



---

## Master II

Ingénierie Mathématique pour la Science de Données

Parcours Modélisation, Calcul et Aide à la Décision (MCAD)

### Rapport de stage

Rédigé par :

Vanga Gustave Hermann MOULO

---

## Développement d'un modèle de Deep Learning pour la prédiction de la récurrence des gliomes

---

Encadrant laboratoire :  
Timothée ZARAGORI

Encadrant universitaire :  
Laurent BOUGRAIN

18 septembre 2024





### Attestation sur l'honneur contre le plagiat

(à joindre obligatoirement à tout travail de recherche ou dossier remis à un enseignant)

Je soussigné(e),

Nom, Prénom, MOULO Vanga Gustave Hermann

Régulièrement inscrit à l'Université de Lorraine

N° de carte d'étudiant : 32219982

Année universitaire : 2023-2024

Niveau d'études : M

Parcours : Ingénierie Mathématiques pour la Science de Données (IMSD)

Certifie que mon rapport de stage de fin de M2 est un travail original et que toutes les sources utilisées ont été indiquées dans leur totalité (y compris les outils d'IA pré-entraînés génératifs). Je certifie, de surcroît, que je n'ai ni recopié ni utilisé des idées ou des formulations tirées d'un ouvrage, article ou mémoire, en version imprimée ou électronique, sans mentionner précisément leur origine et que les citations intégrales (verbatim) sont signalées entre guillemets.

Conformément à la loi, le non-respect de ces dispositions me rend passible de poursuites devant la commission disciplinaire et les tribunaux de la République Française.

Fait à Nancy, le 27/08/2024

Signature

---

# Remerciements

Tout d'abord, je souhaite adresser mes remerciements à Madame FOUGERE Marjorie, pour m'avoir offert l'opportunité d'intégrer le laboratoire Nancyclotep.

Ma profonde reconnaissance va à l'endroit de mon encadrant de stage, Monsieur Timothée ZARAGORI, pour sa bienveillance, son accompagnement effectif, ses conseils constructifs et son engagement total dans le déroulement de mon stage.

Je remercie également toute l'équipe du laboratoire Nancyclotep et IADI pour leur accueil chaleureux.

Enfin, Je tiens à remercier chaleureusement l'ensemble de l'équipe pédagogique du Master 2 d'Ingénierie Mathématiques pour la Science des Données pour leur soutien constant et la qualité des enseignements reçus. Un remerciement particulier à mon tuteur universitaire, Monsieur Laurent BOUGRAIN, pour son suivi.

---

# Résumé

Cette étude s'intéresse aux gliomes, tumeurs cérébrales malignes les plus courantes et particulièrement agressives. Ces tumeurs posent un défi majeur en termes de surveillance post-traitement en raison des limitations inhérentes à l'IRM, laquelle ne permet pas toujours de distinguer de manière fiable entre une progression tumorale réelle et les effets secondaires des traitements, tels que la pseudoprogression et la radionécrose.

Pour répondre à ces défis, l'imagerie TEP aux acides aminés, notamment avec le radiotracer  $^{18}\text{F}$ -FDOPA, a été identifiée comme une méthode prometteuse en raison de sa spécificité supérieure dans la détection des tissus tumoraux actifs. Cette modalité est particulièrement utile pour différencier les récurrences tumorales des effets induits par les traitements, une distinction cruciale pour améliorer la prise en charge des patients.

Dans ce contexte, cette investigation a exploré l'application de l'intelligence artificielle à l'analyse des images TEP réalisées pour cette indication, en testant plusieurs modèles de réseaux neuronaux. Parmi les modèles évalués, VGG11 et ResNet ont été mis à l'essai, ainsi qu'un modèle de réseau de neurones convolutifs 3D, que nous avons développé spécifiquement pour cette étude. Ces modèles ont été optimisés à travers diverses techniques de prétraitement d'images, d'augmentation des données et de paramétrage, visant à améliorer la performance prédictive du modèle.

**Mots clés :** Gliomes, TEP  $^{18}\text{F}$ -FDOPA, Réseaux de neurones convolutifs, Apprentissage profond, Intelligence artificielle.

# Abstract

This study focuses on gliomas, the most common and particularly aggressive malignant brain tumors. These tumors pose a major challenge in post-treatment monitoring due to the inherent limitations of MRI, which does not always reliably distinguish between true tumor progression and treatment-related effects, such as pseudoproggression and radionecrosis.

To address these challenges, amino acid PET imaging, particularly with the  $^{18}\text{F}$ -FDOPA radiotracer, has been identified as a promising method due to its superior specificity in detecting active tumor tissues. This modality is especially useful for distinguishing tumor recurrence from treatment-induced effects, a crucial distinction for improving patient care.

In this context, this investigation explored the application of artificial intelligence in the analysis of PET images for this indication, testing several neural network models. Among the models evaluated, VGG11 and ResNet were tested, as well as a 3D convolutional neural network model specifically developed for this study. These models were optimized through various image preprocessing, data augmentation, and parameter tuning techniques aimed at improving the predictive performance of the model.

**Keywords :** Gliomas,  $^{18}\text{F}$ -FDOPA PET, Convolutional neural networks, Deep learning, Artificial intelligence.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Organisme d'accueil . . . . .	6
1.2	Contexte général et objectif du stage . . . . .	7
<b>2</b>	<b>Etat de l'art</b>	<b>9</b>
2.1	Imagerie TEP - Acquisition - Reconstruction d'images . . . . .	9
2.2	Prédiction de la Récidive/Réurrence en TEP à la $^{18}\text{F}$ -FDOPA . . . . .	10
2.3	Concepts fondamentaux du deep learning . . . . .	11
2.3.1	Réseau de Neurones Convolutifs 3D . . . . .	11
2.3.2	VGG (Visual Geometry Group) . . . . .	12
2.3.3	Residual Network (ResNet) . . . . .	13
2.3.4	ACS Convolutions . . . . .	14
2.3.5	SENet (Squeeze-and-Excitation Networks) . . . . .	15
2.3.6	SqueezeNet . . . . .	16
2.4	Optimisation et Paramétrage . . . . .	17
2.4.1	Fonction de perte . . . . .	17
2.4.2	Normalisation . . . . .	19
2.4.3	Dropout . . . . .	20
2.5	Cartes d'Attention 3D avec M3d-CAM . . . . .	21
<b>3</b>	<b>Matériel et Méthodes</b>	<b>22</b>
3.1	Description de la base de données . . . . .	22
3.2	Techniques de prétraitement et d'augmentation d'images . . . . .	22
3.2.1	Prétraitement d'images . . . . .	22
3.2.2	Techniques d'augmentation d'images . . . . .	24
3.3	Modélisation . . . . .	25
3.4	Entraînement et Validation . . . . .	29
3.4.1	Environnement de travail . . . . .	29
3.4.2	Validation Croisée et Équilibrage des Données . . . . .	29
3.4.3	Configuration de l'Entraînement . . . . .	29
3.4.4	Prédiction . . . . .	29
3.5	Plan d'expériences . . . . .	30
3.6	Métriques d'évaluation . . . . .	30
<b>4</b>	<b>Résultats et discussion</b>	<b>32</b>
4.1	Distribution des patients par classe . . . . .	32
4.2	Performances des Modèles . . . . .	32
4.3	Discussion . . . . .	34
<b>5</b>	<b>Conclusion</b>	<b>38</b>

# Table des figures

1.1	Caméra de tomographie par émission monophotonique VERITON-CT utilisée dans le service de médecine nucléaire du CHU de Nancy . . . . .	7
1.2	Flux de travail et principes des algorithmes de radiomique et d'intelligence artificielle extrait de (PENG et al. 2024) . . . . .	8
2.1	Architecture CNN Basique extrait de O'Shea, K., et al. (2015) . . . . .	12
2.2	Block résiduel. Extrait de HE et al. 2016 . . . . .	13
2.3	Comparaison entre les convolutions ACS proposées et les travaux antérieurs sur la modélisation des images médicales 3D. (YANG et al. 2021) . . . . .	14
2.4	A Squeeze-and-Excitation block (HU, SHEN et SUN 2018) . . . . .	16
2.5	Vue microarchitecturale : Organisation des filtres de convolution dans le module Fire Dans cet exemple, $s_{1 \times 1} = 3$ , $e_{1 \times 1} = 4$ , et $e_{3 \times 3} = 4$ .(IANDOLA 2016) . . . . .	17
2.6	Vue Macroarchitecturale de l'Architecture SqueezeNet. À gauche : SqueezeNet ; Au milieu : SqueezeNet avec un chemin de contournement simple ; À droite : SqueezeNet avec un chemin de contournement complexe (IANDOLA 2016) . . . . .	18
3.1	Différentes interpolation basées sur une grille de $4 \times 4$ avec un espacement de (3,00, 3,00) mm. Extrait de (ZWANENBURG et AL. 2016) . . . . .	23
3.2	Quelques images de patients avant et après prétraitement. . . . .	24
3.3	Architecture CNN3D . . . . .	25
3.4	Architecture VGG11 . . . . .	26
3.5	Architecture ResNet10 . . . . .	27
3.6	Architecture ACS-ResNet18 . . . . .	28
4.1	Répartition des labels . . . . .	32
4.2	Analyse comparative des performances des modèles sur l'ensemble de test en fonction du nombre de paramètres . . . . .	34
4.3	Courbes de Perte d'entraînement et de Validation . . . . .	35
4.4	Grad-CAM . . . . .	36
4.5	Illustration de l'évolution de la perte de validation de quelques patients au fil des epochs . . . . .	36

# Chapitre 1

## Introduction

### 1.1 Organisme d'accueil

Fondé en 2007 par le professeur Gilles KARCHER, Nancyclotep est une plate-forme d'imagerie moléculaire dont l'objectif est de faciliter l'utilisation des techniques innovantes en imagerie moléculaire en pratique médicale courante et de répondre ainsi aux besoins des patients, du système de santé, et de la Recherche et du Développement (R&D) industriels ainsi que de la recherche académique.

Situé sur le site du Centre Hospitalier Universitaire (CHU) de Nancy (France), Nancyclotep dispose des atouts suivants :

- Un environnement hospitalier offrant un accès immédiat à une expertise médicale de haut niveau ainsi qu'à des patients pour des applications technologiques à usage clinique.
- Un laboratoire de radiochimie et radiopharmacie, une plateforme pour les études pré-cliniques, un laboratoire de production industrielle, une équipe dédiée aux traitements d'images (dans laquelle j'ai effectué mon stage), des équipements d'imagerie de haute qualité et une équipe d'e-learning proposant des solutions innovantes.
- Un accès aux radioéléments :  $^{18}\text{F}$ ,  $^{11}\text{C}$ ,  $^{89}\text{Zr}$ ,  $^{68}\text{Ga}$ ,  $^{225}\text{Ac}$ ,  $^{177}\text{Lu}$ , etc.

Nancyclotep a le statut juridique de Groupement d'Intérêt Economique (GIE) qui comporte 3 membres : le CHRU de Nancy, l'Université de Lorraine et la société POSIFIT en charge de la valorisation et des opérations pharmaceutiques.

Au cours de ce stage, j'ai aussi été en lien avec le laboratoire d'Imagerie Adaptative Diagnostique et Interventionnelle (IADI), qui une unité mixte de recherche Université de Lorraine/INSERM, qui développe des techniques et méthodes pour l'imagerie adaptative IRM, pour prendre en compte les mouvements physiologiques du patient et les intégrer aux processus de reconstruction et de traitement des images.

Le but est d'améliorer l'imagerie pour le diagnostic et aussi de développer *l'électro-imaging* qui combine les informations morphologiques et fonctionnelles avec les informations électriques des tissus.

IADI intervient dans des domaines très différents comme la physique, le traitement du signal, la méthodologie et les validations cliniques. Ses domaines d'application sont l'imagerie cardiaque, abdominale, musculosquelettique, mammaire et la médecine nucléaire.

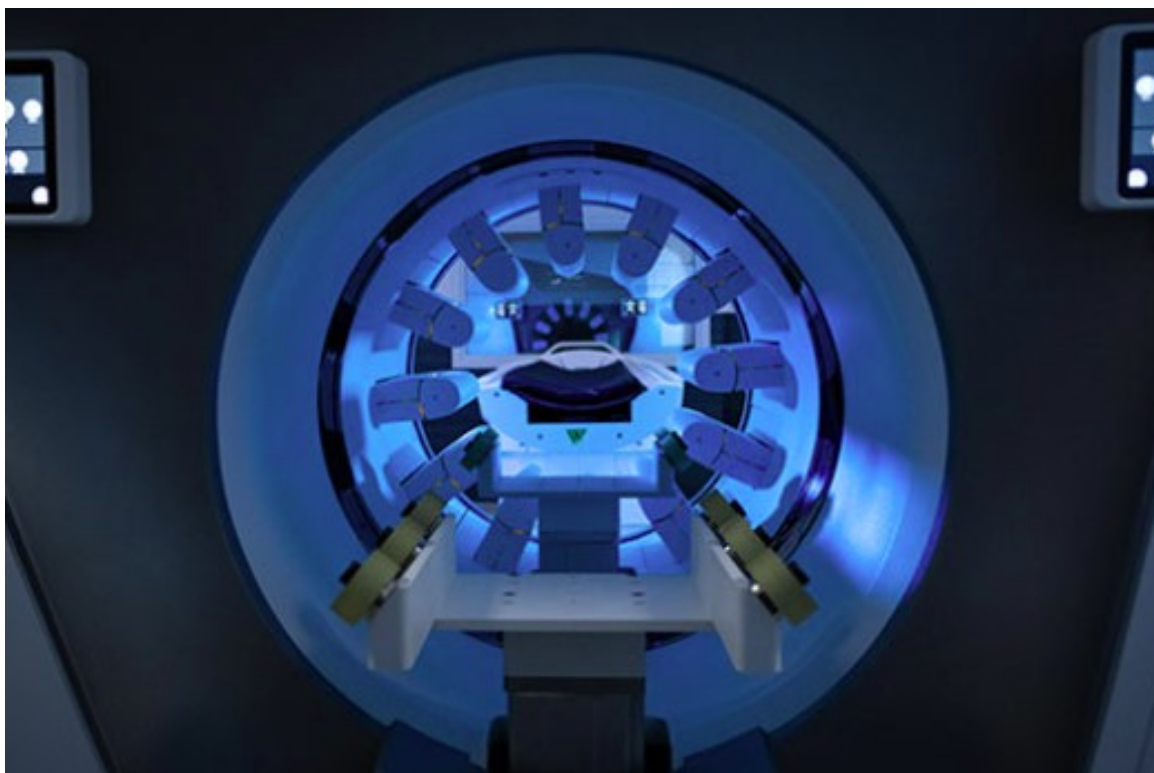


FIGURE 1.1 – Caméra de tomographie par émission monophotonique VERITON-CT utilisée dans le service de médecine nucléaire du CHU de Nancy

## 1.2 Contexte général et objectif du stage

Les gliomes représentent une catégorie de tumeurs cérébrales très hétérogènes, environ 80% des tumeurs malignes du système nerveux central (SNC) ([OSTROM et al. 2015](#)), parmi lesquelles le glioblastome est le plus fréquent et le plus agressif. Ces tumeurs ont un mauvais pronostic et un taux de récurrence élevé dans la première année suivant le traitement. Le pronostic pour les patients atteints de glioblastome reste sombre, avec une survie médiane d'environ 15 mois malgré les progrès dans les traitements multimodaux combinant chirurgie, radiothérapie et chimiothérapie ([LAMBORN et al. 2008](#)). La surveillance post-traitement des patients atteints de gliomes repose principalement sur l'imagerie par Résonance Magnétique (IRM). Cependant, l'IRM présente des limitations notables, notamment l'incapacité à distinguer de manière fiable entre la progression tumorale et les effets induits par les traitements, tels que la pseudoprogession et la radionécrose. ([ALBERT et al. 2016](#))

L'imagerie médicale, notamment l'imagerie TEP aux acides aminés (dont fait partie la  $^{18}\text{F}$ -FDOPA), est recommandée par le groupe d'experts (Response Assessment in Neuro-Oncology (RANO) Group) ([ALBERT et al. 2016](#)) en complément de l'IRM à différents stades de la maladie. Cette recommandation est faite pour les gliomes en récurrence dans le cadre du diagnostic différentiel entre la radionécrose, un changement induit par le traitement, et la véritable progression tumorale, car la TEP à la  $^{18}\text{F}$ -FDOPA est plus spécifique que l'IRM et offre une meilleure spécificité pour la détection des tissus tumoraux actifs. Ce qui la rend particulièrement utile pour la différenciation des récurrences tumorales des effets du traitement ([LAW et al. 2019](#)). De nombreuses études dans la littérature ont montré les très bonnes performances de la TEP aux acides aminés à partir d'analyses simples de l'image ([ROZENBLUM et al. 2023](#)) ou plus avancées avec une extraction massive de caractéristiques tumorales à partir d'un modèle d'intelligence artificielle pour la classification.



L'analyse d'images par intelligence artificielle (IA) quant à elle s'est particulièrement développée à partir des années 90 avec l'introduction d'algorithmes d'apprentissage profonds (deep learning) basés sur les réseaux de neurones, et l'augmentation des puissances de calcul des ordinateurs. Ainsi, ces méthodes, particulièrement puissantes pour la détection des objets, la segmentation et la classification ont rendu possibles des avancées technologiques.

Le sujet de stage consistera à développer un modèle de deep learning de type CNN permettant de faire la prédiction de la récurrence des gliomes pour espérer atteindre de meilleures performances.

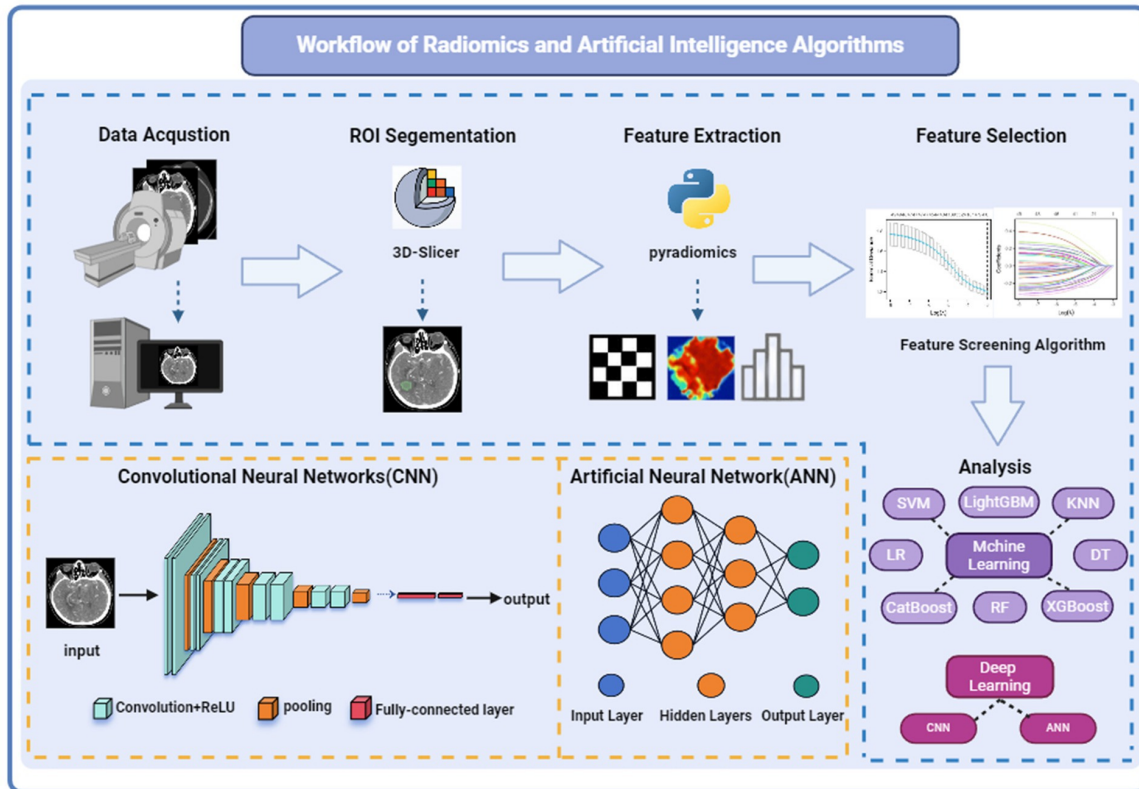


FIGURE 1.2 – Flux de travail et principes des algorithmes de radiomique et d'intelligence artificielle extrait de (PENG et al. 2024)

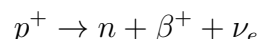
# Chapitre 2

## Etat de l'art

### 2.1 Imagerie TEP - Acquisition - Reconstruction d'images

La Tomographie par Émission de Positons (TEP) est une technique d'imagerie médicale qui connaît, depuis les années 2000, une très forte croissance. Elle s'est imposée comme une technique de choix pour le diagnostic, l'évaluation et la surveillance de pathologies dans de nombreux domaines tels que l'oncologie, la neurologie, les maladies cardiovasculaires, les infections et les inflammations. Son principe consiste à étudier, grâce à un appareil d'imagerie dédié appelé TEP couplé à un tomodensitomètre (TDM), la distribution dans le corps du patient d'un radiotracer, un radionucléide seul ou couplé à un vecteur biologique, administré par voie veineuse. Le vecteur choisi, qui peut être de nature différente (une cellule, une molécule, un anticorps, un acide aminé, ...), vise à cibler spécifiquement un processus biologique sans le perturber la répartition dans l'organisme d'une molécule radiomarquée appelée « médicament radiopharmaceutique », spécifique de la pathologie étudiée.

Les radiotraceurs d'intérêt en imagerie TEP sont basés sur un isotope en excès de protons revenant à son état stable par désintégration  $\beta^+$ . Les isotopes généralement utilisés possèdent une demi-vie relativement courte permettant de limiter la dose moyenne absorbée par les tissus et sont produits à l'aide d'un cyclotron ou d'un générateur. Lors de leur désintégration, un proton du noyau va être converti en neutron entraînant l'émission d'une particule  $\beta^+$ , le positon, et d'un neutrino électronique :



Pendant l'acquisition des images, le patient est positionné sur la table du TEP-Scan et doit rester immobile. Les détecteurs autour du patient captent les photons gamma  $\gamma$  produits par l'annihilation des positons émis par le radiopharmaceutique. Les données brutes recueillies sont utilisées pour estimer une image 3D de la distribution d'activité grâce au procédé de reconstruction tomographique.

De nos jours, l'algorithme itératif de référence en TEP est l'algorithme maximum likelihood expectation maximization (MLEM), un algorithme itératif statistique permettant de modéliser le bruit impactant les mesures selon une distribution de Poisson et reconstruisant l'image en maximisant la vraisemblance entre les projections estimées et celles mesurées ([SHEPP et VARDI 1982](#)). En routine clinique, une version accélérée de MLEM, l'algorithme ordered subset expectation maximization (OSEM) est utilisée sur un grand nombre d'appareils ([HUDSON et LARKIN 1994](#)). L'intérêt des images TEP est d'avoir accès à une information permettant de quantifier le métabolisme étudié au sein du champ de vue à travers la concentration de radiotracer mesurée. ([ZARAGORI 2021](#)) .

## 2.2 Prédiction de la Récidive/Réurrence en TEP à la $^{18}\text{F}$ -FDOPA

La Tomographie par Émission de Positons (TEP) avec le traceur [ $^{18}\text{F}$ ]-F-DOPA est largement utilisée pour évaluer les récurrences et la progression des gliomes après traitement. Au sein de notre service, plusieurs études ont exploré l'efficacité des paramètres statiques et dynamiques de la TEP DOPA pour la détection des récurrences, en mettant l'accent sur les performances des paramètres utilisés en routine et l'analyse radiomique.

### Paramètres statiques et dynamiques de la TEP à la $^{18}\text{F}$ -FDOPA

Les images statiques de la TEP à la  $^{18}\text{F}$ -FDOPA sont couramment utilisées dans notre service pour identifier les récurrences des gliomes. Cependant, afin d'explorer l'apport potentiel des paramètres dynamiques, [ZARAGORI et al. 2020](#) ont mené une étude évaluant à la fois les paramètres statiques et dynamiques chez des patients soupçonnés de récurrence de gliomes (bas grade et haut grade) après traitement. Les résultats de cette étude montrent que les paramètres statiques, comme le ratio de fixation tumeur/cerveau sain (TBR) et le volume tumoral métabolique (MTV), sont des indicateurs solides de récurrence, avec des taux de précision atteignant 94%. Bien que cette étude ait montré une précision élevée de 94% avec les paramètres statiques de la TEP DOPA, elle présente certaines limites. En effet, les résultats ont été obtenus rétrospectivement dans un centre unique, ce qui peut limiter leur généralisation. De plus, le faible nombre relatif de cas vérifiés histologiquement invite à prendre avec précaution la précision très élevée rapportée dans cette étude.

restent les plus fiables pour distinguer la progression tumorale des changements post-traitement (TRC) dans une étude incluant uniquement les gliomes de haut grade. Cependant, pour les gliomes IDH-mutants, la pente des acquisitions dynamiques a montré une valeur ajoutée, soulignant ainsi l'intérêt d'une approche complémentaire pour certaines sous-populations de patients.

Une étude multicentrique menée par [ROZENBLUM et al. 2023](#) a confirmé que les acquisitions statiques de la TEP à la  $^{18}\text{F}$ -FDOPA, notamment le TBR moyen, restent les plus fiables pour distinguer la progression tumorale des changements post-traitement (TRC) (sensibilité 82 %, spécificité 71%,  $p < 0,0001$ ). Cette étude inclue uniquement les gliomes de haut grade.

### Analyse radiomique et avancées

Les avancées dans l'analyse radiomique, qui consiste à extraire des caractéristiques quantitatives complexes à partir des images TEP, ont montré des résultats prometteurs pour améliorer la détection des récurrences. L'étude de [AHRARI et al. 2021](#), menée dans le cadre de notre service, a approfondi cette approche en évaluant la valeur des radiomiques statiques et dynamiques pour différencier la progression des gliomes de haut grade des changements liés au traitement (TRC). Utilisant des modèles d'apprentissage automatique, tels que la régression logistique et le modèle XGBoost, cette étude a montré que les modèles radiomiques, en particulier ceux basés sur les caractéristiques dynamiques, apportaient une légère amélioration des performances diagnostiques par rapport aux modèles statiques. Les meilleurs modèles radiomiques ont présenté des performances légèrement supérieures à celles du modèle de référence basé sur les valeurs TBR, avec des AUC respectives de 0,834 contre 0,792 ( $p < 0,001$ ), indiquant une valeur ajoutée marginale des radiomiques statiques et dynamiques par rapport à l'analyse classique basée uniquement sur les valeurs TBR.

Le sujet de stage s'inscrit dans la continuité de l'étude de [AHRARI et al. 2021](#) en visant à développer un modèle de deep learning pour la prédiction de la récurrence des gliomes à partir

des données TEP DOPA. Contrairement à cette étude, qui se concentrait uniquement sur les gliomes de haut grade, ce projet englobe tous les types de gliomes, offrant ainsi une perspective plus large. De plus, étant donné que les études précédentes n'ont pas montré de véritable avantage de l'analyse dynamique des images, ce stage se concentre principalement sur les images statiques. Enfin, en utilisant des techniques de deep learning, nous visons à introduire des éléments d'explicabilité dans le modèle, tout en bénéficiant d'une plus grande flexibilité dans le traitement des données.

## 2.3 Concepts fondamentaux du deep learning

Dans la programmation traditionnelle, les humains fournissent aux ordinateurs un ensemble de règles pour résoudre une tâche donnée. Avec l'apprentissage automatique, les ordinateurs peuvent arriver à leur propre solution en apprenant à partir de données étiquetées.

Les réseaux de neurones artificiels, conçus pour imiter le fonctionnement des neurones du cerveau humain, permettent de trouver des relations complexes non linéaires entre les données d'entrée  $x$  et la sortie  $y$  à travers une fonction  $f$ .

Les réseaux convolutifs (Convolutional Neural Networks, CNN) représentent une avancée majeure dans le domaine de la reconnaissance d'images. Ils sont particulièrement efficaces pour analyser les images visuelles et sont souvent utilisés en coulisses pour la classification des images.

Le bloc de construction central des CNN est la couche de convolution. Cette couche est représentée par des poids qui permettent d'extraire certaines caractéristiques spécifiques des données d'entrée.

Après la convolution, une fonction d'activation ReLU est souvent appliquée. ReLU introduit la non-linéarité en remplaçant toutes les valeurs négatives par zéro. Cette fonction permet au réseau de capturer des motifs plus complexes et de mieux apprendre les relations non linéaires dans les données. Comparée à d'autres fonctions d'activation comme la sigmoïde, ReLU surmonte le problème de l'atténuation du gradient, ce qui permet aux modèles d'apprendre plus rapidement et de mieux performer.

Ensuite, des couches de pooling sont utilisées pour réduire la dimension spatiale de la représentation de l'image. Cela permet de réduire le nombre de paramètres du modèle et d'accélérer le processus d'apprentissage. La méthode la plus courante est le MaxPooling, qui conserve la valeur maximale dans une fenêtre de taille prédéfinie.

Les propriétés extraites sont ensuite aplaties et introduites dans un réseau de neurones entièrement connecté. Ces couches entièrement connectées sont responsables de l'extraction des caractéristiques finales en attribuant des poids aux caractéristiques extraites. Ces poids sont mis à jour de manière itérative pendant l'entraînement pour optimiser la performance du modèle sur la tâche de classification ou de régression qui suit.

### 2.3.1 Réseau de Neurones Convolutifs 3D

La récente mise à disposition de grandes bases de données d'images 3D a révolutionné plusieurs domaines de la recherche et de l'application, notamment en imagerie médicale. Ces avancées permettent aux chercheurs et aux praticiens de travailler avec des ensembles de données riches en informations, offrant une vision plus complète et détaillée des phénomènes observés.

Dans ce contexte, les réseaux de neurones convolutifs 3D se sont imposés comme des outils puissants pour exploiter pleinement ces données volumétriques. Contrairement aux réseaux convolutifs 2D, qui sont limités à des analyses sur des images planes, les CNN3D sont capables de capturer les relations spatiales entre les plans, en traitant des informations sur plusieurs

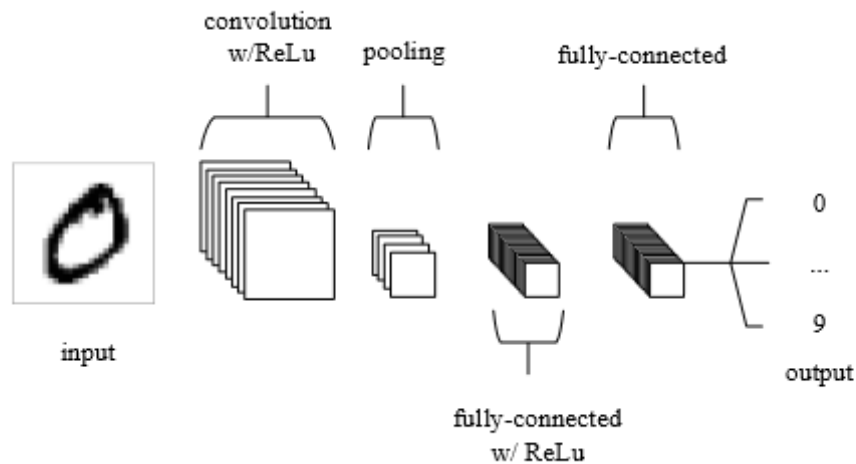


FIGURE 2.1 – Architecture CNN Basique extrait de O'Shea, K., et al. (2015)

dimensions simultanément.

Les réseaux de neurones convolutifs 3D utilisent des composants essentiels tels que la convolution 3D (Conv3D), le MaxPooling 3D.

Nous présenterons par la suite les architectures dérivées des CNNs que nous avons testées.

### 2.3.2 VGG (Visual Geometry Group)

Le modèle VGG (Visual Geometry Group) ou VGGNet, est une architecture de réseau neuronal convolutif profond proposé par [SIMONYAN et ZISSERMAN 2014](#).

Spécificités de VGG :

- Profondeur accrue : VGG se distingue par la grande profondeur de son réseau. C'est un des premiers modèles à montrer qu'une augmentation significative de la profondeur pouvait améliorer la performance sans rendre le réseau inentraînable.
- VGG utilise exclusivement de petits filtres Surmontant ainsi les limitations des architectures existantes. En effet, des architectures comme AlexNet utilisaient des couches convolutionnelles avec des champs réceptifs relativement grands, ce qui limitait leur capacité à capturer des détails fins

Il existe plusieurs variantes telles que VGG11, VGG16, VGG19. Le nombre décrit le nombre de couches et donc la profondeurs du réseau et le nombre de paramètres. Son architecture se compose de :

- **Entrée** : Une image d'entrée,
- **Couches de convolutions** : Une série de couches de convolution avec des filtres et une fonction d'activation ReLU,
- **Couche de pooling** : Après certaines couches de convolution, une couche de pooling est utilisée pour réduire la taille spatiale de la sortie.
- **Couches entièrement connectées** : Trois couches entièrement connectées suivies de fonctions d'activation ReLU.
- **Couche de sortie** : Une couche entièrement connectée avec une fonction d'activation sigmoïde pour la classification des classes.

### 2.3.3 Residual Network (ResNet)

Le modèle ResNet (Residual Network), en référence aux blocs résiduels qui composent l'architecture du réseau, est un type de réseau de neurones convolutif implémenté pour résoudre le problème de la dégradation des performances dans les réseaux très profonds. Ce modèle a été introduit pour la première fois en 2016 par [He et al. 2016](#) de Microsoft Research et a été proposé pour participer à l'ImageNet Large Scale Visual Recognition Challenge (ILSVRC) en 2015. Il a été conçu pour classer les images dans 1 000 catégories différentes, allant des objets courants aux animaux, aux véhicules, etc.

Soient  $\mathcal{H}(x)$  représentant la fonction de transformation que le réseau neuronal doit apprendre pour une entrée donnée  $x$  et  $\mathcal{F}(x)$  la fonction résiduelle.

L'idée théorique derrière ResNet repose sur l'apprentissage des résidus plutôt que d'apprendre directement la transformation souhaitée. Concrètement, au lieu d'optimiser la fonction  $\mathcal{H}(x)$ , ResNet optimise la fonction résiduelle définie par  $\mathcal{F}(x) = \mathcal{H}(x) - x$ , où  $x$  est l'entrée du réseau. Cette reformulation permet de rendre l'apprentissage plus accessible, en simplifiant le processus d'optimisation des réseaux très profonds.

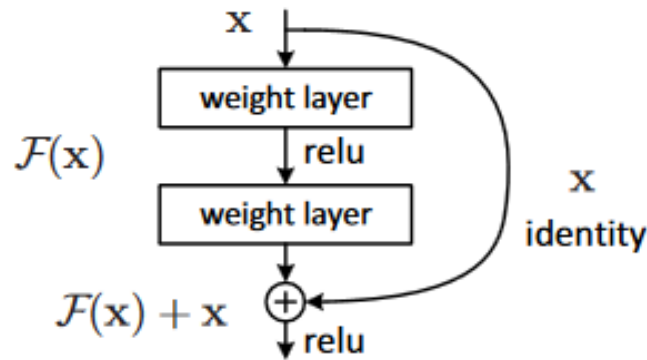


FIGURE 2.2 – Block résiduel. Extrait de [He et al. 2016](#)

L'architecture de base de ResNet est décrite comme suit :

- **Convolutional Block :**

Le réseau commence par une convolution d'entrée suivie de Batch Normalization, d'une activation ReLU, et d'une couche de MaxPooling. L'ensemble de ces couches constitue un bloc de convolution qui est responsable de l'extraction des caractéristiques initiales des données d'entrée.

- **Bloc Résiduel :**

Les blocs résiduels sont l'élément central de l'architecture ResNet. Chaque bloc résiduel inclut des couches de convolution, Batch Normalization et d'activations ReLU, avec une connexion résiduelle qui ajoute directement l'entrée du bloc à la sortie après les convolutions. Cette connexion permet de préserver les informations des couches précédentes et d'atténuer le problème de la dégradation des performances dans les réseaux profonds. Il existe deux types principaux de blocs résiduels :

- **Bloc Résiduel Basique (Basic Block) :**

Utilisé dans les variantes plus légères de ResNet, telles que ResNet-18 et ResNet-34. Il se compose de couches de convolution, chacune suivie de normalisation par lots et d'une activation ReLU.

- **Bloc Résiduel avancé (Bottleneck Block) :**

Présent dans les variantes plus profondes telles que ResNet-50, ResNet-101 et ResNet-

152. Ce bloc est plus complexe.

- **Partie Classification :**

La partie de classification se compose généralement d'une couche de pooling global (Global Average Pooling). Cette étape génère un vecteur résumant les statistiques pour chaque canal, permettant de capturer les informations globales de l'image tout en maintenant les relations entre les canaux. Elle est suivie d'une couche entièrement connectée qui projette les caractéristiques extraites dans l'espace des classes.

### 2.3.4 ACS Convolutions

Dans l'analyse des images médicales 3D, les méthodes basées sur des CNNs préentraînés sur des données 2D ne capturent pas efficacement les contextes volumétriques complexes. Bien que les approches purement 3D soient capables de modéliser ces contextes, elles nécessitent de grandes quantités de données spécifiques au domaine pour un préentraînement efficace, souvent inaccessibles en pratique.

Pour surmonter ces limitations, l'objectif est de tirer parti du transfert learning en utilisant des poids préentraînés sur des ensembles de données 2D massifs tout en intégrant des convolutions qui capturent les représentations 3D. [YANG et al. 2021](#) ont proposé une approche innovante en utilisant des convolutions ACS pour adapter des modèles préentraînés 2D à des tâches 3D. (Voir Figure 2.3)



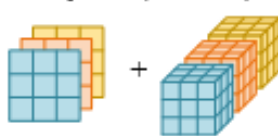
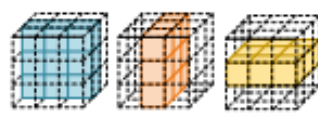
	Pros	Cons
<b>2D Convolutions</b> 	2D pretrained weights on large 2D datasets	Natively 2D representations
<b>3D Convolutions</b> 	Natively 3D representations	Lack of 3D pretrained weights on large datasets
<b>Hybrid (2D + 3D)</b> 	2D + 3D representations	a. 2D representation within 2D parts b. Lack of 3D pretrained weights c. Redundant multi-stage / multi-stream models
<b>ACS Convolutions</b> 	a. Natively 3D representations b. 3D pretrained weights on large 2D datasets c. Converting ANY 2D model into a 3D model seamlessly without extra computation costs	

FIGURE 2.3 – Comparaison entre les convolutions ACS proposées et les travaux antérieurs sur la modélisation des images médicales 3D. ([YANG et al. 2021](#))



## Principe

Les convolutions dans les réseaux de neurones sont essentielles pour capturer les relations spatiales dans les données, qu'il s'agisse d'images 2D ou de volumes 3D. La principale différence entre les convolutions 2D et 3D réside dans la manière dont ces relations sont capturées. Les convolutions 2D utilisent des noyaux 2D ( $C_o \times C_i \times K \times K$ ) pour des entrées 2D ( $C_i \times H_i \times W_i$ ), tandis que les convolutions 3D utilisent des noyaux 3D ( $C_o \times C_i \times K \times K \times K$ ) pour des entrées 3D ( $C_i \times D_i \times H_i \times W_i$ ), où  $C_i$  et  $C_o$  désignent les canaux d'entrée et de sortie,  $K$  désigne la taille du noyau, et  $D_i \times H_i \times W_i$  désigne la taille de l'entrée.

Lorsqu'il s'agit de transférer des modèles 2D préexistants pour qu'ils puissent traiter des données 3D, deux approches sont souvent utilisées dans le transfert des noyaux 2D vers 3D :

- **L'inflation des noyaux 2D en 3D** : Il s'agit d'augmenter les noyaux 2D en ajoutant une dimension supplémentaire, créant ainsi des noyaux 3D. Par exemple, un noyau 2D de taille  $K \times K$  est transformé en noyau 3D de taille  $K \times K \times K$ .
- **L'expansion des noyaux 2D en pseudo-3D** : ici, un noyau 2D est simplement étendu en ajoutant une dimension triviale, ce qui donne un noyau de taille  $1 \times K \times K$ . Bien que cette méthode permette de traiter des volumes 3D, elle peut ne pas capturer pleinement les relations 3D dans les données. D'où la mise en place des convolutions ACS.

Pour une entrée 3D  $X_i$  de dimensions  $C_i \times D_i \times H_i \times W_i$  (où  $C_i$  est le nombre de canaux, et  $D_i, H_i, W_i$  sont les dimensions du volume), les noyaux 2D sont réorganisés en trois sous-ensembles de noyaux ACS :

- $W_a$  pour la vue axiale ( $K \times K \times 1$ ),
- $W_c$  pour la vue coronale ( $K \times 1 \times K$ ),
- $W_s$  pour la vue sagittale ( $1 \times K \times K$ ).

Chaque sous-ensemble de noyaux est utilisé pour extraire les caractéristiques correspondantes des différentes vues du volume. Les résultats obtenus sont ensuite combinés pour produire une sortie 3D  $X_o$  qui intègre l'information des trois perspectives.

### 2.3.5 SENet (Squeeze-and-Excitation Networks)

SENet, ou Squeeze-and-Excitation Network (HU, SHEN et SUN 2018), est une architecture de réseau de neurones qui introduit un mécanisme appelé blocs *Squeeze-and-Excitation* (SE). Ces blocs sont conçus pour ajuster les réponses des canaux de caractéristiques en tenant compte des interdépendances entre eux.

Le bloc SE est basé sur une transformation  $F_{tr}$ , qui mappe une entrée  $X \in \mathbb{R}^{H' \times W' \times C'}$  vers des cartes de caractéristiques  $U \in \mathbb{R}^{H \times W \times C}$ . En général,  $F_{tr}$  est une opération de convolution, et l'ensemble des filtres appris est représenté par  $V = [v_1, v_2, \dots, v_C]$ , où chaque  $v_c$  correspond aux paramètres du filtre  $c$ . Les sorties peuvent être écrites comme  $U = [u_1, u_2, \dots, u_C]$ , avec

$$u_c = v_c * X = \sum_{s=1}^{C'} v_s^c * x_s, \quad (2.1)$$

où  $*$  désigne l'opération de convolution,  $v_c = [v_1^c, v_2^c, \dots, v_{C'}^c]$ , et  $X = [x_1, x_2, \dots, x_{C'}]$ . Dans cette équation,  $u_c \in \mathbb{R}^{H \times W}$  représente la sortie pour le  $c$ -ième canal, et  $v_s^c$  est un noyau spatial 2D appliqué à chaque canal correspondant de  $X$ .



### Squeeze : Agrégation de l'Information Globale

Ce processus consiste à compresser l'information spatiale d'une carte de caractéristiques en utilisant une opération de pooling global par moyenne.

### Excitation : Réajustement Adaptatif

La phase d'Excitation utilise l'information agrégée pour réajuster les canaux de manière adaptative. Le vecteur résultant de l'opération de compression passe à travers un mécanisme de calibrage ou *gating* activée par une fonction sigmoïde.

L'opération de *squeeze* est désignée par  $F_{sq}$ , tandis que l'opération de *gating* est désignée par  $F_{ex}$ .

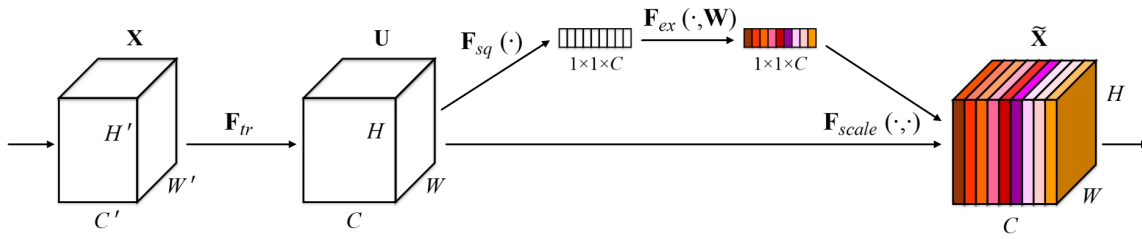


FIGURE 2.4 – A Squeeze-and-Excitation block (HU, SHEN et SUN 2018)

### 2.3.6 SqueezeNet

SqueezeNet (IANDOLA 2016) est une architecture de réseau de neurones convolutifs (CNN) conçue pour atteindre des niveaux de précision comparables à ceux des architectures de référence telles qu'AlexNet, mais avec un nombre bien plus réduit de paramètres. Ce qui permet de rendre les modèles plus efficaces en termes de mémoire et crucial pour le déploiement dans des environnements avec des ressources limitées.

#### Module Fire

L'architecture de SqueezeNet repose sur un composant essentiel appelé le module *Fire*. Ce module se compose d'une étape de compression utilisant des petits filtres, suivie d'une étape d'expansion qui intègre à la fois des petits filtres et des filtres de taille plus grande. Plus précisément, dans un module Fire, la couche de compression est constituée exclusivement de petits filtres, tandis que la couche d'expansion utilise un mélange de petits filtres et de filtres de taille plus grande. Les nombres de filtres dans chaque couche sont déterminés par des paramètres spécifiques, avec des filtres de tailles différentes utilisés dans l'étape d'expansion pour extraire des caractéristiques à différentes échelles.

#### Architecture de SqueezeNet

L'architecture de SqueezeNet est conçue pour être à la fois efficace et compacte tout en conservant une précision élevée. Voici une vue générale de cette architecture :

- **Couche Convolutionnelle Initiale** : SqueezeNet commence par une couche de convolution qui traite les données d'entrée.
- **Modules Fire** : Le réseau est constitué de plusieurs modules Fire. Le nombre de filtres dans chaque module augmente progressivement du début à la fin du réseau.

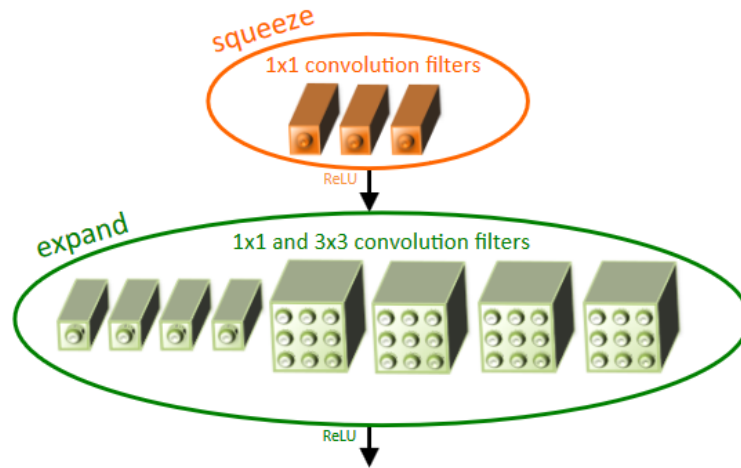


FIGURE 2.5 – Vue microarchitecturale : Organisation des filtres de convolution dans le module Fire Dans cet exemple,  $s_{1 \times 1} = 3$ ,  $e_{1 \times 1} = 4$ , et  $e_{3 \times 3} = 4$ . (IANDOLA 2016)

- **Couche Convolutionnelle Finale** : Le réseau se termine par une couche de convolution finale qui prépare les résultats pour la classification.
- **Max-Pooling** : Des opérations de max-pooling sont appliquées à différents points du réseau, après certaines couches spécifiques. Ce placement relativement tardif du pooling est conçu pour conserver des cartes d'activation plus grandes, ce qui peut améliorer la précision.
- **Design des Filtres** : Des petits filtres sont principalement utilisés pour réduire le nombre de paramètres, tandis que des filtres de taille plus grande sont employés dans les couches d'expansion pour capturer des caractéristiques plus complexes.
- **Zero-Padding** : Pour assurer que les activations de sortie des différents filtres aient la même hauteur et largeur, un bord de padding avec des zéros est ajouté autour des entrées des filtres de taille plus grande.
- **Activation ReLU** : La fonction d'activation ReLU est utilisée dans les couches de compression et d'expansion pour introduire de la non-linéarité.
- **Dropout** : Une technique de dropout est appliquée après le dernier module Fire pour réduire le risque de sur-apprentissage.
- **Absence de Couches Complètement Connectées** : Contrairement à d'autres architectures, SqueezeNet ne comporte pas de couches complètement connectées.

## 2.4 Optimisation et Paramétrage

### 2.4.1 Fonction de perte

Les fonctions de perte sont essentielles pour l'entraînement des modèles de deep learning, car elles mesurent l'écart entre les prédictions du modèle et les valeurs réelles. Dans cette section, nous décrivons les fonctions de perte que nous avons utilisées : Binary cross entropy, et Symmetric Cross Entropy Loss.

#### Binary cross entropy

La Binary Cross-Entropy (BCE) est une fonction de perte utilisée pour évaluer les performances d'un modèle de classification binaire. Elle quantifie la différence entre les prédictions du modèle

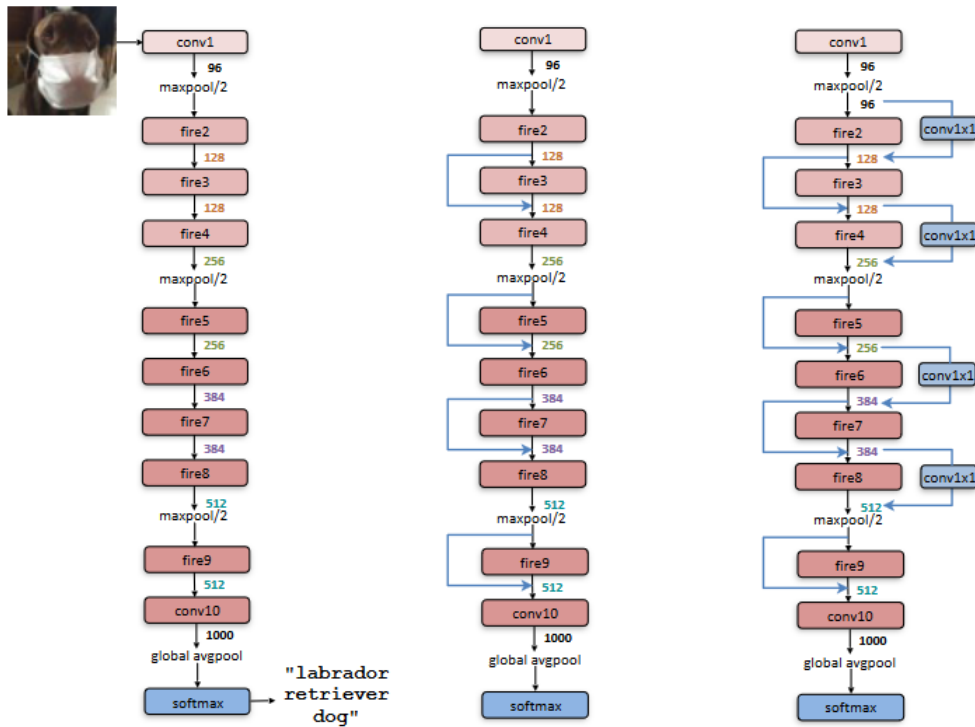


FIGURE 2.6 – Vue Macroarchitecturale de l'Architecture SqueezeNet. À gauche : SqueezeNet ; Au milieu : SqueezeNet avec un chemin de contournement simple ; À droite : SqueezeNet avec un chemin de contournement complexe (IANDOLA 2016)

(exprimées en probabilités) et les étiquettes cibles réelles, qui sont soit 0 (classe négative) soit 1 (classe positive).

La BCE est définie comme suit :

$$\text{BCE}(p, y) = -[y \log(p) + (1 - y) \log(1 - p)]$$

où :

- $p$  représente la probabilité prédite par le modèle pour que l'exemple appartienne à la classe positive.
- $y$  est l'étiquette cible, qui prend la valeur 0 ou 1.

La fonction BCE compare les probabilités prédites aux étiquettes réelles. Si le modèle prédit une probabilité proche de 1 pour un exemple dont l'étiquette est effectivement 1, la BCE sera faible, indiquant une bonne prédiction. À l'inverse, si le modèle prédit une probabilité incorrecte (proche de 0 pour une étiquette de 1, ou proche de 1 pour une étiquette de 0), la BCE sera élevée, indiquant une erreur significative.

Cette fonction de perte est couramment utilisée dans les tâches de classification binaire, car elle pénalise fortement les mauvaises prédictions, ce qui incite le modèle à apprendre à mieux séparer les deux classes.

## Symmetric Cross Entropy

L'entropie croisée classique (Cross Entropy, CE) est largement utilisée dans l'apprentissage supervisé des réseaux neuronaux profonds pour les tâches de classification. Cependant, elle présente des limitations notables lorsqu'elle est appliquée à des ensembles de données contenant

des étiquettes bruitées. En particulier, la CE tend à surapprendre les erreurs dans les classes "faciles" (celles avec des échantillons majoritairement corrects) et à sous-apprendre pour les classes "difficiles" (celles avec un plus grand nombre d'échantillons mal étiquetés). Pour surmonter ces limitations, [WANG et AL. 2019](#) ont proposé la Symmetric Cross Entropy (SCE), une méthode qui combine l'entropie croisée classique avec un terme supplémentaire robuste au bruit, appelé Reverse Cross Entropy (RCE).

La Symmetric Cross Entropy (SCE) est formulée comme suit :

$$\text{SCE} = \text{CE} + \text{RCE} = H(q, p) + H(p, q),$$

où  $H(q, p)$  est l'entropie croisée classique définie par :

$$H(q, p) = - \sum_{k=1}^K q(k|x) \log p(k|x),$$

et  $H(p, q)$  est l'entropie croisée inversée (Reverse Cross Entropy, RCE), définie par :

$$H(p, q) = - \sum_{k=1}^K p(k|x) \log q(k|x),$$

où  $K$  est le nombre de classes,  $q(k|x)$  est la distribution réelle, et  $p(k|x)$  est la distribution prédite par le modèle.

La perte totale de Symmetric Cross Entropy pour un échantillon  $x$  est alors donnée par :

$$\ell_{\text{sce}} = \ell_{\text{ce}} + \ell_{\text{rce}},$$

où  $\ell_{\text{ce}}$  est la perte d'entropie croisée classique et  $\ell_{\text{rce}}$  est la perte d'entropie croisée inversée.

La Symmetric Cross Entropy présente plusieurs avantages en comparaison avec l'entropie croisée classique :

- Robustesse au bruit : Le terme RCE est moins sensible aux étiquettes bruitées, ce qui permet de réduire le risque de surapprentissage sur des classes comportant des erreurs d'annotation.
- Meilleure généralisation : La combinaison de CE et RCE favorise un apprentissage plus équilibré entre les classes, améliorant ainsi la généralisation du modèle sur des données non vues.

## 2.4.2 Normalisation

La normalisation est une technique utilisée dans les réseaux de neurones pour améliorer la convergence de l'entraînement et la performance générale du modèle. Elle est utilisée pour normaliser les activations dans un réseau de neurones afin de stabiliser et d'accélérer l'apprentissage. Deux des méthodes courantes de normalisation sont la Batch Normalization et l'Instance Normalization.

### Batch Normalization

La Batch Normalization (BN) est une technique qui normalise les activations de chaque couche en utilisant les statistiques du mini-batch pendant l'entraînement. Cette méthode vise à réduire le problème de *covariate shift* interne, où la distribution des activations change au fur et à

mesure que les poids du réseau sont mis à jour. Pendant l'entraînement, pour chaque mini-batch, les activations sont normalisées en utilisant la moyenne et l'écart-type calculés sur la mini-batch :

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

où

- $x$  est l'activation de la couche,
- $\mu_B$  est la moyenne des activations de la mini-batch,
- $\sigma_B^2$  est la variance des activations de la mini-batch,
- $\epsilon$  est une constante.

### Instance Normalization

Tout comme la Batch Normalization, l'Instance Normalization (IN) ([ULYANOV et AL. 2016](#)) est une technique de normalisation utilisée pour ajuster les activations dans un réseau de neurones. Cependant, contrairement à la Batch Normalization, qui utilise les statistiques de la mini-batch, l'Instance Normalization normalise les activations à l'échelle de chaque instance d'image indépendamment.

Pour chaque instance d'image, l'Instance Normalization normalise les activations en utilisant la moyenne et l'écart-type calculés pour chaque instance :

$$\hat{x}_i = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

où

- $x_i$  est l'activation pour la  $i$ -ème instance,
- $\mu_i$  est la moyenne des activations pour la  $i$ -ème instance,
- $\sigma_i^2$  est la variance des activations pour la  $i$ -ème instance,
- $\epsilon$  est une constante.

La principale différence entre Batch Normalization et Instance Normalization réside dans la manière dont les statistiques sont calculées et appliquées.

En effet, Batch Normalization normalise les activations en utilisant les statistiques calculées sur l'ensemble de la mini-batch.

En revanche, Instance Normalization normalise les activations pour chaque instance d'image individuellement, ce qui la rend plus adaptée aux applications où les caractéristiques spécifiques de chaque image doivent être traitées séparément. De plus, elle est particulièrement utile lorsque la taille des batchs est faible, car dans ces cas-là, l'estimation de la moyenne et de l'écart-type sur le batch peut manquer de robustesse, entraînant des variations importantes entre les batchs. Cela permet à Instance Normalization de mieux gérer les fluctuations d'un batch à l'autre par rapport à des méthodes comme Batch Normalization.

### 2.4.3 Dropout

Le Dropout est une technique de régularisation utilisée pour prévenir le surapprentissage dans les réseaux de neurones. Pendant l'entraînement, Dropout met aléatoirement à zéro les sorties d'une couche spécifiée, avec une certaine probabilité. Cette approche empêche les neurones de devenir trop dépendants les uns des autres, ce qui réduit le risque de *co-adaptation* excessive des neurones.

Plus précisément, à chaque étape d'entraînement, Dropout ignore chaque unité avec une probabilité  $p$ , tandis que les unités restantes sont mises à l'échelle par un facteur de  $\frac{1}{1-p}$  pour

maintenir la somme totale des entrées inchangée.

Dropout est uniquement actif durant l'entraînement et peut être appliquée à la plupart des types de couches, y compris les couches entièrement connectées.

## 2.5 Cartes d'Attention 3D avec M3d-CAM

Les cartes d'attention sont des outils de visualisation et d'interprétation essentiels pour comprendre les décisions des modèles de deep learning. Elles permettent de déterminer quelles régions spécifiques des données d'entrée ont eu le plus d'impact sur la prédiction du modèle. En fournissant une représentation visuelle des parties des données qui influencent le plus les décisions du modèle, les cartes d'attention offrent une meilleure compréhension du comportement interne des réseaux de neurones.

Pour générer ces cartes d'attention, l'utilisation de techniques spécialisés est nécessaire pour traiter efficacement les données et produire des visualisations pertinentes. Dans ce contexte, MedCAM ([GOTKOWSKI et AL. 2020](#)) se distingue comme une bibliothèque spécialisée facilitant la création de cartes d'attention, avec un accent particulier sur les applications médicales. MedCAM offre des fonctionnalités avancées pour appliquer différentes méthodes de visualisation, permettant ainsi une interprétation approfondie des prédictions des modèles.

La méthode Grad-CAM, l'une des techniques proposées par MedCAM, utilise les gradients de la classe cible par rapport aux activations de la dernière couche convolutionnelle pour générer une *heatmap*. Cette *heatmap* met en évidence les régions de l'image qui ont joué un rôle clé dans la prédiction finale du modèle. Les zones avec des valeurs plus élevées sur cette *heatmap* sont celles qui ont eu le plus d'influence sur la décision du modèle.

# Chapitre 3

## Matériel et Méthodes

### 3.1 Description de la base de données

Notre étude se concentre sur des patients ayant été investigués par TEP au  $^{18}\text{F}$ -FDOPA dans deux institutions différentes : le CHRU de Nancy et l'hôpital de la Pitié-Salpêtrière à Paris, France. Les données du CHRU de Nancy ont été obtenues à partir de deux systèmes d'imagerie Biograph et Vereos. Les données provenant du CHRU de Nancy ont été utilisées pour l'entraînement du modèle, tandis que celles de l'hôpital de la Pitié-Salpêtrière à Paris, obtenues à partir de deux autres systèmes différents, ont servi à procéder à une validation externe. Initialement, la base de données d'entraînement comprenait 58 patients. Elle a été étendue à 100 puis à 180 patients. Après sélection et filtrage, le nombre final de patients retenus pour l'entraînement du modèle est de 164, tandis que 32 patients ont été utilisés pour les tests.

#### **Classification des Patients :**

Pour évaluer la progression de la récurrence tumorale et des changements dus au traitement (TRC), les diagnostics finaux ont été déterminés soit à partir de l'histopathologie, soit à partir du suivi clinique et radiologique pendant la période de suivi de 6 mois, selon les critères du groupe de travail RANO.

Les patients ont été classifiés en deux groupes distincts :

- Récurrence : Ce groupe inclut les patients pour lesquels le diagnostic a révélé une progression tumorale dans les 6 mois suivant le traitement.
- Changements dus au traitement (TRC) : Ce groupe comprend les patients chez lesquels les changements observés sont attribuables aux effets du traitement plutôt qu'à une progression tumorale

En utilisant ce seuil de 6 mois, nous avons pu catégoriser les patients en fonction de la durée pendant laquelle la maladie est restée stable après le traitement.

### 3.2 Techniques de prétraitement et d'augmentation d'images

#### 3.2.1 Prétraitement d'images

Dans notre processus de prétraitement, nous nous assurons que les images médicales sont correctement transformées et normalisées pour les rendre adaptées au passage dans le réseau de neurones. Dans le cas des images TEP, la conversion des données consiste simplement à transformer les images de  $\text{Bq.mL}^{-1}$  (Becquerels par millilitre) en  $SUV$  (Standard Uptake Value). Cette conversion prend en compte plusieurs paramètres physiologiques, comme le poids du patient et l'activité radioactive injectée.



Les images sont ensuite normalisées en utilisant la méthode min-max dans le but de mettre à l'échelle toutes les valeurs des pixels dans une plage uniforme entre 0 et 1. Ce qui permet d'accélérer la convergence.

Conformément aux recommandations IBSI (Image Biomarker Standardization Initiative) ([ZWANENBURG et AL. 2016](#)) , nous effectuons un rééchantillonnage des images pour les aligner sur une grille de résolution standardisée. Cela garantit une résolution uniforme avec des voxels de 2x2x2 mm et comparable entre les différentes images.

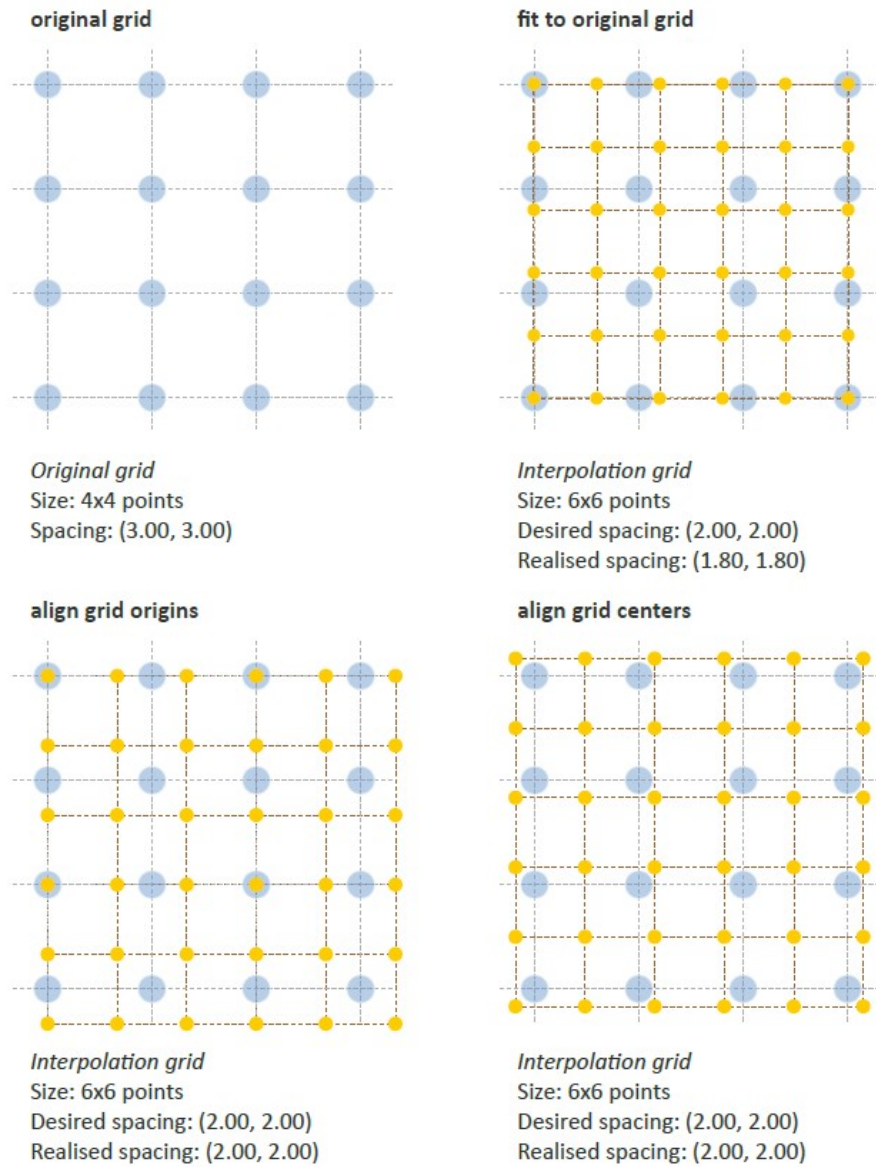


FIGURE 3.1 – Différentes interpolation basées sur une grille de  $4 \times 4$  avec un espacement de (3,00, 3,00) mm. Extrait de ([ZWANENBURG et AL. 2016](#))

Pour garantir que nos analyses se concentrent sur la région cérébrale, nous générons un masque du cerveau. Ce masque nous permet de centrer le cerveau au sein de l'image, assurant ainsi une position cohérente des images.

Toutes nos images ont été redimensionnées à la même taille (128, 128, 82) en utilisant un zero padding adapté autour du masque avant de les introduire dans le modèle.



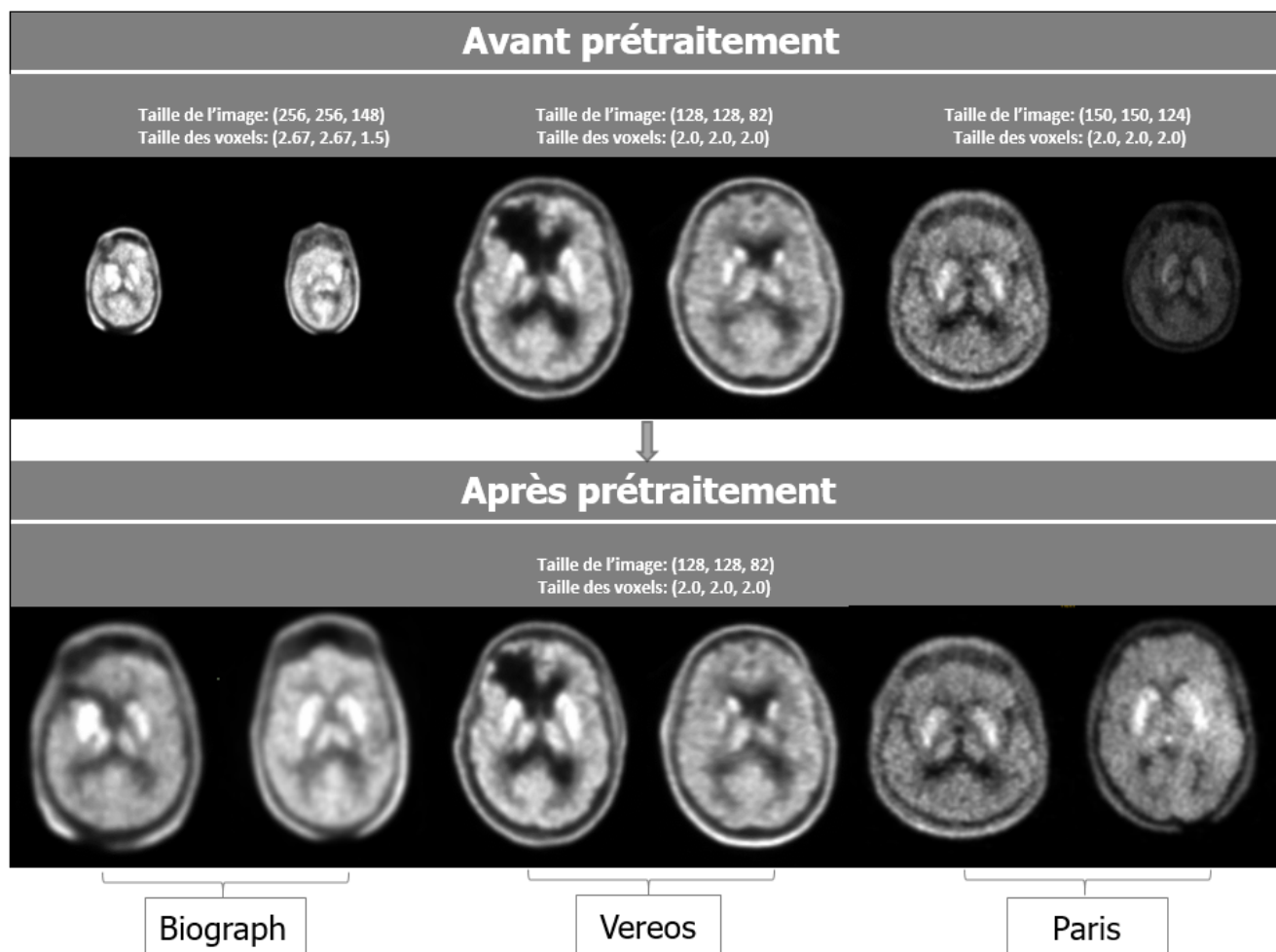


FIGURE 3.2 – Quelques images de patients avant et après prétraitement.

### 3.2.2 Techniques d'augmentation d'images

Le processus d'augmentation de données que nous avons utilisé est exactement le même que celui implémenté dans nnU-Net et repose sur une série de transformations de la librairie batch-generators (ISENSE et al. 2020) appliquées aux données d'entraînement pour améliorer la robustesse et la généralisation du modèle. Il inclut des techniques comme :

- Augmentations Spatiales : translation, déformations élastiques, mirror et rotations autour des axes  $Ox$ ,  $Oy$ ,  $Oz$ ...
- Augmentations de Couleur : luminosité, contraste...
- Augmentations de Bruit gaussien

Le chargement des données est optimisé grâce à l'utilisation de loaders permettant de charger les données par lots tout en appliquant les transformations. Un multi-threading est utilisé pour paralléliser le processus d'augmentation des données ce qui accélère considérablement le traitement et assure une bonne gestion de la mémoire.

### 3.3 Modélisation

#### Modèle de base : CNN3D

Notre modèle de base (CNN3D) prend en entrée un tenseur PyTorch, de forme  $(1, 128, 128, 82)$ , ce qui indique un canal, une résolution de  $128 \times 128$  voxels et une profondeur de 82.

L'architecture de notre modèle se compose de quatre blocs convolutionnels décrite de la façon suivante :

- Chaque bloc commence par une couche de convolution 3D avec des noyaux de taille  $3 \times 3 \times 3$ . Le nombre de filtres varie selon le bloc :
  - Premier et deuxième blocs : 64 filtres.
  - Troisième bloc : 128 filtres.
  - Quatrième bloc : 256 filtres.
- Une fonction d'activation ReLU, suivie d'une opération de MaxPooling 3D avec un noyau de  $2 \times 2 \times 2$ , est ensuite appliquée à la suite des couches de convolutions.
- À chaque bloc, nous utilisons l'Instance Normalization pour capturer des caractéristiques locales spécifiques à chaque image.

Après ces quatre blocs, nous appliquons une opération d'Average Pooling de taille  $(1, 1, 1)$ . Cette sortie est ensuite aplatie pour obtenir un vecteur 1D, qui passe à travers une couche linéaire de 512 neurones, suivie d'une activation ReLU et d'une régularisation Dropout d'un taux de 0,5 signifiant que signifie que 50% des neurones seront aléatoirement désactivés. Enfin, une seconde couche linéaire projette les caractéristiques dans l'espace des classes pour la prédiction. Cette couche linéaire a 2 neurones, correspondant aux deux classes de notre tâche.

En termes de paramètres, notre modèle de base compte un total de 5 414 081 paramètres, tous entraînaibles, ce qui en fait un modèle robuste et capable de capturer des représentations complexes des données d'entrée.

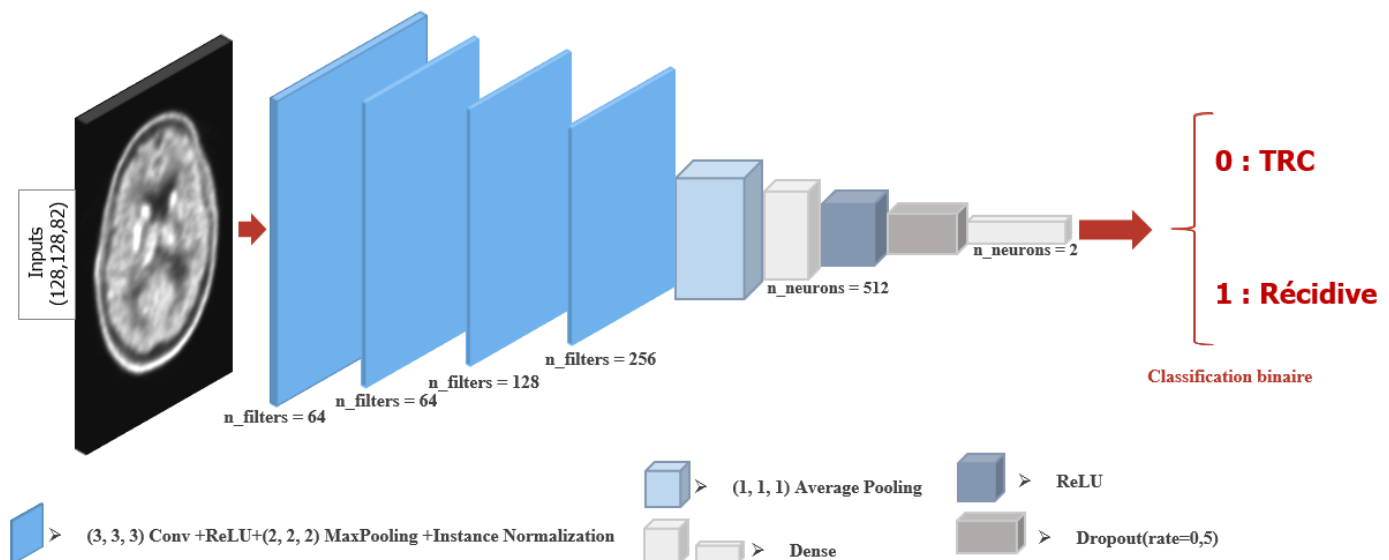


FIGURE 3.3 – Architecture CNN3D

## VGG11

Pour notre étude, nous avons adapté le modèle VGG11 pour traiter des images 3D. Ce modèle a été choisi en raison de sa simplicité relative par rapport aux variantes que VGG16 ou VGG19 qui sont plus complexes.

L'architecture de VGG11 se décrit comme suit :

- 8 couches convolutionnelles :
  - Chaque couche de convolution est suivie d'une opération de Batch Normalization et d'une activation ReLU.
  - Les couches convolutionnelles sont organisées en blocs successifs avec les nombres de filtres et sont séparées par des opérations de MaxPooling comme indiqué dans la séquence : 64, "M", 128, "M", 256, 256, "M", 512, 512, "M", 512, 512, "M".
- Une couche de Global Average pooling utilisée pour réduire les dimensions spatiales.
- Une couche entièrement connectée de 2 neurones.

Le modèle VGG11 adapté contient un total de 27,658,497 paramètres.

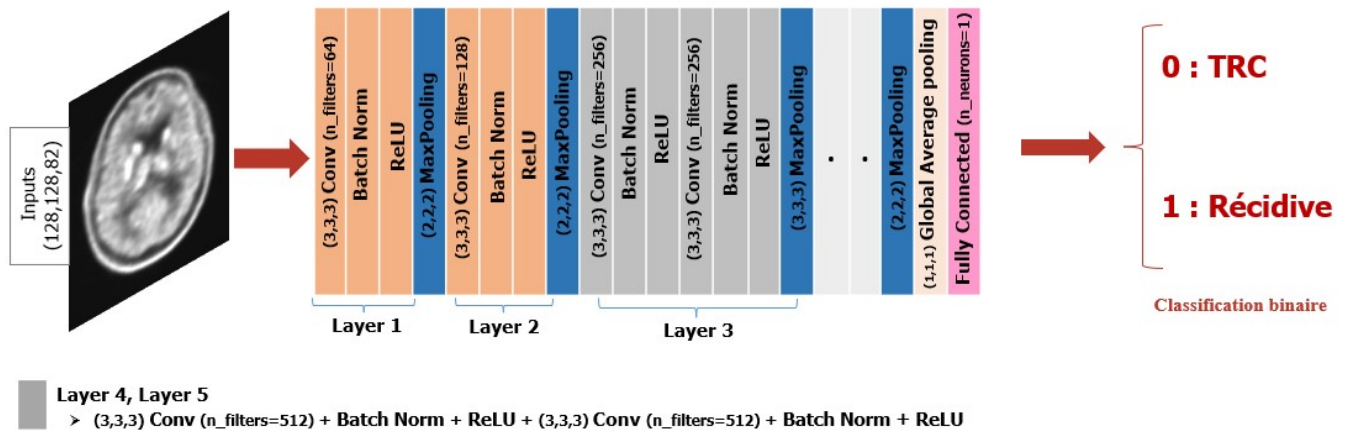


FIGURE 3.4 – Architecture VGG11

## ResNet10

Parmi les variantes de ResNet, nous avons testé ResNet10 en raison de sa complexité relative inférieure par rapport aux autres versions de ResNet. Sa structure est moins complexe que celle des variantes plus profondes comme ResNet18, ResNet34, ou ResNet50, ce qui permet une meilleure gestion des ressources de calcul.

Le modèle ResNet10 a été chargé via MONAI (Medical Open Network for AI), une bibliothèque open-source qui étend PyTorch pour les applications d'imagerie médicale.

L'entrée est un tenseur 5D correspondant à nos images de taille (128, 128, 82). L'enchaînement des couches dans le ResNet10 se décrit ainsi :

- 1 bloc convolutionnel qui comprend une couche Convolution, suivie d'une Batch Normalization (BN), d'une activation ReLU, et d'une opération de MaxPooling.
- 4 blocs résiduels :
  - Le premier bloc résiduel contient deux couches de convolution, chacune suivie d'une Batch Normalization et d'une activation ReLU.

- Les blocs résiduels 2, 3 et 4 contiennent chacun trois couches de convolution, suivies de Batch Normalization et d’une activation ReLU.
- Les blocs résiduels appliquent 64 filtres pour le premier bloc, 128 pour le deuxième, 256 pour le troisième, et 512 pour le quatrième.
- **Couche finale :**
  - Une couche de Global Average pooling utilisée pour réduire les dimensions spatiales à une taille de 1, 1, 1,
  - Suivi d’une couche linéaire qui produit la sortie finale.

Le modèle ResNet10 contient un total de 14,356,929 paramètres.

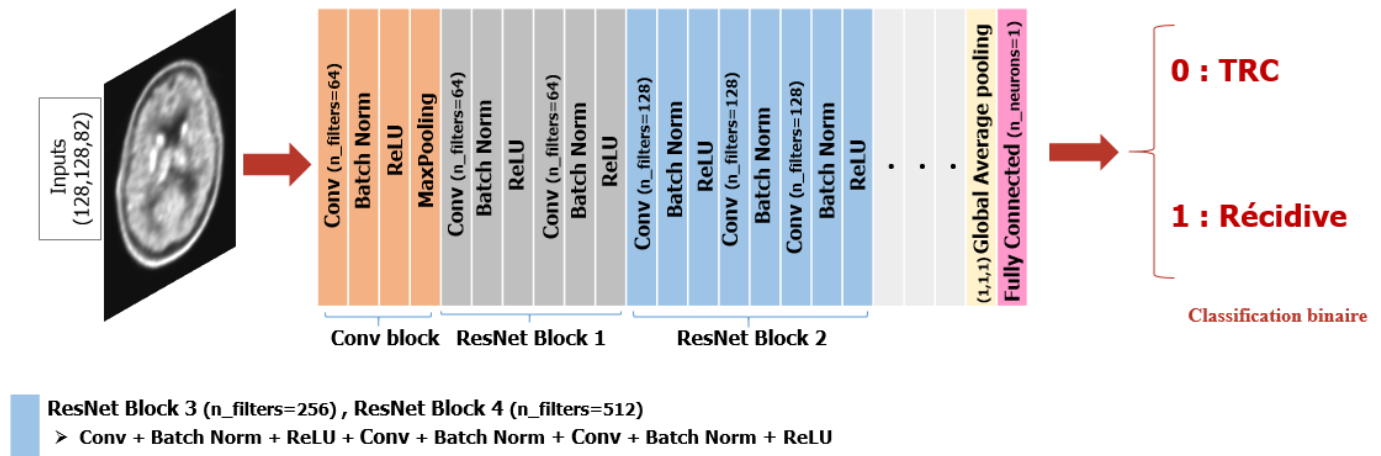


FIGURE 3.5 – Architecture ResNet10

### ACS-ResNet18

Nous avons opté pour le modèle ResNet18 combiné avec la méthode ACS afin de tirer parti des poids préentraînés de ResNet18 sur des données 2D pour les adapter efficacement notre tâches 3D.

Dans un premier temps, nous avons défini une classe *CustomResNet18* qui hérite de la classe *nn.Module* de PyTorch, permettant ainsi de personnaliser le modèle ResNet18 préexistant. Lors de l’initialisation, nous avons utilisé la bibliothèque *timm* pour créer le modèle, ce qui nous a permis de charger un ResNet18 préentraîné sur ImageNet. Étant donné que ce modèle était initialement conçu pour des images RGB (trois canaux en entrée), nous avons dû modifier la première couche de convolution (*conv1*) afin qu’elle accepte des images en niveaux de gris, qui ne possèdent qu’un seul canal. Concrètement, nous avons réduit le nombre de canaux d’entrée de cette couche de trois à un, tout en maintenant les autres paramètres (comme le nombre de canaux de sortie, la taille du noyau, le pas de la convolution et le biais) inchangés pour préserver la structure d’origine du modèle.

Ensuite, nous avons remplacé la dernière couche entièrement connectée du ResNet18, conçue pour produire des sorties pour 1000 classes (comme dans ImageNet (DENG et al. 2009)), par une nouvelle couche adaptée à notre tâche de classification binaire.

Après avoir personnalisé le modèle, nous avons appliqué une conversion à l’aide de la méthode *ACSCConverter* pour adapter notre modèle aux images médicales 3D, tout en utilisant les poids des convolutions 2D préentraînés.

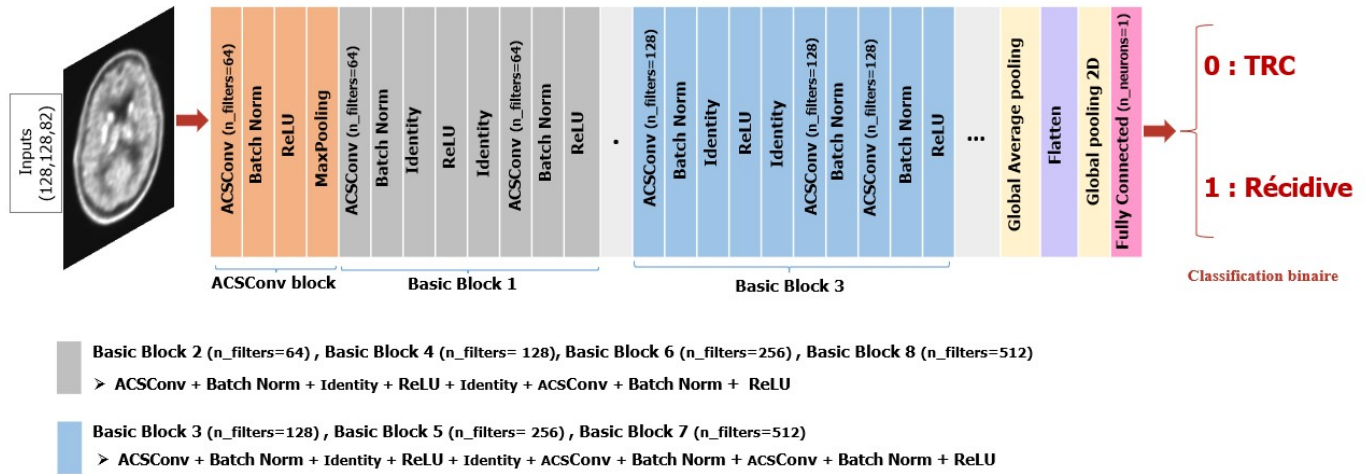


FIGURE 3.6 – Architecture ACS-ResNet18

## SENet (Squeeze-and-Excitation Networks)

Nous avons également testé le modèle SENet via MONAI en raison de capacité à représenter les caractéristiques importantes en ajustant dynamiquement l'importance de chaque canal, conduisant à une amélioration des performances et à une meilleure efficacité du modèle. Nous avons configuré le modèle avec les paramètres suivants :

- **Bloc de base** : Nous avons opté pour un bloc de type *Squeeze-and-Excitation*
- **Configuration des couches** (*layers*) : nous avons choisi une structure simple avec une seule couche par étage.
- **Groupes de convolutions** (*groups*) : Nous avons appliqué des convolutions standard sans partitionnement en groupes.
- **Facteur de réduction** : Avec un facteur *reduction* de quatre, nous ajustons le facteur de réduction dans les blocs Squeeze-and-Excitation, ce qui influence la taille de la couche intermédiaire lors des opérations d'excitation.

En tout, le modèle comprend 30 couches de convolution, 4 blocs Squeeze-and-Excitation (SE), et 9 couches entièrement connectées un total de 159,671,426 paramètres.

## SqueezeNet

Les modèles précédemment examinés présentent tous un nombre élevé de paramètres, posant des soucis en termes de mémoire et possiblement de surapprentissage vu notre base de données limitée, ce qui nous a conduits à explorer le modèle SqueezeNet, qui, lui, possède un nombre de paramètres bien moins élevé.

Son architecture comprend :

- **Couches de Convolution** : 16 couches distinctes de convolution suivie chacune d'une Batch Normalization et d'une activation ReLU et d'une opération de MaxPooling 3D
- **Module Fire** : 6 blocs Fire chacune contenant une opération de *squeeze* suivie d'une opération d'*expand*
- **Dropout** : Utilisée pour régulariser le modèle en éteignant aléatoirement une fraction des neurones pendant l'entraînement

Le modèle SqueezeNet présente un total de 1,861,186 paramètres, ce qui est significativement inférieur à celui des autres modèles examinés

## 3.4 Entraînement et Validation

### 3.4.1 Environnement de travail

Nous avons réalisé l'entraînement et la validation des modèles dans un environnement virtualisé sous Docker, en utilisant une carte graphique NVIDIA A100-PCIE-40GB avec 40 Go de mémoire GPU. Le développement a été effectué en Python (version 3.10.13) avec la bibliothèque de deep learning PyTorch.

### 3.4.2 Validation Croisée et Équilibrage des Données

Nous avons appliqué la validation croisée stratifiée à 10 plis pour évaluer les performances des modèles. Cette technique permet de maintenir une répartition proportionnelle des classes dans chaque pli, minimisant ainsi le risque de biais. À chaque itération de la validation croisée, 9 sous-ensembles ont été utilisés pour l'entraînement, tandis que le 10<sup>e</sup> sous-ensemble servait à la validation.

Pour un équilibrage optimal des données, nous avons utilisé une stratégie combinant les étiquettes des patients et les informations relatives au dispositif d'imagerie utilisé, garantissant ainsi une représentation équitable des différentes modalités d'imagerie dans les données d'entraînement.

### 3.4.3 Configuration de l'Entraînement

Les hyperparamètres de l'entraînement sont les suivants :

- Taille de batch : 10
- Nombre d'epochs : 150
- Taux d'apprentissage : 1e-4
- Décroissance du poids : 1e-3

L'optimiseur Adam a été utilisé pour l'entraînement initial sans augmentation de données. Par la suite, une augmentation des données a été mise en place pour améliorer la généralisation. Ainsi, à chaque epoch, un nombre défini de batch a été traité, avec génération dynamique d'images augmentées. Cette approche a permis d'exposer le modèle à une plus grande variété d'exemples à chaque itération.

Un *scheduler* a été appliqué pour ajuster dynamiquement le taux d'apprentissage en fonction de la perte de validation du modèle. Plus précisément, si la perte sur le jeu de validation ne s'améliorait pas après 15 itérations, le *scheduler* réduisait automatiquement le taux d'apprentissage de 50%.

### 3.4.4 Prédiction

Pour chaque patient, les prédictions ont été calculées sur chacun des plis lors de la validation croisée. L'évaluation sur l'ensemble de validation s'est faite en prenant la prédiction obtenue lorsque le patient faisait partie de l'ensemble de validation pour le pli correspondant.

Quant à l'évaluation sur les données de test, elle a été réalisée en utilisant une approche d'ensemble : pour chaque patient, nous avons combiné les prédictions des différents modèles issus de chaque pli en calculant la moyenne des probabilités.

Ensuite, nous avons calculé les métriques de performance à partir de ces prédictions agrégées, aussi bien sur les données de validation que sur les données de test.

## 3.5 Plan d'expériences

Nous avons tout d'abord concentré nos efforts sur l'optimisation de notre modèle de base, CNN3D, en réalisant plusieurs ajustements. Nous avons réduit les canaux d'entrée des couches de convolution (64, 32, 16) et diminué parallèlement le nombre de neurones dans la première couche entièrement connectée (512, 256, 128). Ces ajustements combinés avaient pour but de simplifier la structure du réseau tout en réduisant le nombre de paramètres afin de mieux contrôler le surajustement.

Lors des tests, nous avons constaté que la data augmentation (DA) améliorait les performances du modèle de base. En conséquence, tous les modèles suivants ont également été testés avec DA, afin de garantir des comparaisons justes et de maximiser les performances de chaque architecture.

Nous avons évalué différents optimiseurs, dont Adam et SGD, pour déterminer celui qui permettait la meilleure convergence. De plus, nous avons ajusté plusieurs hyperparamètres tels que le batch size (entre 4 et 10) et le taux d'apprentissage (1e-3 et 1e-4), afin de trouver la configuration optimale. Enfin, nous avons appliqué une validation croisée avec 5 et 10 folds pour assurer la robustesse des résultats et leur capacité à généraliser.

En plus du modèle de base, nous avons testé les autres architectures, notamment SENet, qui a été évalué avec les deux fonctions de perte, BCE (Binary Cross-Entropy) et SCE (Symmetric Cross-Entropy), et les résultats ont été meilleurs avec SCE Loss. Suite à cette observation, nous avons appliqué SCE Loss à d'autres modèles, notamment le modèle de base CNN3D et SqueezeNet, un modèle plus léger, pour évaluer s'ils pouvaient également bénéficier de cette fonction de perte.

## 3.6 Métriques d'évaluation

Pour évaluer la performance de nos modèles de deep learning, nous utilisons quatre métriques principales : l'Accuracy, l'AUC (Area Under the Curve), la Balanced Accuracy, et l'Average Precision (AP).

### - Accuracy

L'Accuracy est la proportion de prédictions correctes parmi toutes les prédictions faites par le modèle. Elle est calculée comme suit :

$$\text{Accuracy} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

### - AUC (Area Under the Curve)

L'AUC mesure la capacité du modèle à distinguer entre les classes positives et négatives en traçant la courbe ROC (Receiver Operating Characteristic) et en calculant l'aire sous cette courbe.

Une AUC proche de 1 indique une excellente capacité du modèle à séparer les classes positives des négatives, tandis qu'une AUC proche de 0.5 indique une performance aléatoire.

**- Balanced Accuracy**

La balanced accuracy est une version améliorée de l'accuracy, particulièrement utile pour les jeux de données déséquilibrés. Ses caractéristiques sont :

- Donne un poids égal aux classes positives et négatives
- Calculée comme la moyenne arithmétique de la sensibilité (taux de vrais positifs) et de la spécificité (taux de vrais négatifs)
- Formule :

$$\text{Balanced Accuracy} = \frac{1}{2} \left( \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right)$$

où :

- $TP$  est le nombre de vrais positifs,
- $FN$  est le nombre de faux négatifs,
- $TN$  est le nombre de vrais négatifs,
- $FP$  est le nombre de faux positifs.

Elle donne une évaluation plus équitable de la performance du modèle pour chaque classe. Elle équilibre les taux de détection des classes positives et négatives, ce qui permet de mieux comprendre la performance globale dans un contexte de déséquilibre.

**- Average Precision (AP)**

Average Precision calcule la précision moyenne à partir des scores de prédiction. Sa valeur est comprise entre 0 et 1, où une valeur plus élevée indique une meilleure performance du modèle. La Précision Moyenne est définie comme suit :

$$AP = \sum_n (R_n - R_{n-1}) P_n$$

où  $P_n$  et  $R_n$  représentent respectivement la précision et le rappel au seuil  $n$ .

De plus, nous avons calculé les intervalles de confiance à 95% des métriques en utilisant la méthode percentile bootstrap, afin de fournir une estimation robuste de la variabilité des performances du modèle. Les intervalles de confiance (IC) fournissent des indications importantes sur la stabilité et la fiabilité des performances des modèles. En général, les IC pour les métriques de performance des modèles montrent une variabilité qui reflète la dispersion des résultats sur les ensembles de validation et de test.



# Chapitre 4

## Résultats et discussion

### 4.1 Distribution des patients par classe

La répartition des 164 patients dans le jeu de données d'entraînement est de **(Figure 4.1)** :

- 46 patients (28%) dans la classe TRC
- 118 patients (72%) dans la classe Récidives

Les données de test sont répartis comme suit :

- 6 patients (18,7%) sont dans la classe TRC
- 26 patients (81,3%) sont dans la classe Récidives

Cette répartition montre une nette prépondérance de la classe Récidives dans les deux ensembles, avec une proportion particulièrement élevée dans les données de test.

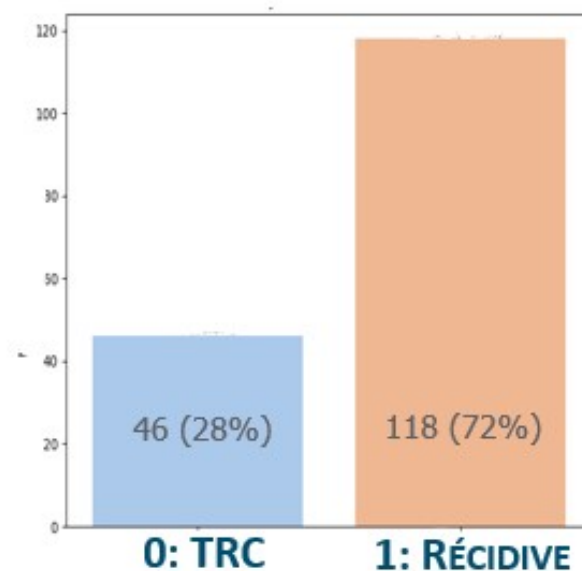


FIGURE 4.1 – Répartition des labels

### 4.2 Performances des Modèles

Les résultats des performances des différents modèles sur les ensembles de données de validation et de test sont présentés dans les tableaux ci-dessous.

TABLE 4.1 – Résultats des modèles sur l'ensemble de validation avec intervalles de confiance.

Modèle	Accuracy	Balanced Accuracy	AUC	AP
CNN3D	0.78 [0.72, 0.84]	0.72 [0.64, 0.80]	0.84 [0.77, 0.90]	0.93 [0.89, 0.96]
CNN3D (Avec DA)	0.85 [0.80, 0.90]	0.82 [0.76, 0.89]	0.92 [0.87, 0.96]	0.97 [0.94, 0.99]
VGG11	0.83 [0.77, 0.89]	0.79 [0.72, 0.87]	0.87 [0.80, 0.93]	0.94 [0.90, 0.97]
ResNet10	0.77 [0.71, 0.84]	0.76 [0.68, 0.83]	0.83 [0.76, 0.90]	0.93 [0.88, 0.96]
ACS ResNet18	0.80 [0.73, 0.86]	0.75 [0.67, 0.82]	0.87 [0.80, 0.92]	0.93 [0.88, 0.97]
SENet	0.80 [0.74, 0.86]	0.76 [0.70, 0.84]	0.86 [0.80, 0.92]	0.94 [0.90, 0.97]
SqueezeNet	0.84 [0.79, 0.90]	0.80 [0.73, 0.88]	0.86 [0.79, 0.93]	0.92 [0.87, 0.97]

TABLE 4.2 – Résultats des modèles sur le testset

Modèle	Accuracy	Balanced Accuracy	AUC	AP
CNN3D	0.69	0.68	0.70	0.89
CNN3D (Avec DA)	0.75	0.60	0.59	0.86
VGG11	0.65	0.66	0.73	0.93
ResNet10	0.78	0.48	0.75	0.94
ACS ResNet18	0.65	0.66	0.76	0.94
SENet	0.84	0.84	0.82	0.95
SqueezeNet	0.72	0.63	0.65	0.90

Les performances des différents modèles varient selon les métriques évaluées :

- **CNN3D** : Ce modèle présente une accuracy de 0.78 et une AUC de 0.84 sur l'ensemble de validation, montrant qu'il parvient à bien capturer les caractéristiques des données d'entraînement. Cependant, ses performances diminuent sur l'ensemble de test, avec une accuracy de 0.69 et une AUC de 0.70, suggérant une difficulté à généraliser sur des données nouvelles, malgré une AP élevée de 0.89. Ce qui révèle un manque de généralisation significatif.

- **CNN3D (Avec Data Augmentation)** : L'ajout de techniques de data augmentation améliore les performances sur l'ensemble de validation, avec une accuracy de 0.85 et une AUC de 0.92, indiquant que ces techniques aident à rendre le modèle plus robuste. Cependant, les performances sur l'ensemble de test montrent une accuracy de 0.75 et une AUC de 0.59, suggérant que, malgré une amélioration générale, le modèle reste sensible aux variations dans les nouvelles données. La balanced accuracy de 0.65 sur l'ensemble de test montre une amélioration notable dans la gestion des classes déséquilibrées, mais des défis subsistent.

- **VGG11** : Ce modèle montre de bonnes performances sur l'ensemble de validation avec une accuracy de 0.83 et une AUC de 0.87. Toutefois, ses performances sont légèrement plus faibles sur l'ensemble de test, avec une accuracy de 0.65 et une AUC de 0.73. Bien que l'AP de 0.93 soit élevée, indiquant une bonne capacité de précision.

- **ResNet10** : ResNet10 présente une accuracy de 0.77 et une AUC de 0.83 sur l'ensemble de validation. Cependant, sur l'ensemble de test, bien que l'accuracy augmente légèrement à 0.78, l'AUC diminue à 0.75. Cette diminution de l'AUC indique une perte de performance en termes de capacité de discrimination sur les nouvelles données. De plus, la balanced accuracy sur l'ensemble de test est de 0.48, ce qui révèle une difficulté notable à gérer les classes déséquilibrées. Cette difficulté pourrait expliquer la baisse générale des performances observée sur l'ensemble de test par rapport à l'ensemble de validation.

- **ACS ResNet18** : ACS ResNet18 présente une accuracy de 0.80 et une AUC de 0.87 sur l'ensemble de validation. Sur l'ensemble de test, l'accuracy tombe à 0.65 et l'AUC reste stable à 0.76, ce qui suggère que le modèle parvient à maintenir un bon niveau de discrimination, bien que sa capacité à généraliser soit limitée. La balanced accuracy de 0.53 sur l'ensemble de test montre une performance moyenne en termes de gestion des classes déséquilibrées.

- **SENet** : SENet se distingue par de bonnes performances sur l'ensemble de validation, avec une accuracy de 0.80 et une AUC de 0.86. Sur l'ensemble de test, on a une accuracy de 0.84 et une AUC de 0.82, indiquant que ce modèle parvient à généraliser efficacement et à maintenir des performances élevées même sur de nouvelles données. De plus, une balanced accuracy de 0.72 sur l'ensemble de test souligne sa capacité à bien gérer les classes déséquilibrées, renforçant ainsi ses bonnes performances globales.

- **SqueezeNet** : Ce modèle, affiche une accuracy de 0.84 et une AUC de 0.86 sur l'ensemble de validation. Les performances chutent sur l'ensemble de test à 0.72 pour l'accuracy et 0.65 pour l'AUC, ce qui indique que SqueezeNet, bien que performant sur les données d'entraînement, a des difficultés à généraliser.

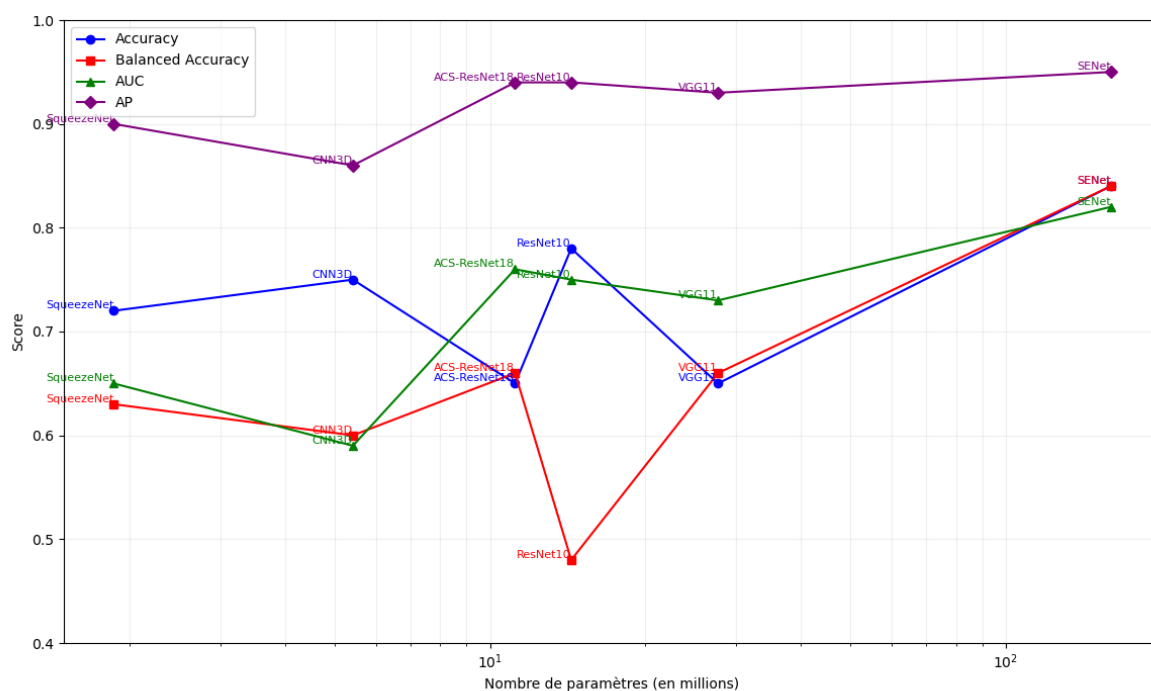


FIGURE 4.2 – Analyse comparative des performances des modèles sur l'ensemble de test en fonction du nombre de paramètres

Le modèle SENet avec le plus grand nombre de paramètres a les meilleures performances. (Voir Figure 4.2)

## 4.3 Discussion

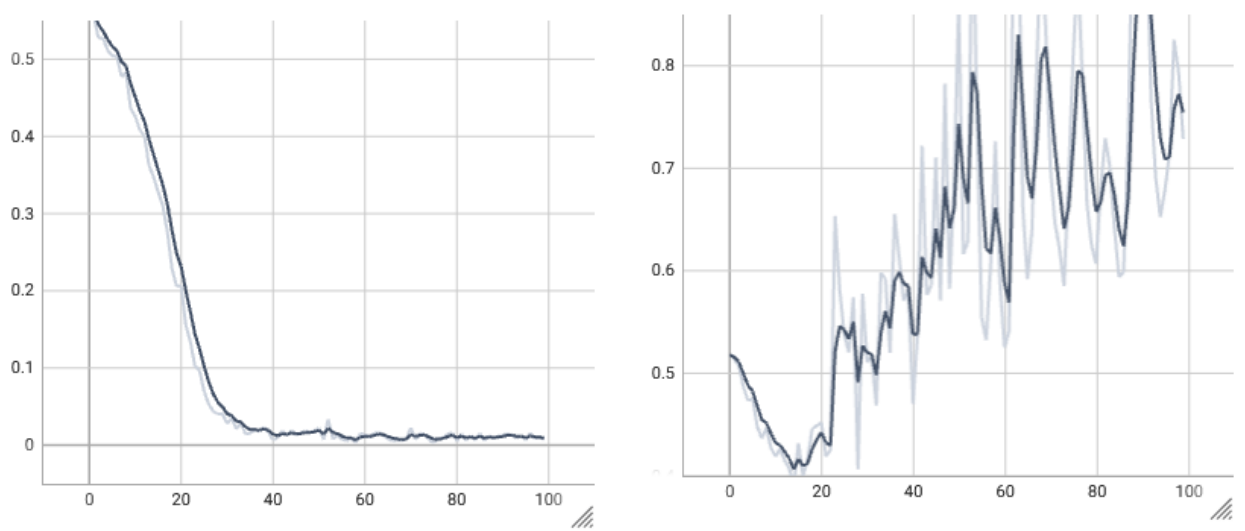
Les résultats de cette étude soulignent l'énorme potentiel des réseaux de neurones convolutifs (CNNs) pour l'analyse des images TEP à la  $^{18}\text{F}$ -FDOPA dans la détection des récidives de

gliomes. Cependant, plusieurs défis ont été rencontrés tout au long de l'étude, notamment liés à la taille de la base de données et aux problèmes de surapprentissage.

### Surapprentissage

Nous avons observé des signes potentiels de surapprentissage dans notre modèle. En effet, la courbe de perte sur l'ensemble d'entraînement continuait de diminuer progressivement au fil des epochs, tandis que la perte sur l'ensemble de validation stagnait, voire augmentait. Ce phénomène suggère que le modèle apprenait trop bien les détails et les bruits spécifiques des données d'entraînement, au détriment de sa capacité à se généraliser aux nouvelles données.

En réponse à ces signes de surapprentissage, nous avons mis en œuvre des techniques de data augmentation pour enrichir notre base de données et améliorer la capacité de généralisation du modèle. De plus, nous avons également essayé de diminuer le nombre de paramètres du modèle afin de réduire la complexité et d'éviter le surajustement. Cependant, malgré ces efforts, les gains en performance n'ont pas toujours été aussi significatifs que prévu.



1- Courbes de Perte d'entraînement

2- Courbes de Perte de Validation

FIGURE 4.3 – Courbes de Perte d'entraînement et de Validation

En outre, nous avons utilisé des cartes d'attention générées par MedCAM en combinaison avec la méthode Grad-CAM pour identifier les zones que le modèle apprenait à reconnaître. Nous avons découvert que le modèle se concentrait parfois sur des zones non pertinentes telles que les parties crâniennes plutôt que sur les régions cérébrales spécifiques aux récurrences de gliomes. Cette découverte a conduit à modifier notre preprocessing avec le centrage du cerveau.

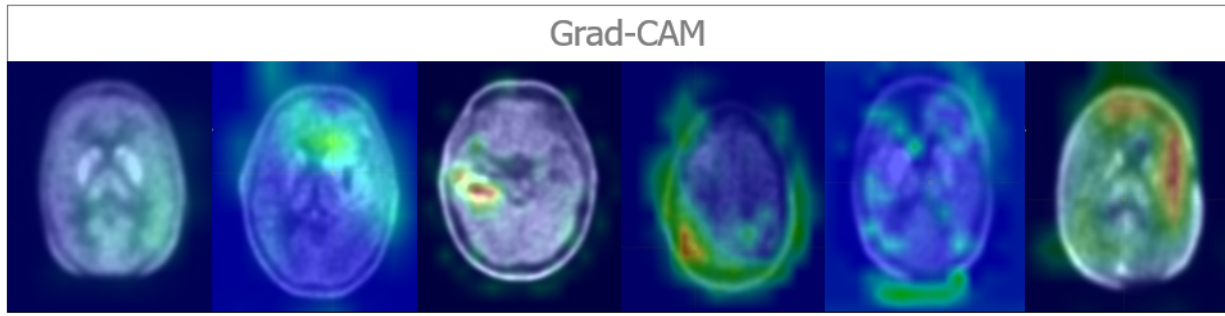


FIGURE 4.4 – Grad-CAM

### Limites

Malgré les améliorations, plusieurs limitations subsistent :

- *Taille de l'Échantillon* : La taille limitée de l'échantillon reste une contrainte majeure. Un plus grand nombre d'échantillons pour l'entraînement est nécessaire afin d'améliorer la robustesse du modèle et sa capacité à généraliser les résultats.

- *Exactitude des étiquetages* : Un autre facteur influençant la performance de notre modèle est l'exactitude des étiquetages des données. Les étiquetages ont été réalisés par trois médecins différents à différentes époques, ce qui peut introduire une variabilité et une incertitude dans les annotations. Cette variation dans le processus d'étiquetage pourrait affecter la qualité et la cohérence des données d'entraînement, contribuant ainsi aux défis observés dans la généralisation du modèle.

Dans cette optique, nous avons évalué l'évolution de la perte (loss) patient par patient au cours de l'entraînement. Cette analyse a révélé qu'un certains nombre de cas avait un impact disproportionné sur l'augmentation de la perte globale, suggérant la présence de cas potentiellement mal étiquetés.

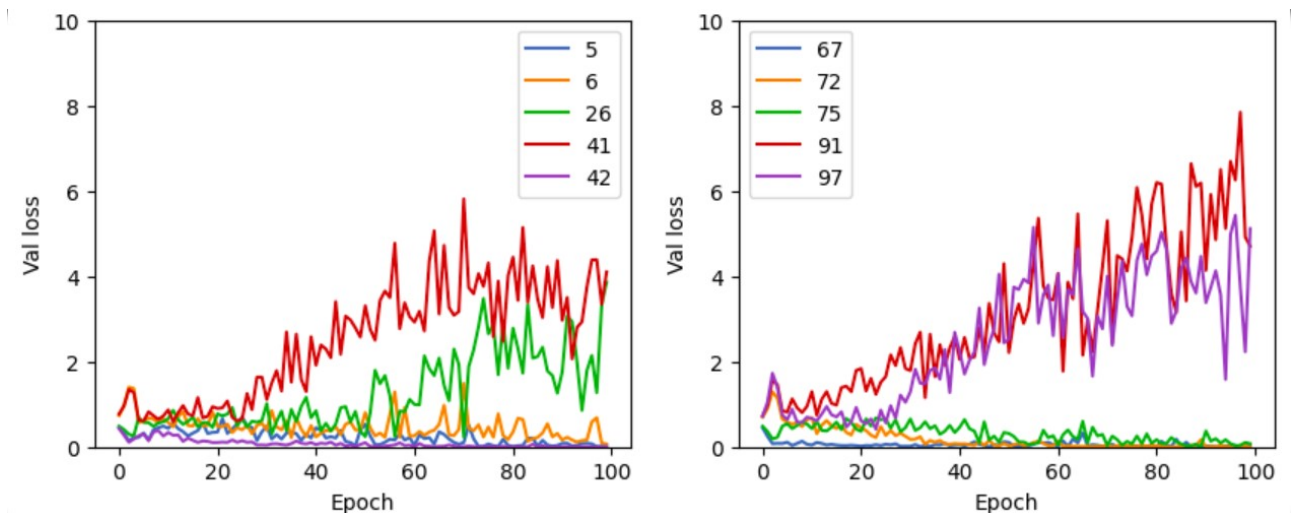


FIGURE 4.5 – Illustration de l'évolution de la perte de validation de quelques patients au fil des epochs

- *Généralisation des Modèles* : Un des défis majeurs de cette étude est la capacité du modèle à généraliser sur des images provenant d'autres centres et qui présenteraient donc des caractéristiques différentes en termes de bruit et de résolution spatiale. Notre base de données d'entraînement provient de deux machines différentes : Biograph et Vereos. Cette diversité dans

les équipements, associée aux différences dans les paramètres de reconstruction des images, crée une variabilité importante qui peut compliquer l'apprentissage. Toutefois, cette variabilité est bénéfique pour la généralisation du modèle, car elle l'entraîne à être plus robuste face à des données hétérogènes.

Cependant, la variabilité entre les données entre les deux machines de la base de données de Nancy est inférieure à celle d'une base de données externe telle que celle de Paris. De ce fait, la généralisation sur des données externes semble difficile avec un nombre limité de paramètres. Il est possible que des techniques de data augmentation plus adaptées et représentatives des différences observées entre les images issues de différents centres pourraient améliorer la capacité de généralisation du modèle.

– *Interprétabilité des Modèles* : Malgré les précisions élevées obtenues, la compréhension des processus décisionnels de nos modèles d'apprentissage automatique reste un défi majeur. Cette complexité peut limiter la confiance des cliniciens et freiner l'adoption en routine clinique. Nous avons utilisé Grad-CAM pour visualiser les régions d'images influençant les décisions du modèle, améliorant ainsi partiellement l'interprétabilité. Cependant, une compréhension complète des prédictions demeure difficile. L'intégration de techniques d'interprétabilité plus avancées, telles que SHAP (LUNDBERG et AL. 2017), Visual Saliency Maps (REYES et al. 2020) etc... pourrait être nécessaire pour éclairer davantage le processus décisionnel et renforcer la confiance dans nos modèles.

### **Choix du meilleur Modèle**

Malgré les défis et les limitations rencontrés, SENet se distingue comme le meilleur modèle globalement. Il présente une précision de 0.84 et une AUC de 0.82 sur l'ensemble de test, surpassant la plupart des autres modèles. De plus, SENet a la meilleure balanced accuracy de 0.72, ce qui indique une gestion efficace des classes déséquilibrées. Les intervalles de confiance pour SENet sont également les plus étroits, ce qui témoigne de la stabilité et de la fiabilité de ses performances. Les IC pour SENet montrent que ses métriques de performance sont moins sujettes à la variabilité, renforçant sa position comme le modèle le plus robuste et le plus fiable dans cette évaluation. Les résultats suggèrent que SENet est le choix optimal pour la généralisation et la gestion des déséquilibres de classe, consolidant ainsi sa position de modèle supérieur dans notre analyse.

# Chapitre 5

## Conclusion

Cette étude a exploré l'application des réseaux de neurones convolutifs (CNN) pour la prédiction des récurrences de gliomes à partir des images TEP  $^{18}\text{F}$ -FDOPA, visant à améliorer la précision diagnostique et la gestion des récurrences tumorales. Nous avons comparé plusieurs modèles de deep learning, dont VGG11, ResNet, SENet et un modèle de base CNN3D, afin de déterminer lequel offrait les meilleures performances.

Les résultats ont montré que les modèles CNNs ont un potentiel significatif pour l'analyse des images TEP dans la détection des récurrences de gliomes. Toutefois, plusieurs défis ont été rencontrés, notamment le surapprentissage, les limitations de taille d'échantillon, et la variabilité dans les étiquetages des données. Ces défis ont impacté la capacité du modèle à généraliser et ont mis en lumière la nécessité d'une base de données plus étendue et mieux annotée pour améliorer la performance du modèle.

Malgré ces défis, le modèle SENet s'est distingué comme le modèle le plus performant dans notre étude. Avec une précision de 0.84 et une AUC de 0.82, SENet a surpassé les autres modèles en termes de gestion des classes déséquilibrées et de stabilité des performances. Sa capacité à maintenir des intervalles de confiance étroits indique une fiabilité accrue et une robustesse dans la prédiction des récurrences tumorales. Ces caractéristiques font de SENet le choix optimal pour la généralisation sur de nouvelles données et pour l'application clinique potentielle.

Les résultats de cette étude soulignent l'importance des techniques d'apprentissage profond pour l'amélioration des diagnostics en oncologie, en particulier dans le contexte des gliomes où la distinction entre récurrence tumorale et effets du traitement est cruciale. La mise en œuvre de techniques d'interprétabilité, telles que Grad-CAM, a fourni des aperçus précieux sur les zones d'intérêt du modèle, bien que des méthodes supplémentaires comme LIME ou SHAP puissent être nécessaires pour une compréhension plus complète.

En conclusion, cette recherche ouvre la voie à des améliorations continues dans l'application des modèles de deep learning pour l'analyse d'images médicales. Les prochaines étapes pourraient inclure l'expansion de la base de données, et l'intégration de techniques d'interprétabilité plus avancées pour renforcer encore la précision et la fiabilité des modèles. L'adoption de ces approches pourrait conduire à des avancées significatives dans la gestion des gliomes et à une amélioration des résultats cliniques pour les patients.

# Bibliographie

- [1] QT OSTROM et al. « CBTRUS Statistical Report : Primary Brain and Central Nervous System Tumors Diagnosed in the United States in 2008-2012 ». In : *Neuro-Oncology* 17.Suppl 4 (2015), p. iv1-62.
- [2] KR LAMBORN et al. « Progression-free survival : An important end point in evaluating therapy for recurrent high-grade gliomas ». In : *Neuro-Oncology* 10 (2008), p. 162-170.
- [3] Nathalie L. ALBERT et al. « Response Assessment in Neuro-Oncology working group and European Association for Neuro-Oncology recommendations for the clinical use of PET imaging in gliomas ». In : *Neuro-Oncology* 18.9 (2016), p. 1199-1208. DOI : [10.1093/neuonc/now258](https://doi.org/10.1093/neuonc/now258).
- [4] I LAW et al. « Joint EANM/EANO/RANO practice guidelines/SNMMI procedure standards for imaging of gliomas using PET with radiolabelled amino acids and [18F]FDG : version 1.0 ». In : *European Journal of Nuclear Medicine and Molecular Imaging* 46 (2019), p. 540-557.
- [5] L. ROZENBLUM et al. « Differentiating high-grade glioma progression from treatment-related changes with dynamic [18F] FDOPA PET : a multicentric study ». In : *European Radiology* 33.4 (2023), p. 2548-2560.
- [6] Y PENG et al. « Deep learning and machine learning predictive models for neurological function after interventional embolization of intracranial aneurysms ». In : *Frontiers in Neurology* 15 (2024), p. 1321923. DOI : [10.3389/fneur.2024.1321923](https://doi.org/10.3389/fneur.2024.1321923).
- [7] LA SHEPP et Y VARDI. « Maximum likelihood reconstruction for emission tomography ». In : *IEEE Transactions on Medical Imaging* 1.2 (1982), p. 113-122.
- [8] H. Malcolm HUDSON et Richard S. LARKIN. « Accelerated image reconstruction using ordered subsets of projection data ». In : *IEEE Transactions on Medical Imaging* 13.4 (1994), p. 601-609.
- [9] T. ZARAGORI. « Integration of an optimized dynamic and radiomics analyses in 18F-FDOPA PET for the characterization of gliomas in clinical routine practice ». Thèse de doct. Université de Lorraine, 2021.
- [10] T. ZARAGORI et al. « Use of static and dynamic [18 F]-F-DOPA PET parameters for detecting patients with glioma recurrence or progression ». In : *EJNMMI Research* 10 (2020), p. 1-10.
- [11] S. AHRARI et al. « Relevance of Dynamic [18F]-DOPA PET Radiomics for Differentiation of High-Grade Glioma Progression from Treatment-Related Changes ». In : *Biomedicines* 9.12 (déc. 2021), p. 1924. DOI : [10.3390/biomedicines9121924](https://doi.org/10.3390/biomedicines9121924).
- [12] K SIMONYAN et A ZISSERMAN. *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv :1409.1556. 2014.
- [13] Kaiming HE et al. « Deep Residual Learning for Image Recognition ». In : *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [arXiv :1512.03385]. Las Vegas, NV, USA, juin 2016, p. 770-778.



- [14] Jiancheng YANG et al. « Reinventing 2D Convolutions for 3D Images ». In : *IEEE Journal of Biomedical and Health Informatics* 25.8 (2021), p. 3009-3018.
- [15] Jie HU, Li SHEN et Gang SUN. « Squeeze-and-Excitation Networks ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [16] Forrest N IANDOLA. *SqueezeNet : AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5 MB Model Size*. arXiv preprint arXiv :1602.07360. 2016.
- [17] Yisen WANG et et AL. « Symmetric Cross Entropy for Robust Learning with Noisy Labels ». In : *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.
- [18] Dmitry ULYANOV et et AL. « Instance Normalization : The Missing Ingredient for Fast Stylization ». In : *arXiv e-prints* (2016). arXiv : [arXiv:1607.08022 \[cs.CV\]](https://arxiv.org/abs/1607.08022).
- [19] Karol GOTKOWSKI et et AL. *M3d-CAM : A PyTorch Library to Generate 3D Data Attention Maps for Medical Deep Learning*. arXiv preprint arXiv :2007.00453. 2020.
- [20] Arnaud ZWANENBURG et et AL. « Image biomarker standardisation initiative ». In : *arXiv preprint arXiv :1612.07003* (2016).
- [21] Fabian ISENSEE et al. *batchgenerators - a python framework for data augmentation*. Zenodo. 2020. DOI : [10.5281/zenodo.3632567](https://doi.org/10.5281/zenodo.3632567).
- [22] J DENG et al. « Imagenet : A Large-Scale Hierarchical Image Database ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, p. 248-255.
- [23] SM LUNDBERG et et AL. « A Unified Approach to Interpreting Model Predictions ». In : *Advances in Neural Information Processing Systems (NeurIPS)*. T. 30. Curran Associates, Inc., 2017.
- [24] Mauricio REYES et al. « On the interpretability of artificial intelligence in radiology : challenges and opportunities ». In : *Radiology : Artificial Intelligence* 2.3 (2020), e190043. DOI : [10.1148/ryai.2020190043](https://doi.org/10.1148/ryai.2020190043).