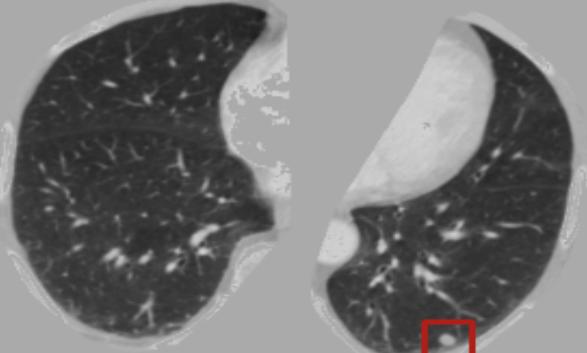
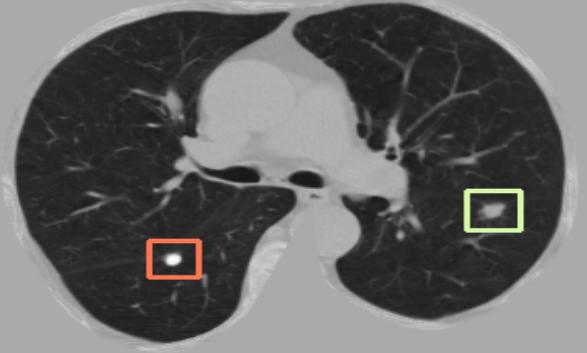
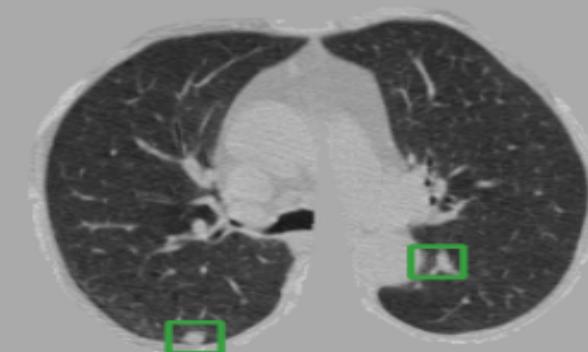
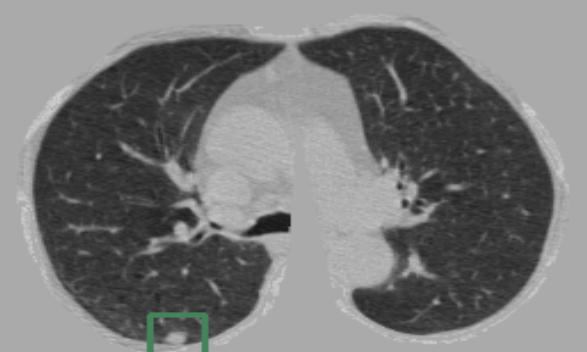
Résultats de la segmentation 2D	Résultats de la segmentation 3D
	
	
	
	

TABLE 4.4 – Tableau récapitulatif d'exemples de la segmentation 2d et 3d pour des scans aléatoires

### 4.5.3 Évaluation des résultats de la classification

Comme nous l'avons mentionné dans le chapitre précédent, les données ont été divisées 4 fois, ce qui donne 4 ensembles d'entraînement et de test.

Chaque modèle d'apprentissage profond a été entraîné sur tous les ensembles d'entraînement et évalué sur l'ensemble du test correspondant et également une évaluation moyenne des tests est calculée pour chaque métrique. Une autre évaluation pour l'ensemble complet de données est calculée.

Les résultats sont présentés dans un seul tableau qui contient les résultats de test moyen :

	Accuracy	Précision	Rappel	spécificité	F-Mesure	MAE
DenseNet121	<b>0.73</b>	<b>0.76</b>	<b>0.74</b>	<b>0.72</b>	<b>0.75</b>	<b>0.27</b>
ResNet50	<b>0.73</b>	<b>0.73</b>	<b>0.77</b>	<b>0.64</b>	<b>0.75</b>	<b>0.28</b>
InceptionV3	<b>0.74</b>	<b>0.76</b>	<b>0.77</b>	<b>0.69</b>	<b>0.76</b>	<b>0.26</b>
InceptionResNetv2	<b>0.73</b>	<b>0.77</b>	<b>0.72</b>	<b>0.73</b>	<b>0.74</b>	<b>0.27</b>
THJ	<b>0.75</b>	<b>0.77</b>	<b>0.76</b>	<b>0.71</b>	<b>0.77</b>	<b>0.25</b>

TABLE 4.5 – Tableau des résultats de l'évaluation test moyen pour tout les modeles

#### Discussions

- Dans le tableau ci-dessus, nous remarquons que le modèle THJ a les meilleurs résultats pour les métriques Accuracy ,Précision , MAE et aussi F-Measure.
- Le modèle de théorie des jeux n'a pas obtenu les meilleurs résultats pour les métriques de spécificité et de rappel, car dans certains cas, lorsque 3 modèles ou plus prennent la même décision, cela peut être faux, donc le modèle de théorie des jeux prend une mauvaise décision.

Il y a une autre méthode pour comparer les modèles en utilisant la courbe ROC et la courbe Précision-Rappel, et l'AUC (surface sous la courbe) des deux .

#### 4.5.3.1 Courbes ROC des résultats des modèles

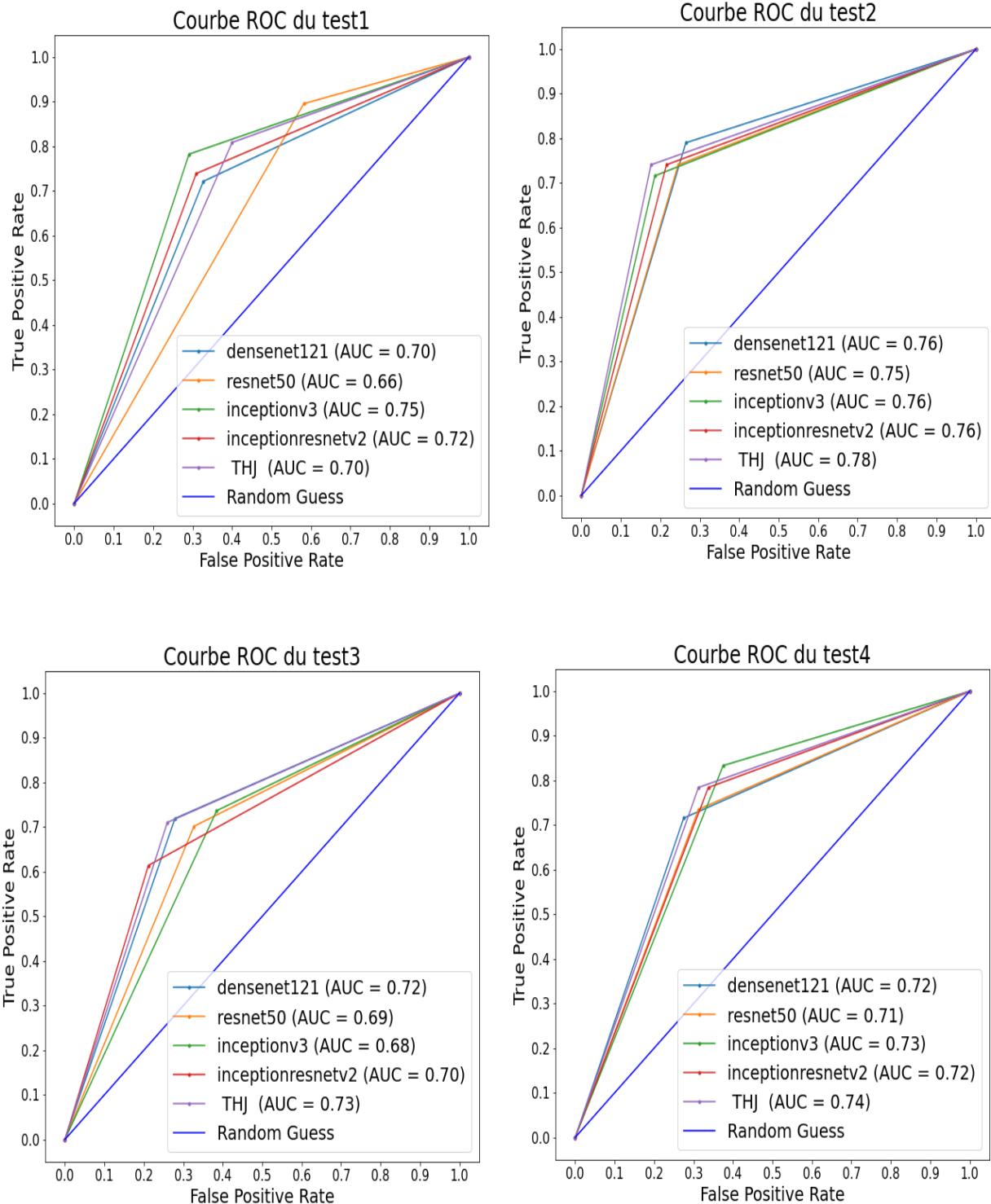


FIGURE 4.33 – Les courbes ROC de tous les modèles dans les ensembles de test différents

#### Discussions

- À partir de ces courbes ROC, nous constatons que le modèle THJ a obtenu les meilleurs résultats pour les tests 2, 3 et 4 avec des surfaces AU-ROC de 0,78, 0,73 et 0,74 respectivement.

### 4.5.3.2 Courbes Précision-Rappel des résultats des modèles

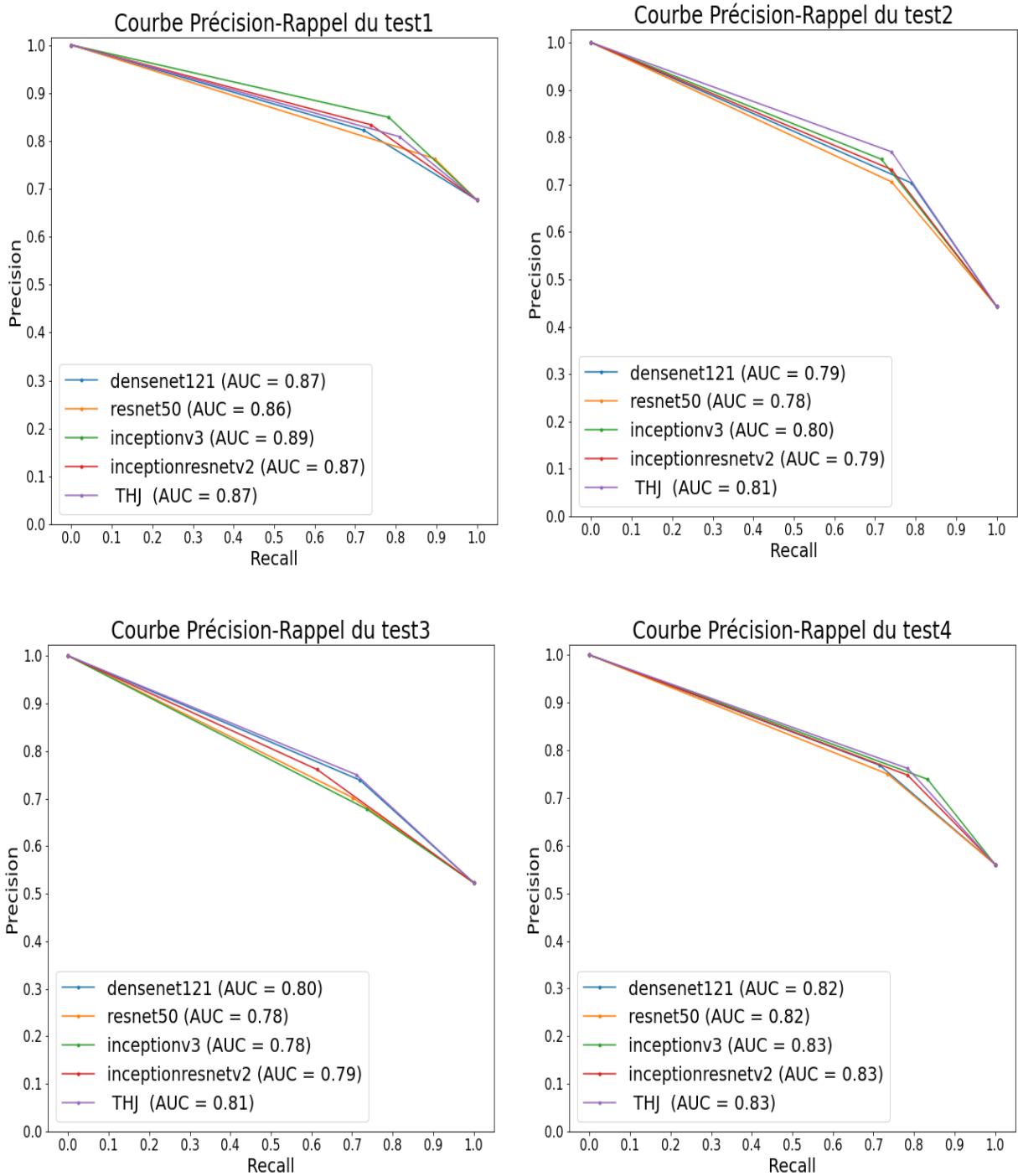


FIGURE 4.34 – les courbes Précision-Rappel de tous les modèles dans les ensembles de test différents

#### Discussions

- À partir de ces courbes Précision-Rappel, nous constatons que le modèle THJ a obtenu les meilleurs résultats pour les tests 2, 3 et 4 avec des surfaces AU-Précision-Rappel de 0,81, 0,81 et 0,83 respectivement.

#### 4.5.3.3 Comparaison des AUC-ROC et AUC-Précision-Rappel entre les modèles

Le tableau suivant regroupe l'AUC des courbes ROC et Précision-Rappel de tous les modèles dans tous les tests :

	Mesure	test1	test2	test3	test4	test-moy
DenseNet121	AU-ROC	0.7	0.76	0.72	0.72	0.73
	AU-Précision-Rappel	0.87	0.79	0.8	0.82	0.82
ResNet50	AU-ROC	0.66	0.75	0.69	0.71	0.7
	AU-Précision-Rappel	0.87	0.78	0.78	0.82	0.81
InceptionV3	AU-ROC	0.75	0.77	0.68	0.73	0.73
	AU-Précision-Rappel	0.89	0.8	0.78	0.83	0.83
InceptionResNetv2	AU-ROC	0.72	0.76	0.7	0.72	0.73
	AU-Précision-Rappel	0.87	0.79	0.79	0.83	0.82
THJ	AU-ROC	0.7	0.78	0.73	0.74	0.74
	AU-Précision-Rappel	0.87	0.81	0.81	0.83	0.83

TABLE 4.6 – Tableau récapitulatif des 5 modèles avec leurs AUC et AU-Précision-Rappel

#### Discussions

- Comme nous le remarquons dans le tableau, le modèle de la théorie des jeux a obtenu les meilleurs AU-ROC et AU-Precision-Recall pour presque tous les tests.

De tous les résultats précédents, nous pouvons conclure que le modèle de la théorie des jeux a amélioré le résultat de la classification.

#### 4.5.3.4 Des exemples de la classification

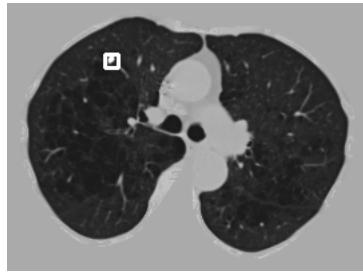
Exemple 1 :



Ground Truth	DenseNet	ResNet	InceptionV3	InceptionResNetV2
0	1.0	1.0	0.999	0.905
	0.0	0.0	0.001	0.095

- La théorie du jeu n'est pas appliquée car il n'y a donc pas de conflit et la classification finale est Bénin avec un degré 97.6%.

Exemple 2 :

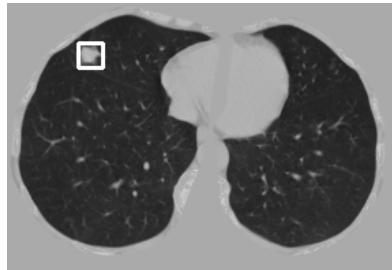


Ground Truth	DenseNet	ResNet	InceptionV3	InceptionResNetV2
1	0.002	0.0	0.986	0.74
	0.998	1.0	0.014	0.26

		Joueur bénin	
		InceptionResN-etcv2	InceptionV3
Joueur malin	DenseNet121	( 0.26 , -0.26 )	( 0.01 , -0.01 )
	ResNet50	( 0.26 , -0.26 )	( 0.01 , -0.01 )

- La valeur finale du jeu est de 0.01 et La classification finale est Malin avec un degré 99.9%.

Exemple 3 :

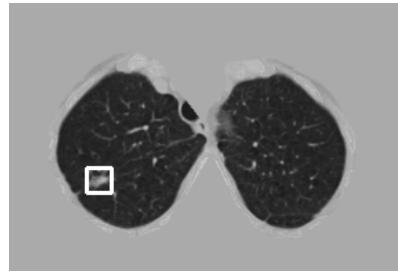


Ground Truth	DenseNet	ResNet	InceptionV3	InceptionResNetV2
0	0.996	1.0	0.02	0.348
	0.004	0.0	0.98	0.652

		Joueur bénin	
		DenseNet121	ResNet50
Joueur malin	InceptionResNetv2	( -0.34 , 0.34 )	( -0.35 , 0.35 )
	InceptionV3	( -0.02 , 0.02 )	( -0.02 , 0.02 )

- La valeur finale du jeu est de -0.02 et La classification finale est Bénin avec un degré 99.8%

Exemple 4 :



Ground Truth	DenseNet	ResNet	InceptionV3	InceptionResNetV2
0	0.001	1.0	1.0	0.398
	0.999	0.0	0.0	0.602

		Joueur bénin	
		InceptionV3	ResNet50
Joueur malin	DenseNet121	( 0.0 , -0.0 )	( 0.0 , 0.0 )
	InceptionResNetv2	( -0.4 , 0.4 )	( -0.4 , 0.4 )

- La valeur finale du jeu est de 0.0 et La classification finale est Bénin avec un degré 100%.

## 4.6 Conclusion

Tout au long de ce chapitre, nous avons défini l'environnement de travail dans lequel nous avons conçu notre application web ainsi que les différentes fonctionnalités de cette dernière à travers des captures d'écran.

Par la suite, nous avons cité quelques critères pour mesurer les résultats de la segmentation et de classification de notre méthode tels que le rappel, la précision ,Dice Coefficient , les courbes ROC et les courbes Précision-Rappel . Après l'évaluation de notre système nous avons comparé nos résultats des modèles 2d et 3d pour la segmentation et pour la classification nous avons comparé les résultats des 4 modèles d'apprentissage profond et du modèle de théorie des jeux avec les différents tests.

Ce qui nous a permis de conclure que le modèle de la théorie des jeux a amélioré les résultats de la classification et que notre système de détection a atteint une très bonne performance dans la détection et la classification des nodules pulmonaires.

# Conclusion générale

L'objectif principal de notre travail est de réaliser un système de détection et de classification des nodules pulmonaires à partir d'un scanner 3D. Au cours de ce mémoire, nous avons étudié et implémenté un nombre de méthodes pour aboutir à cet objectif.

Tout d'abord, nous avons collecté les données et compris leur nature et comment nous pouvons les exploiter. Ensuite, nous procédons au prétraitement des données afin de conserver que les détails importants en utilisant les différentes techniques de traitement des images telles que le filtrage, les transformations, etc.

Ensuite, nous avons préparé les données pour qu'elles servent d'entrée à notre modèle de segmentation 3D U-Net afin de détecter et de segmenter uniquement les nodules. Ce modèle a été entraîné plusieurs fois jusqu'à ce qu'il atteigne le meilleur résultat possible après avoir ajusté de nombreux paramètres et techniques de régularisation. Après cela, nous avons utilisé le résultat de la dernière étape pour servir d'entrée aux modèles de classification : ResNet50 , Inception V3 , DensNet121 , InceptionResnt V2. Nous avons entraîné chaque modèle individuellement, puis nous avons introduit la théorie des jeux afin d'aider à prendre des décisions en utilisant la sortie de chaque modèle dans un jeu à somme nulle.

Enfin, nous avons implémenté une interface graphique qui contient toutes les fonctionnalités suivantes avec des visualisations qui aident l'utilisateur à interagir avec et à comprendre les résultats souhaités. En général, notre système a donné des résultats intéressants (mentionnés dans le chapitre 4) et avons globalement atteint les objectifs fixés malgré les limitations du matériels et la disponibilité des données (Très peu de données), car la plupart des modèles d'apprentissage profond et des architectures nécessitent beaucoup de données et utilisent beaucoup de ressources, en particulier celles liées au domaine de la 3D. Le système peut être amélioré sur plusieurs aspects, nous suggérons de :

- Améliorer la précision de la détection en augmentant le taux des données d'entrée.  
La façon de le faire est de collaborer avec les cliniques locales pour nous fournir des ensembles de données précises si cela est possible.
- La possibilité de donner plus de détails sur le cancer détecté comme le type.

# Bibliographie

- [1] Chafik KERMAD. « Segmentation d'image : recherche d'une mise en oeuvre automatique par coopération de méthodes ». Président : A. Chehikian Professeur INPG ENSERG - Grenoble Rapporteurs : J.-M. Chassery Directeur de Recherche CNRS-IMAG S. Philipp Professeur ENSEA - Cergy Pontoise Examinateurs : P. Bouthemy Directeur de Recherche IRISA-INRIA K. Chehdi Professeur ENSSAT - Lannion R. Collorec Professeur Université de Rennes I. Theses. Université Rennes 1, juill. 1997. URL : <https://tel.archives-ouvertes.fr/tel-00008781>.
- [2] J.S. DUNCAN et N. AYACHE. « Medical image analysis : progress over two decades and the challenges ahead ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.1 (2000), p. 85-106. DOI : 10.1109/34.824822.
- [3] B. GUERRIN. In : "La théorie des jeux", *Economica* (2002).
- [4] S. NATKIN. In : *Utilisation de la théorie des jeux dans la conception de jeux vidéo* (2002).
- [5] Sun-Chong WANG. « Artificial Neural Network ». In : *Interdisciplinary Computing in Java Programming*. Boston, MA : Springer US, 2003, p. 81-100. ISBN : 978-1-4615-0377-4. DOI : 10.1007/978-1-4615-0377-4\_5. URL : [https://doi.org/10.1007/978-1-4615-0377-4\\_5](https://doi.org/10.1007/978-1-4615-0377-4_5).
- [6] Francesco BUCCOLIERI, Cosimo DISTANTE et Alessandro LEONE. « Human posture recognition using active contours and radial basis function neural network ». In : t. 2005. Jan. 2005, p. 213-218. DOI : 10.1109/AVSS.2005.1577269.
- [7] Ronald A. CASTELLINO. « Computer Aided Detection (CAD) : An overview ». In : *Cancer Imaging* 5.1 (2005), p. 17-19. DOI : 10.1102/1470-7330.2005.0018.
- [8] T. PÉNARD. In : "Introduction à la théorie des jeux" Faculté de Sciences Économiques Université de Rennes 1 (2007).
- [9] Noam NISAN et al. *Algorithmic game theory*. Cambridge University Press, 2008.
- [10] Adam O HEBB et Andrew V POLIAKOV. « Imaging of deep brain stimulation leads using extended Hounsfield unit CT ». en. In : *Stereotact Funct Neurosurg* 87.3 (mars 2009), p. 155-160.
- [11] Maurizio INFANTE et al. « A randomized study of lung cancer screening with Spiral Computed Tomography ». In : *American Journal of Respiratory and Critical Care Medicine* 180.5 (2009), p. 445-453. DOI : 10.1164/rccm.200901-0076oc.

- [12] Pang-Ning TAN. « Receiver Operating Characteristic ». In : *Encyclopedia of Database Systems*. Sous la dir. de LING LIU et M. TAMER ÖZSU. Boston, MA : Springer US, 2009, p. 2349-2352. ISBN : 978-0-387-39940-9. DOI : 10.1007/978-0-387-39940-9\_569. URL : [https://doi.org/10.1007/978-0-387-39940-9\\_569](https://doi.org/10.1007/978-0-387-39940-9_569).
- [13] Maïtine BERGOUNIOUX. « Quelques méthodes de filtrage en Traitement d'Image ». Cours donné dans le cadre d'une école CIMPA - en attente de publication dans les actes. Août 2010. URL : <https://hal.archives-ouvertes.fr/hal-00512280>.
- [14] Samuel G. ARMATO et al. « The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI) : A completed reference database of lung nodules on CT scans ». In : *Medical Physics* 38.2 (2011), p. 915-931. DOI : 10.1118/1.3528204.
- [15] Johannes FÜRNKRANZ et al. « Mean absolute error ». In : *Encyclopedia of Machine Learning* (2011), p. 652-652. DOI : 10.1007/978-0-387-30164-8\_525.
- [16] Muthukrishnan R. « Edge Detection Techniques For Image Segmentation ». In : *International journal of computer science and information technology* 3 (déc. 2011), p. 259-267. DOI : 10.5121/ijcsit.
- [17] Satinder P SINGH et al. « Reader variability in identifying pulmonary nodules on chest radiographs from the national lung screening trial ». en. In : *J. Thorac. Imaging* 27.4 (juill. 2012), p. 249-254.
- [18] FANYANG2012. *Edge detection and Sobel operator*. Mars 2013. URL : <https://gtms1337.wordpress.com/2013/03/16/edge-detection-and-sobel-operator/>.
- [19] Phan NGUYEN et al. « CELL BLOCK SAMPLES FROM ENDOBRONCHIAL ULTRASOUND TRANSBRONCHIAL NEEDLE ASPIRATION (EBUS-TBNA) PROVIDE SUFFICIENT MATERIAL FOR ANCILLARY TESTING IN LUNG CANCER ». In : *Respirology* 18 (avr. 2013), p. 31-31. DOI : 10.21037/jtd.2016.08.74.
- [20] Hoo-Chang SHIN et al. « Stacked Autoencoders for Unsupervised Feature Learning and Multiple Organ Detection in a Pilot Study Using 4D Patient Data ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), p. 1930-1943. DOI : 10.1109/TPAMI.2012.277.
- [21] Brett W. CARTER et al. *Small cell lung carcinoma : Staging, imaging, and treatment considerations*. Oct. 2014. URL : <https://pubs.rsna.org/doi/10.1148/rg.346140178>.
- [22] Christian SZEGEDY et al. *Going Deeper with Convolutions*. 2014. DOI : 10.48550/ARXIV.1409.4842. URL : <https://arxiv.org/abs/1409.4842>.
- [23] Daniel BELL et Kyle GREENWAY. « Hounsfield unit ». In : *Radiopaedia.org* (2015). DOI : 10.53347/rid-38181.
- [24] Omar BOUDRAA. « Segmentation d'Images par Coopération Régions-Contours en utilisant la Théorie des jeux ». Thèse de doct. Avr. 2015. DOI : 10.13140/RG.2.2.21450.62402.
- [25] Kaiming HE et al. *Deep Residual Learning for Image Recognition*. 2015. DOI : 10.48550/ARXIV.1512.03385. URL : <https://arxiv.org/abs/1512.03385>.
- [26] M.BERGOUNIOUX. *Introduction Au Traitement De L'image*. 2015. URL : <http://maitine.disque.math.cnrs.fr/RTRcoursBergounioux.pdf>.

- [27] Keiron O'SHEA et Ryan NASH. *An Introduction to Convolutional Neural Networks*. 2015. DOI : 10.48550/ARXIV.1511.08458. URL : <https://arxiv.org/abs/1511.08458>.
- [28] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX. *U-Net : Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI : 10.48550/ARXIV.1505.04597. URL : <https://arxiv.org/abs/1505.04597>.
- [29] Christian SZEGEDY et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. DOI : 10.48550/ARXIV.1512.00567. URL : <https://arxiv.org/abs/1512.00567>.
- [30] Renaud BOURLÈS et Dominique HENRIET. *Theorie des jeux - théorie des Jeux Renaud Bourlès Dominique Henriet EAO-32-O-FIST 2ème année*. 2016. URL : <https://www.studocu.com/en-gb/document/london-business-school/management/theorie-des-jeux/3537739>.
- [31] Varun GULSHAN et al. « Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs ». In : *JAMA* 316.22 (déc. 2016), p. 2402-2410. ISSN : 0098-7484. DOI : 10.1001/jama.2016.17216. eprint : <https://jamanetwork.com/journals/jama/articlepdf/2588763/joi160132.pdf>. URL : <https://doi.org/10.1001/jama.2016.17216>.
- [32] Gao HUANG et al. *Densely Connected Convolutional Networks*. 2016. DOI : 10.48550/ARXIV.1608.06993. URL : <https://arxiv.org/abs/1608.06993>.
- [33] A. MAROUANI et al. « Le cancer broncho-pulmonaire primitif : à propos de 780 cas ». In : *Revue des Maladies Respiratoires* 33 (2016). 20e Congrès CPLF, A99-A100. ISSN : 0761-8425. DOI : <https://doi.org/10.1016/j.rmr.2015.10.147>. URL : <https://www.sciencedirect.com/science/article/pii/S0761842515005124>.
- [34] Christian SZEGEDY et al. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016. DOI : 10.48550/ARXIV.1602.07261. URL : <https://arxiv.org/abs/1602.07261>.
- [35] Mohamed BENDAOUD. *These Bendaoud Mohamed Habib Partie 3*. 2017. URL : <https://www.scribd.com/document/540531678/These-Bendaoud-Mohamed-Habib-Partie-3>.
- [36] Andre ESTEVA et al. « Dermatologist-level classification of skin cancer with Deep Neural Networks ». In : *Nature* 542.7639 (2017), p. 115-118. DOI : 10.1038/nature21056.
- [37] *Explorations thoraciques principales*. Sept. 2017. URL : <https://www.chirvtt.fr/chirurgie-thorax/explorations-thoraciques-principales/>.
- [38] Guixia KANG et al. « 3D multi-view convolutional neural networks for lung nodule classification ». In : *PLOS ONE* 12 (nov. 2017), e0188290. DOI : 10.1371/journal.pone.0188290.
- [39] Geert LITJENS et al. « A survey on deep learning in medical image analysis ». In : *Medical Image Analysis* 42 (déc. 2017), p. 60-88. DOI : 10.1016/j.media.2017.07.005. URL : <https://doi.org/10.1016%5C2Fj.media.2017.07.005>.
- [40] Heber MACMAHON et al. « Guidelines for management of incidental pulmonary nodules detected on CT images : From the fleischner society 2017 ». In : *Radiology* 284.1 (2017), p. 228-243. DOI : 10.1148/radiol.2017161659.

- [41] Anh H. REYNOLDS. *Convolutional Neural Networks (cnns)*. Oct. 2017. URL : <https://anhreynolds.com/blogs/cnn.html>.
- [42] Arnaud Arindra SETIO et al. « Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images : The luna16 challenge ». In : *Medical Image Analysis* 42 (2017), p. 1-13. DOI : 10.1016/j.media.2017.06.015.
- [43] David S. BATISTA. Août 2018. URL : [https://www.davidsbatista.net/blog/2018/08/19/NLP\\_Metrics/](https://www.davidsbatista.net/blog/2018/08/19/NLP_Metrics/).
- [44] Daniel BERRAR. « Cross-Validation ». In : jan. 2018. ISBN : 9780128096338. DOI : 10.1016/B978-0-12-809633-8.20349-X.
- [45] Jeremy JORDAN. *Evaluating image segmentation models*. Déc. 2018. URL : <https://www.jeremyjordan.me/evaluating-image-segmentation-models/>.
- [46] Didier MÜLLER. *Chapitre 4 traitement d'images - apprendre en Ligne*. 2018. URL : <https://www.apprendre-en-ligne.net/info/images/images.pdf>.
- [47] Vishnu SUBRAMANIAN. *Deep learning with pytorch : A practical approach to building neural network models using pytorch*. Packt Publishing, 2018.
- [48] Wentao ZHU et al. *DeepEM : Deep 3D ConvNets With EM For Weakly Supervised Pulmonary Nodule Detection*. 2018. DOI : 10.48550/ARXIV.1805.05373. URL : <https://arxiv.org/abs/1805.05373>.
- [49] Wentao ZHU et al. *DeepLung : Deep 3D Dual Path Nets for Automated Pulmonary Nodule Detection and Classification*. 2018. DOI : 10.48550/ARXIV.1801.09555. URL : <https://arxiv.org/abs/1801.09555>.
- [50] Luc COLLARD et Corinne FANTONI. « COURS DE THÉORIE DES JEUX ». In : *COURS DE THÉORIE DES JEUX, UFR STAPS Paris Descartes* (2019).
- [51] Erika D. *Accuracy, Recall & Precision*. Déc. 2019. URL : <https://medium.com/@erika.dauria/accuracy-recall-precision-80a5b6cbd28d>.
- [52] D.DAHMANI. In : *Théorie des jeux cours Master 1 Informatique visuelle « MIV » ,Université de Science et Technologie Houari Boumediene* (2019).
- [53] DEEPAI. *Evaluation metrics*. Mai 2019. URL : <https://deepai.org/machine-learning-glossary-and-terms/evaluation-metrics>.
- [54] *Explain about gaussian filtering ?* Oct. 2019. URL : <https://www.i2tutorials.com/explain-about-gaussian-filtering/>.
- [55] Richa GROVER. *Deep Learning - Overview, practical examples, popular algorithms*. Sept. 2019. URL : <https://www.analyticssteps.com/blogs/deep-learning-overview-practical-examples-popular-algorithms>.
- [56] Yuemeng LI et Yong FAN. *DeepSEED : 3D Squeeze-and-Excitation Encoder-Decoder Convolutional Neural Networks for Pulmonary Nodule Detection*. 2019. DOI : 10.48550/ARXIV.1904.03501. URL : <https://arxiv.org/abs/1904.03501>.
- [57] Mr.KARIM MEZZOUG. *TRAITEMENT ET ANALYSE DES IMAGES NUMÉRIQUES*. 2019. URL : [http://fmi.univ-tiaret.dz/images/1GI/2020/S2/TAI/TAI\\_COURS\\_GI\\_2019\\_2020.pdf](http://fmi.univ-tiaret.dz/images/1GI/2020/S2/TAI/TAI_COURS_GI_2019_2020.pdf).
- [58] Alex MITRANI. *Evaluating categorical models II : Sensitivity and specificity*. Déc. 2019. URL : <https://towardsdatascience.com/evaluating-categorical-models-ii-sensitivity-and-specificity-e181e573cff8>.

- [59] Miao ZHANG et al. *Multi-level CNN for lung nodule classification with Gaussian Process assisted hyperparameter optimization*. 2019. DOI : 10.48550/ARXIV.1901.00276. URL : <https://arxiv.org/abs/1901.00276>.
- [60] Zongwei ZHOU et al. *Models Genesis : Generic Autodidactic Models for 3D Medical Image Analysis*. 2019. DOI : 10.48550/ARXIV.1908.06912. URL : <https://arxiv.org/abs/1908.06912>.
- [61] Sai BALAJI. *Binary image classifier CNN using tensorflow*. Août 2020. URL : <https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697>.
- [62] Djamila DAHMANI, Mehdi CHEREF et Slimane LARABI. « Zero-sum game theory model for segmenting skin regions ». In : *Image and Vision Computing* 99 (2020), p. 103925. ISSN : 0262-8856. DOI : <https://doi.org/10.1016/j.imavis.2020.103925>. URL : <https://www.sciencedirect.com/science/article/pii/S0262885620300573>.
- [63] Xuehai HE et al. *Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans*. Avr. 2020. DOI : 10.1101/2020.04.13.20063941.
- [64] Sourav KUMAR. *Data augmentation increases accuracy of your model but how ?* Août 2020. URL : <https://medium.com/secure-and-private-ai-writing-challenge/data-augmentation-increases-accuracy-of-your-model-but-how-aa1913468722>.
- [65] Siddharth SHARMA, Simone SHARMA et Anidhya ATHAIYA. « ACTIVATION FUNCTIONS IN NEURAL NETWORKS ». In : *International Journal of Engineering Applied Sciences and Technology* 04 (mai 2020), p. 310-316. DOI : 10.33564/IJEAST.2020.v04i12.054.
- [66] Doug STEEN. *Precision-recall curves*. Sept. 2020. URL : <https://medium.com/@douglassteen/precision-recall-curves-d32e5b290248>.
- [67] STUDIOUSGUY. *5 examples of game theory in Real life*. Mars 2020. URL : [https://studiousguy.com/game-theory-examples/#4\\_Product\\_Pricing\\_Decisions](https://studiousguy.com/game-theory-examples/#4_Product_Pricing_Decisions).
- [68] Ekin TIU. *Metrics to evaluate your semantic segmentation model*. Oct. 2020. URL : <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>.
- [69] Zhitao XIAO et al. « Segmentation of Lung Nodules Using Improved 3D-UNet Neural Network ». In : *Symmetry* 12.11 (2020). ISSN : 2073-8994. DOI : 10.3390/sym12111787. URL : <https://www.mdpi.com/2073-8994/12/11/1787>.
- [70] Laith ALZUBAIDI et al. « Review of deep learning : concepts, CNN architectures, challenges, applications, future directions ». In : *Journal of Big Data* 8.1 (mars 2021), p. 53. ISSN : 2196-1115. DOI : 10.1186/s40537-021-00444-8. URL : <https://doi.org/10.1186/s40537-021-00444-8>.
- [71] MICROSOFT. *Documentation for visual studio code*. Nov. 2021. URL : <https://code.visualstudio.com/docs>.
- [72] N.BAHA. *Cours de traitement et Analyse d'images et de la vidéo MIV 2 ,USTHB*. 2021.
- [73] Sarang NARKHEDE. *Understanding AUC - roc curve*. Juin 2021. URL : <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

- [74] *OpenCV - overview*. Août 2021. URL : <https://www.geeksforgeeks.org/opencv-overview/>.
- [75] SIMPLILEARN. *What is Keras and why it so popular in 2021 : Simplilearn*. Sept. 2021. URL : <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras>.
- [76] Mrinal TYAGI. *Image segmentation : Part 2*. Juill. 2021. URL : <https://towardsdatascience.com/image-segmentation-part-2-8959b609d268>.
- [77] Rahul AGARWAL. *The 5 classification evaluation metrics every data scientist must know*. Mars 2022. URL : <https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226>.
- [78] *CNN : Introduction to pooling layer*. Août 2022. URL : <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>.
- [79] M.D. EDWARD T. CREAGAN. *What to know if you have lung nodules*. Juill. 2022. URL : <https://www.mayoclinic.org/diseases-conditions/lung-cancer/expert-answers/lung-nodules/faq-20058445#:~:text=Lung%5C%20nodules%5C%20are%5C%20small%5C%20clumps,usually%5C%20don't%5C%20cause%5C%20symptoms..>
- [80] HUDASALEH1. *CNN using Keras*. Avr. 2022. URL : <https://www.kaggle.com/code/hudasaleh1/cnn-using-keras>.
- [81] Purva HUILGOL. *Precision vs recall : Precision and recall machine learning*. Juin 2022. URL : [https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/#h2\\_10](https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/#h2_10).
- [82] *Jaccard index*. Août 2022. URL : [https://en.wikipedia.org/wiki/Jaccard\\_index](https://en.wikipedia.org/wiki/Jaccard_index).
- [83] Wenfa JIANG et al. « Application of Deep Learning in Lung Cancer Imaging Diagnosis ». In : *Journal of Healthcare Engineering* 2022 (jan. 2022), p. 6107940. ISSN : 2040-2295. DOI : 10.1155/2022/6107940. URL : <https://doi.org/10.1155/2022/6107940>.
- [84] Chou-Chin LAN et al. « Deep Learning-based Artificial Intelligence Improves Accuracy of Error-prone Lung Nodules ». In : *Int J Med Sci* 19 (2022), p. 490-498. DOI : 10.7150/ijms.69400. URL : <https://www.medsci.org/v19p0490.htm>.
- [85] Head of AI Growth @Datatron LANCE WILLS. *Understanding the confusion matrix for Model Evaluation & Monitoring*. Juill. 2022. URL : <https://datatron.com/understanding-the-confusion-matrix-for-model-evaluation-monitoring/>.
- [86] SIMPLILEARN. *An introduction to deep learning [2022 updated edition]*. Juill. 2022. URL : <https://www.simplilearn.com/tutorials/deep-learning-tutorial/introduction-to-deep-learning>.
- [87] Roman SOLOVYEV, Alexandr A KALININ et Tatiana GABRUSEVA. « 3D convolutional neural networks for stalled brain capillary detection ». In : *Computers in Biology and Medicine* 141 (2022), p. 105089. DOI : 10.1016/j.compbiomed.2021.105089.
- [88] James WILSON. *What is python ? - definition from Techopedia*. Juill. 2022. URL : <https://www.techopedia.com/definition/3533/python>.
- [89] El Khader A. *Cours de théorie des jeux : Jeux coopératifs et non-coopératifs*. URL : <http://gestion.coursgratuits.net/theorie-des-jeux/jeux-cooperatifs-et-non-cooperatifs.php>.

- [90] Robert J. AUMANN et Sergiu HART. *Handbook of Game Theory (1994) : With economic applications volumen 2*. Elsevier Science Ltda.
- [91] *Basics of Image Processing*. URL : <https://vincmazet.github.io/bip/digital-images/operations.html#f-intro-subtraction>.
- [92] I. BLOCH. *Filtrage morphologique*. URL : <https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/geomatique-42641210/morphologie-mathematique-et-traitement-d-images-af1515/filtrage-morphologique-af1515niv10005.html>.
- [93] *Cancer du poumon - symptômes, causes, traitements et prévention*. URL : <https://www.vidal.fr/maladies/cancers/cancer-poumon.html>.
- [94] Plateforme Logicielle CÉPIA. *Plateforme Logicielle Cépia*. URL : <https://www.pfl-cepia.inra.fr/index.php?page=tutoImg-watershed>.
- [95] *CIE XYZ*. URL : <https://boowiki.info/art/espace-couleur/cie-xyz.html>.
- [96] *Computed Tomography*. URL : <https://us.medical.canon/products/computed-tomography/>.
- [97] *CT lung cancer screening - Los Angeles, CA*. URL : <https://www.cedars-sinai.org/programs/imaging-center/exams/ct-scans/lung.html>.
- [98] *Data science bowl 2017*. URL : <https://www.kaggle.com/c/data-science-bowl-2017>.
- [99] Center for DEVICES et Radiological HEALTH. *Medical Imaging*. URL : <https://www.fda.gov/radiation-emitting-products/radiation-emitting-products-and-procedures/medical-imaging>.
- [100] *Getting started with pydicom*. URL : [https://pydicom.github.io/pydicom/0.9/getting\\_started.html#:~:text=pydicom%5C%20is%5C%20a%5C%20pure%5C%20python,again%5C%20to%5C%20DICOM%5C%20format%5C%20files..](https://pydicom.github.io/pydicom/0.9/getting_started.html#:~:text=pydicom%5C%20is%5C%20a%5C%20pure%5C%20python,again%5C%20to%5C%20DICOM%5C%20format%5C%20files..)
- [101] Bernard GUERRIEN. *Théorie des Jeux*. URL : <https://www.universalis.fr/encyclopedie/theorie-des-jeux/>.
- [102] Matthew HOFFMAN. *Lungs (human anatomy) : Picture, function, definition, conditions*. URL : <http://www.webmd.com/lung/picture-of-the-lungs>.
- [103] Sid LEE. *Les Poumons*. URL : <https://cancer.ca/fr/cancer-information/cancer-types/lung/what-is-lung-cancer/the-lungs>.
- [104] *Luna16 - Grand Challenge*. URL : <https://luna16.grand-challenge.org/>.
- [105] *Lung cancer : Symptoms, signs, stages, and more*. URL : <https://www.medicalnewstoday.com/articles/323701>.
- [106] *Mean filter*. URL : <https://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm>.
- [107] *Nash equilibrium*. URL : <https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100223327>.
- [108] *NCI Dictionary of Cancer terms*. URL : <https://www.cancer.gov/publications/dictionaries/cancer-terms/def/sputum-cytology>.
- [109] *Non-linear filtering of images (image processing) part 1*. URL : <http://what-when-how.com/embedded-image-processing-on-the-tms320c6000-dsp/non-linear-filtering-of-images-image-processing-part-1/>.

- [110] OEPNCV. *Canny edge detection*. URL : [https://docs.opencv.org/4.x/da/d22/tutorial\\_py\\_canny.html](https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html).
- [111] Xavier PHILIPPEAU. *Traitement d'images - les filtres usuels*. URL : [https://xphilipp.developpez.com/articles/filtres/?page=page\\_3](https://xphilipp.developpez.com/articles/filtres/?page=page_3).
- [112] *Pixel multiplication and scaling*. URL : <https://homepages.inf.ed.ac.uk/rbf/HIPR2/pixmult.htm>.
- [113] *PyQt tutorial*. URL : <https://www.tutorialspoint.com/pyqt/index.htm>.
- [114] *Scikit learn - introduction*. URL : [https://www.tutorialspoint.com/scikit\\_learn/scikit\\_learn\\_introduction.htm](https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm).
- [115] *Shannon sampling theorem*. URL : <https://www.sciencedirect.com/topics/computer-science/shannon-sampling-theorem>.
- [116] *SIMPLEITK*. URL : <https://pypi.org/project/SimpleITK/>.
- [117] *Teaching*. URL : <https://vknight.org/>.
- [118] *Tensorflow*. URL : <https://www.tensorflow.org/>.
- [119] *Traitement d'histogramme*. URL : <https://perso.esiee.fr/~perretb/I5FM/TAI/histogramme/index.html>.
- [120] *Traitement d'image*. URL : [https://www.lama.univ-savoie.fr/mediawiki/index.php/Traitement\\_d%27image](https://www.lama.univ-savoie.fr/mediawiki/index.php/Traitement_d%27image).