Analyse des pathologies neuro-dégénératives par apprentissage profond

Cécilia Ostertag

LA ROCHELLE UNIVERSITÉ

ÉCOLE DOCTORALE 618

LABORATOIRE : INFORMATIQUE, IMAGE, ET INTERACTION (L3i)

THÈSE

présentée par :

Cécilia Ostertag

soutenue le : **05 janvier 2022**

pour obtenir le grade de : **Docteur de La Rochelle Université**

Discipline : **Informatique et Applications**

Analyse des pathologies neuro-dégénératives par apprentissage profond

---

JURY :

Hélène AMIEVA	Professeur	Université de Bordeaux	Examinatrice
Karell BERTET	Maître de conférence/HDR	La Rochelle Université	Directrice
Marie BEURTON-AIMAR	Maître de conférence/HDR	Université de Bordeaux	Directrice
Benoît NAEGEL	Professeur	Université de Strasbourg	Rapporteur
Su RUAN	Professeur	Université de Rouen	Rapportrice
Thierry URRUTY	Maître de conférence/HDR	Université de Poitiers	Encadrant
Muriel VISANI	Maître de conférence/HDR	La Rochelle Université	Directrice
Akka ZEMMARI	Professeur	Université de Bordeaux	Président du jury

Table des matières

Résumé

Le suivi et l'établissement de pronostiques sur l'état cognitif des personnes affectées par une maladie neurologique sont cruciaux, car ils permettent de fournir un traitement approprié à chaque patient, et cela le plus tôt possible. Ces patients sont donc suivis régulièrement pendant plusieurs années, dans le cadre d'études longitudinales. A chaque visite médicale, une grande quantité de données est acquise : présence de facteurs de risque associés à la maladie, imagerie médicale (IRM ou PET-scan), résultats de tests cognitifs, prélèvements de molécules identifiées comme bio-marqueurs de la maladie, *etc.* Ces différentes modalités apportent des informations sur la progression de la maladie, certaines complémentaires et d'autres redondantes.

De nombreux modèles d'apprentissage profond ont été appliqués avec succès aux données bio-médicales, notamment pour des problématiques de segmentation d'organes ou de diagnostique de maladies. Ces travaux de thèse s'intéressent à la conception d'un modèle de type "réseau de neurones profond" pour la prédiction du déclin cognitif de patients à l'aide de données multimodales.

Ainsi, nous proposons une architecture composée de sous-modules adaptés à chaque modalité : réseau convolutif 3D pour les IRM de cerveau, et couches entièrement connectées pour les données cliniques quantitatives et qualitatives. Pour évaluer l'évolution du patient, ce modèle prend en entrée les données de deux visites médicales quelconques. Ces deux visites sont comparées grâce à une architecture siamoise. Après avoir entraîné et validé ce modèle en utilisant comme cas d'application la maladie d'Alzheimer, nous nous intéressons au transfert de connaissance avec d'autres maladies neuro-dégénératives, et nous utilisons avec succès le transfert d'apprentissage pour appliquer notre modèle dans le cas de la maladie de Parkinson. Enfin, nous discutons des choix que nous avons pris pour la prise en compte de l'aspect temporel du problème, aussi bien lors de la création de la vérité terrain en fonction de l'évolution au long terme d'un score cognitif, que pour le choix d'utiliser des paires de visites au lieu de plus longues séquences.

Abstract

Monitoring and predicting the cognitive state of a subject affected by a neuro-degenerative disorder is crucial to provide appropriate treatment as soon as possible. Thus, these patients are followed for several years, as part of longitudinal medical studies. During each visit, a large quantity of data is acquired: risk factors linked to the pathology, medical imagery (MRI or PET scans for example), cognitive tests results, sampling of molecules that have been identified as bio-markers, *etc.* These various modalities give information about the disease's progression, some of them are complementary and others can be redundant.

Several deep leraning models have been applied to bio-medical data, notably for organ segmentation or pathology diagnosis. This PhD is focused on the conception of a deep neural network model for cognitive decline pediction, using multimodal data, here both structural brain MRI images and clinical data.

In this thesis we propose an architecture made of sub-modules tailored to each modality: 3D convolutional network for the brain MRI, and fully connected layers for the quantitative and qualitative clinical data. To predict the patient's evolution, this model takes as input data from two medical visits for each patient. These visits are compared using a siamese architecture. After training and validating this model with Alzheimer's disease as our use case, we look into knowledge transfer to other neuro-degenerative pathologies, and we use transfer learning to adapt our model to Parkinson's disease. Finally, we discuss the choices we made to take into account the temporal aspect of our problem, both during the ground truth creation using the long-term evolution of a cognitive score, and for the choice of using pairs of visits as input instead of longer sequences.

Remerciements

Je souhaiterais commencer par remercier mes directrices de thèse, *Marie Beurton*, *Muriel Visani* et *Karell Bertet*, ainsi que mon co-encadrant *Thierry Urruty*, pour m'avoir fait confiance en me choisissant pour mener cette thèse. Je remercie en particulier *Muriel* et *Thierry* pour m'avoir encadré à distance, et toujours fourni des remarques et idées constructives, que ce soit lors de mes expérimentations, ou lors de la rédaction d'articles et de ce manuscrit. Je remercie encore plus en particulier Marie pour son soutien non seulement professionnel, mais amical, le long de ces trois ans qui n'ont pas été de tout repos. Enfin, je vous remercie tous les trois pour m'avoir soutenu sans sourciller sur des sujets plus personnels.

Je voudrais également adresser mes remerciements aux membres de mon jury de thèse, *Hélène Amieva*, *Akka Zemmari*, et en particulier mes deux rapporteurs *Benoît Naegel* et *Su Ruan*, pour leur intérêt et pour avoir accepté d'évaluer mes travaux.

Je remercie également mon collègue doctorant *Antoine Pirronne*, ainsi que l'égyptologue *Sandra Lippert*, qui m'ont fait m'intéresser au sujet de l'informatique pour l'héritage culturel, et qui ont été le point de départ de tout un projet annexe réalisé pendant les périodes creuses de ces années de thèse.

Bien sûr, je remercie mes ami.e.s et collègues de bureau : *Linh*, *Myriam*, *Gala*, *Claire*, *Trang*, *Tú*, et *Kévin*, ainsi que quelques membres de notre GT deep learning : *Charlotte*, *Joris*, et *Benoît*, pour leur soutien, leur intérêt pour mon sujet de thèse, et leurs questions, compliments, et remarques constructives lors de mes présentations. Je remercie également les membres de ma famille, qui ne comprendront jamais vraiment mon sujet de thèse, mais qui ont bien compris toute l'importance que celle-ci avait pour moi.

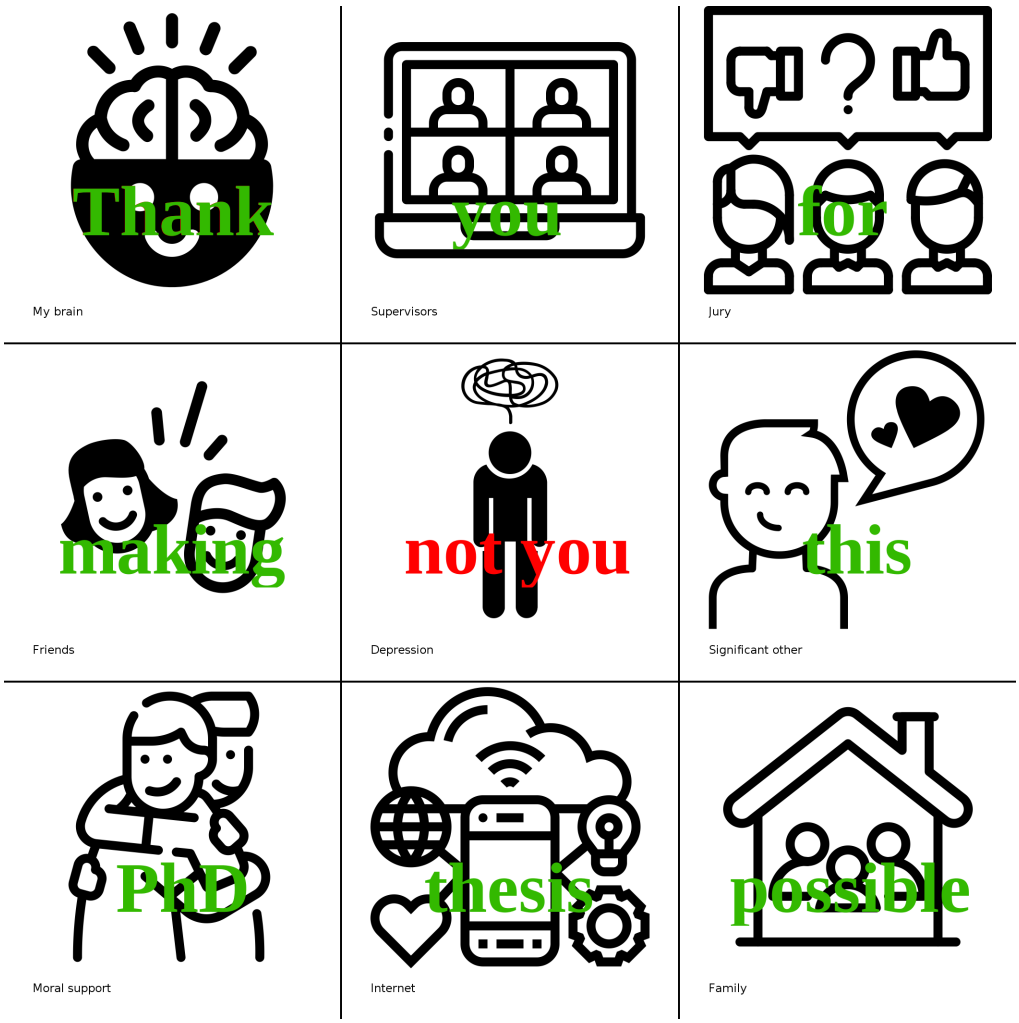
Enfin, je remercie *François*, mon compagnon, pour avoir partagé chaque joie et chaque doute durant toutes ces années, et pour avoir eu le courage de relire ce manuscrit plusieurs fois, avec toujours autant de patience.

J'ai beaucoup changé au cours de ces trois dernières années, et vous avez tous influencé d'une manière ou d'une autre ces changements. Pour cela, je vous remercie du fond du coeur.

Cécil/Cécilia Ostertag

Dédicace

Je dédie cette thèse à mon moi du passé, qui rêvait un jour d'être chercheur et d'écrire des articles scientifiques. Merci d'y avoir cru, ça valait vraiment le coup :)



Introduction

Contexte et Motivation

Les **maladies neuro-dégénératives**, comme la maladie de Parkinson ou la maladie d'Alzheimer, sont caractérisées par une destruction progressive des cellules nerveuses dans diverses zones du cerveau, due à l'agrégation de molécules indésirables [1]. Cette dégénérescence neuronale conduit alors à des troubles d'ordre cognitif, comportementaux, et / ou moteurs. D'après Santé Publique France, en 2019 *« Plus d'un million de personnes [étaient] touchées par la maladie d'Alzheimer et autres démences, [et] environ 160 000 personnes [étaient] traitées pour la maladie de Parkinson »* [2]. De plus, étant donné que la population française est vieillissante, de plus en plus de personnes sont touchées par ces maladies neuro-dégénératives. Ces pathologies constituent donc un problème de santé publique majeur en France.

Le but de ces travaux de thèse est d'utiliser les informations issues des **multiples modalités** disponibles pour les sujets suivis pour ce type de pathologies, afin de **détecter de façon précoce le déclin cognitif**. Ainsi, cela permettrait d'identifier les sujets les plus à risques pour leur proposer un traitement adapté. Ces informations sont d'ordre morphologique, grâce à l'imagerie médicale (**IRM de cerveau**), ou d'ordre clinique (**facteurs de risques** et résultats à des **tests neuro-cognitifs**), et sont toutes utiles pour caractériser l'état de santé de ces patients. Nous cherchons ici à développer une méthode pour comparer l'état de santé d'un même sujet entre deux visites médicales, afin d'**étudier indépendamment l'évolution de la maladie chez chaque sujet**.

Les études les plus courantes sur les maladies neuro-dégénératives intègrent rarement l'analyse détaillée des images médicales brutes, mais reposent en général sur un pré-traitement des images, potentiellement source d'erreurs, comprenant une segmentation du cerveau en zones d'intérêt suivant des atlas de cerveau humain. Ce pré-traitement permet l'obtention de vecteurs de caractéristiques, appelées marqueurs radiomiques, qui permettent de décrire les régions d'intérêt, par exemple la forme d'une tumeur ou la texture d'un tissu [3].

Depuis une dizaine d'années, les méthodes d'**apprentissage profond** (*deep learning*), basées sur les **réseaux de neurones artificiels**, ont gagné en popularité. Le but de ces méthodes est d'apprendre automatiquement, en minimisant une fonction d'erreur, les meilleurs paramètres pour le calcul de ces descripteurs [4]. Les modèles *deep learning* sont

également utilisés pour calculer des représentations plus informatives à partir de données non-images, de type données cliniques, sémantiques, ou textuelles. De tels réseaux ont aussi la capacité de traiter simultanément des données provenant de modalités différentes. Ainsi, l'originalité de ces travaux de thèse sera l'utilisation de méthodes *deep learning*, d'une part pour **analyser directement les images médicales brutes** (donc en 3D volumique), et d'autre part pour **intégrer dans un seul modèle l'ensemble des modalités** de données disponibles pour les patients suivis.

Les travaux de cette thèse s'inscrivent dans le projet région Nouvelle Aquitaine ICE / ADDICTO (RCB 2018-A02699-46). Dans ce mémoire, nous présenterons notre modèle, "Multimodal3DSiameseNet", de type réseau de neurones profond, conçu et implémenté pour répondre à la problématique décrite ci-dessus. Ce modèle possède une **architecture dite "siamoise"**, qui nous permet de **comparer les données de deux visites médicales d'un même sujet**. Notre modèle "Multimodal3DSiameseNet" traite les IRM de cerveau en tant qu'**images 3D volumiques**, afin d'en extraire les informations spatiales nécessaires pour caractériser la morphologie du cerveau des sujets. Conjointement à cela, ce modèle traite des **vecteurs de données cliniques**. Cette **multimodalité** est rendue possible par la définition de **sous-modules dédiés**, et adaptés aux spécificités de chaque modalité.

Les résultats de ces travaux ont été obtenus en utilisant une base de données internationale, nommée ADNI [5], spécifique à l'étude de la maladie d'Alzheimer. Nous avons également montré que notre modèle est **transférable** à une autre pathologie, en utilisant la base de données internationale PPMI [6], dont les participants sont suivis pour l'étude de la maladie de Parkinson.

Organisation de la thèse

Dans le **Chapitre 1**, nous réalisons un état de l'art sur les différentes architectures de réseaux de neurones profonds existants, et leur application aux différentes modalités de données que nous avons traitées : données quantitatives, images 2D, images 3D, et séries temporelles, ainsi qu'aux différentes tâches d'apprentissage pertinentes dans cette thèse : classification, comparaison, et transfert d'apprentissage.

Dans le **Chapitre 2**, nous détaillons les applications bio-médicales des réseaux de neurones profonds, avec une attention particulière sur les modèles multimodaux et le transfert d'apprentissage. Cet état de l'art nous a permis d'identifier les difficultés liées à l'utilisation de données médicales, ainsi que les principales forces et faiblesses des modèles existants. Il a aussi mis en évidence le manque de travaux pré-existants portant sur l'utilisation des réseaux de neurones pour l'apprentissage multimodal (autres que l'utilisation combinée de plusieurs modalités d'imagerie), ainsi que le manque de modèles conçus pour donner un pronostique sur l'évolution des maladies au lieu d'un simple diagnostique ponctuel.

Dans le **Chapitre 3**, nous présentons le cas d'application principal que nous avons choisi pour développer notre modèle. Il s'agit de la maladie d'Alzheimer, maladie neurologique très étudiée de nos jours, et pour laquelle il existe de nombreuses bases de données accessibles et de tailles suffisantes pour entraîner et tester nos modèles d'apprentissage profond. En utilisant les données de la base de données ADNI, nous avons créé une vérité terrain pour les sujets de cette base, afin de les catégoriser en deux groupes, selon la présence ou non d'un déclin cognitif sur plusieurs années.

Nous avons ensuite progressivement conçu et implémenté notre architecture multimodale, "Multimodal3DSiameseNet". Cette architecture est un réseau de neurones profond basé sur deux principes. Premièrement, une division en modules différents (un par type de modalité), avec des opérations adaptées à chaque modalité. Deuxièmement une architecture en branches parallèles (pour chaque modalité avec une évolution dans le temps), permettant la comparaison de paires de visites médicales pour obtenir des informations sur l'évolution de l'état de santé des sujets dans un laps de temps donné.

Dans le **Chapitre 4**, nous présentons des résultats supplémentaires, en utilisant un jeu de données plus large, issu de la même base de données (ADNI). Grâce à ces nouveaux résultats, nous avons montré que notre modèle est capable de prédire l'évolution au long-terme d'un sujet, à partir de deux visites médicales précoces, et sans contraintes sur la durée de l'intervalle entre les deux visites choisies. Nous fournissons ensuite une preuve de concept de l'utilisation de notre architecture, ainsi que des paramètres appris par notre modèle sur la maladie d'Alzheimer, pour des pathologies similaires, avec comme cas d'application le transfert d'apprentissage vers la maladie de Parkinson. Pour cela, nous avons utilisé la base de données PPMI.

Enfin, dans le **Chapitre 5**, nous comparons nos résultats avec une autre stratégie de l'état de l'art, basée sur l'approche des réseaux récurrents, très utilisés pour les données temporelles. Nous questionnons notre choix de protocole pour la création de la vérité terrain, et étudions l'importance de ce choix pour la phase d'entraînement et la phase d'inférence avec notre modèle. Nous terminons en rappelant les contributions de notre modèle, ainsi que ses limites et quelques perspectives d'amélioration que nous avons pu identifier.

A noter que nous présentons en **Annexe C** une application hors contexte médical des réseaux siamois, pour le problème de la reconstruction de documents anciens. Ces travaux ont été réalisés en parallèle avec le projet sur les maladies neuro-dégénératives, et ont abouti sur une publication dans le journal *Pattern Recognition Letters*, et une publication dans un *workshop* de la conférence internationale *ICPR 2020*.

Première partie

Etat de l'art des réseaux de neurones et leur application aux données biomédicales

Chapitre 1

Réseaux de neurones profonds, stratégies d'optimisation, et évaluation

1.1 Fonctionnement des réseaux de neurones profonds

1.1.1 Généralités

Les premiers réseaux de neurones artificiels ont été proposés dans les années 1970, mais leur utilité était limitée à cette époque par plusieurs contraintes techniques, comme la puissance de calcul et l'espace mémoire disponible sur les machines. A partir des années 80, et surtout des années 2000, l'architecture des ordinateurs ayant profondément évolué, avec entre autre l'apparition des GPUs, cette discipline a connu un renouveau. En 1998, le succès des réseaux de neurones pour la classification d'images a été démontré par LeCun, avec un réseau à 7 couches nommé LeNet [7]. Ce sont des architectures composées d'un ensemble de cellules (neurones artificiels), réunies en couches successives, qui permettent l'extraction automatique de descripteurs (*features*) dans des données. Lorsque ces réseaux possèdent plus d'une couche cachée (*hidden layer*), on parle de réseaux de neurones profonds (*Deep Neural Network* (DNN)). Ce sont ces types de réseaux que nous utiliserons dans ces travaux de thèse.

Dans un réseau, l'unité de base est le neurone artificiel (Figure 1.1). La valeur de sortie du neurone j est donnée par la formule

$$o_j = \varphi\left(\sum_{i=0}^{n} w_{ij} x_i\right)$$

où n est le nombre total de variables (attributs) d'entrées, x_i est la i^{eme} entrée, et w_{ij} est le poids associé à la i^{eme} entrée du neurone j. φ est la fonction d'activation associée au neurone.

Dans les réseaux de neurones profonds dédiés à des tâches de classification, les fonctions softmax (fonction exponentielle normalisée) ou sigmoïde sont souvent utilisées pour la couche de sortie. Ces deux fonctions transforment leur entrée (vecteur de valeurs non

normalisées) en un vecteur de valeurs comprises entre 0 et 1, et dont la somme fait 1, que l'on peut ensuite considérer comme les probabilités d'appartenance à chaque classe.

La *Rectified Linear Unit (ReLU)*, *i. e.* $min(x, 0)$ avec x une valeur d'entrée, est une fonction d'activation utilisée en particulier après les couches cachées des réseaux de neurones profonds. C'est une fonction très simple, qui permet d'introduire de la non-linéarité, et permet au réseau de résoudre des problèmes complexes avec relativement peu de neurones. Cependant, l'utilisation de la ReLU peut conduire à l'obtention de neurones dont la sortie devient invariablement 0 ("*dying ReLU problem*"). Pour pallier à cela, il est possible d'utiliser une *Leaky ReLU*, pour laquelle on a $x = 0.01x$ si $x < 0$ [8].

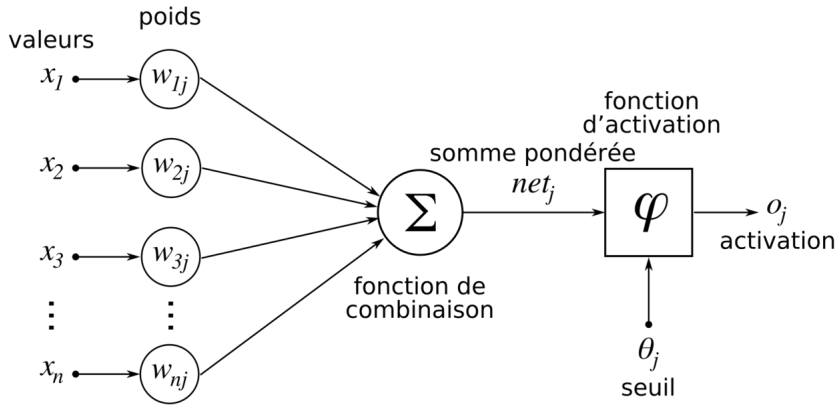


FIGURE 1.1 – Représentation schématique d'un neurone artificiel (Source : Wikipédia, auteur : Chrislb, licence GFDL)

L'utilisation de ces modèles est basée sur une phase d'entraînement, durant laquelle les paramètres du modèle sont ajustés de façon automatique pour répondre à une tâche donnée.

1.1.2 Entraînement

L'entraînement des réseaux *feed forward* classiques est basé sur la méthode de rétropropagation du gradient (*backpropagation*). Une fonction de coût est calculée pour quantifier l'erreur commise par le modèle à chaque itération. L'erreur est calculée entre le résultat donné par le modèle et la vérité terrain (résultat à prédire). La phase d'entraînement vise à minimiser le gradient de cette fonction de coût en ajustant les paramètres du réseau, afin d'améliorer progressivement ses performances. Enfin, une fois le modèle entraîné, il pourra être utilisé sur de nouvelles données, il pourra être utilisé sur de nouvelles données avec ses paramètres ayant été ajustés lors de l'apprentissage.

Dans le cas des réseaux profonds, le nombre de paramètres à ajuster étant potentiellement très grand, cet entraînement repose sur l'utilisation de grands jeux de données. Lorsque le jeu de données est de petite taille, il est possible de l'augmenter artificiellement, en simulant de nouvelles données à partir des données originales. Dans le cas du

traitement d'images, de nombreuses stratégies d'augmentation de données existent (rotation, translation, zoom, etc.) [9], mais ces nouvelles images doivent avoir un sens par rapport au type de données traitées. Ainsi, les stratégies d'augmentation utilisées sont conditionnées par le type d'image d'entrée et la tâche à accomplir. Par exemple, pour la classification d'images naturelles, si l'on veut que le modèle soit invariant à la rotation, on pourra augmenter le jeu de données en faisant aléatoirement des rotations sur les images. Il est également possible de générer des images artificiellement, en utilisant par exemple les réseaux antagonistes génératifs [10] (voir Section 1.2.4).

1.1.3 Exemples de fonctions de coût

Comme nous l'avons expliqué précédemment, les fonctions de coût (*loss functions*) sont utilisées durant l'entraînement, afin d'évaluer les erreurs commises par le modèle. Pour des tâches de classification supervisées, les fonctions les plus couramment utilisées sont les suivantes :

Mean Squared Error loss : $MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$
avec n : nombre d'éléments, y_i : classe de l'élément i,
et $\hat{y}_i$: classe prédite pour l'élément i

Mean Averaged Error loss : $MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|^2}{n}$

Hinge loss : $loss = \sum_{j \neq y_i} max\left(0, s_j - s_{y_i} + 1\right)$
avec s_j la probabilité associée à la classe j

Cross-entropy loss : $loss = -\left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\right)$
pour chaque élément i

Contrastive loss : $loss = y_i d(f(x_1), f(x_2)) + (1 - y_i) max(0, m - d(f(x_1), f(x_2))$
pour chaque paire i, avec les deux éléments x_1 et x_2 de la paire, $y_i = 0$ si la paire est négative (classes différentes), et 1 si la paire est positive (mêmes classes), m la marge entre paires positives et négatives, f le réseau de neurones, et d une fonction de distance

Triplet loss : $loss = max(0, m + d(f(x_a), f(x_p)) - d(f(x_a), f(x_n))$
pour chaque triplet i avec x_a un élément (*anchor*), x_p un échantillon positif respectivement à x_a, et x_n un échantillon négatif respectivement à x_a

1.1.4 Limites

Un des problèmes particulièrement courants avec les réseaux de neurones profonds est le sur-apprentissage (*overfitting*), c'est-à-dire l'obtention d'un modèle incapable de généralisation et incapable de faire des prédictions correctes pour de nouvelles données. Cela est dû au grand nombre de paramètres de ce type de modèle au regard du volume de données d'apprentissage disponibles. Pour prévenir ce problème, en plus de l'augmentation de

données évoquée précédemment, un jeu de données distinct appelé jeu de validation est utilisé en même temps que le jeu d'entraînement, mais ne sert pas à ajuster les poids du réseau. Ainsi, si durant l'entraînement la fonction de coût de l'entraînement diminue, alors que la fonction de coût de la validation augmente, cela signifie qu'il y a sur-apprentissage (voir Figure 1.2). L'objectif est alors de réduire ce sur-apprentissage en ajustant certains paramètres, par exemple la vitesse d'apprentissage (*learning rate*), ou en ajoutant des couches où certains neurones sont mis à zéro (*dropout*) afin d'introduire de la variabilité dans les données.

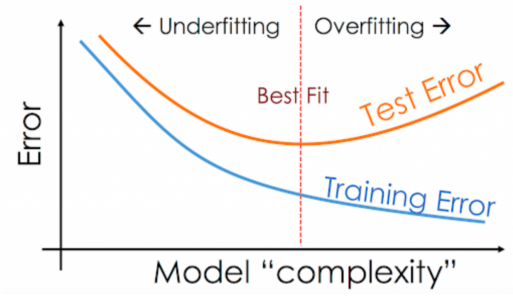


FIGURE 1.2 – Visualisation du sur-apprentissage d'un modèle à l'aide la fonction de coût d'entraînement et de validation/test (Source : [11]

Dans la suite de ce chapitre, nous allons présenter quelques architectures courantes de réseaux de neurones profonds, dont celles que nous avons utilisées dans ces travaux.

1.2 Architectures courantes de réseaux de neurones profonds

1.2.1 Réseaux de neurones convolutifs

Pour le traitement de données à deux ou trois dimensions, comme les images, une variante de ces architectures est utilisée, appelée réseaux de neurones convolutifs (*Convolutional Neural Networks* (CNN)) (voir Figure 1.3). Souvent, ces réseaux sont constitués d'une suite de couches de convolution suivies de réduction spatiale (*pooling*), grâce auxquelles les informations caractéristiques des images sont extraites dans un espace de plus faible dimension que l'image d'origine. A la fin du réseau, un classifieur constitué de couches dites "entièrement connectées" (*fully connected*) utilise ces informations pour la classification.

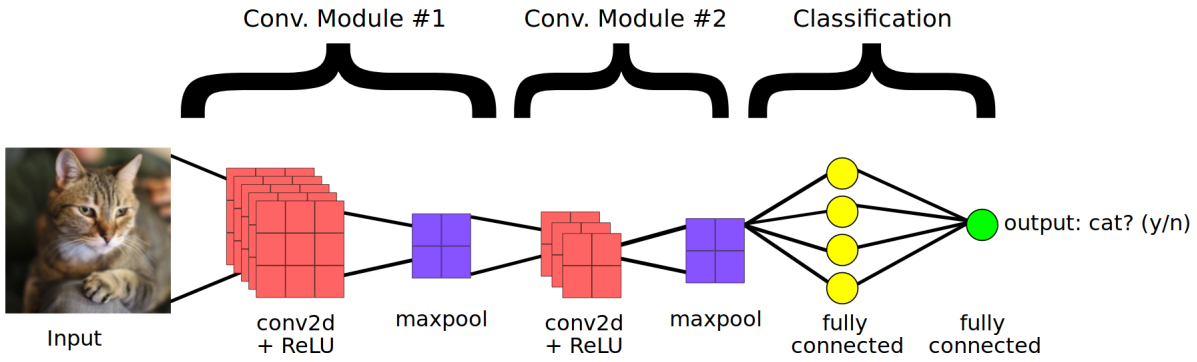


FIGURE 1.3 – Représentation simplifiée d'un réseau convolutif (CNN) [12]

Dans les couches de convolution et de *pooling*, chaque neurone correspond à un filtre (*kernel*) à 1, 2, ou 3 dimensions, selon le type d'opération souhaitée. Il faut également noter que pour les images couleur, le nombre de canaux (par exemple R,G,B) compte pour une dimension supplémentaire. Lorsqu'une image a plusieurs canaux (pas nécessairement des canaux correspondant à des couleurs), à chaque couche un résultat intermédiaire sera calculé indépendamment pour chaque canal, puis le résultat final sera donné par la moyenne des valeurs de chaque pixel sur l'ensemble des canaux.

Parmi les architectures CNN les plus connues, ont peut citer :

— GoogleInceptionV1 (GoogLeNet) [13], réseau de 22 couches, ayant marqué l'état de l'art pour la classification d'images naturelles (*challenge* ImageNet 2014).

— VGG-16 [14], qui a obtenu des performances supérieures à GoogLeNet, avec une *accuracy* de 92.7% sur le *challenge* ImageNet, mais dont l'inconvénient est le nombre élevé de paramètres (138 millions) et le temps de calcul nécessaire pour l'entraînement.

— ResNet [15], avec 152 couches, qui introduit le principe de *skip connections* (connections entre couches non successives)

— EfficientNet [16], qui est basé sur le principe de l'adaptation du nombre de paramètres, et de la taille et du nombre de couches du modèle en fonction des contraintes de ressources de calcul imposées par l'utilisateur.

Ces réseaux ont fait leurs preuves dans le domaine de l'apprentissage profond pour l'analyse d'images naturelles en 2D, mais ils ne sont pas utilisables dans notre contexte de données d'imagerie médicale en 3D volumique.

1.2.2 Réseaux de neurones siamois

Pour répondre au problème de la comparaison d'échantillons de données appariées, parmi les architectures les plus utilisées, on peut citer des réseaux de type siamois. Les réseaux siamois pour la comparaison d'échantillons de données appariées commencent par deux branches DNN identiques (mêmes opérations, même nombre de neurones), qui possèdent les mêmes poids depuis l'initialisation jusqu'à la fin de l'entraînement. Une fonction de distance ou de similarité est ensuite utilisée pour comparer les *features* des deux données d'entrée (Figure 1.4).

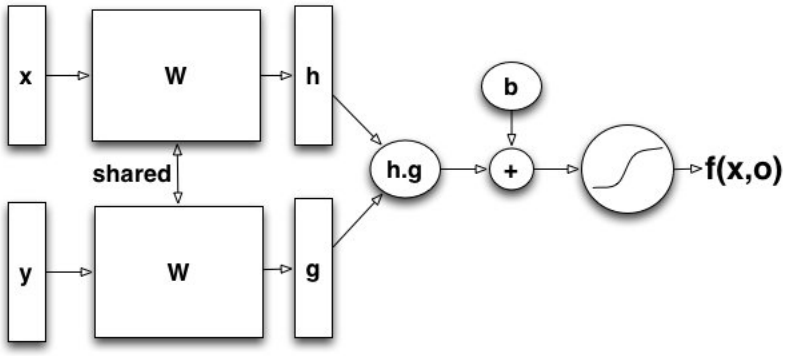


FIGURE 1.4 – Représentation simplifiée d'un réseau siamois [17]

Ces modèles sont ensuite entraînés à distinguer les paires similaires des paires dont les *features* diffèrent, en se basant sur ce score de distance. On retrouve ces réseaux pour traiter de nombreux problèmes, notamment la reconnaissance d'écriture [18], de reconnaissance d'images, [19], reconnaissance faciale [20], ou d'identification d'alphabet [21].

Dans ce contexte de comparaison de données appariées, des fonctions de coût spécifiques sont utilisées, la plus connue étant la *contrastive loss* [22] dont la formule est donnée plus haut. Cette fonction utilise un score de distance calculé entre les résultats des deux branches d'un réseau siamois, afin d'apprendre la différence entre des paires "négatives" et des paires "positives". Par exemple, dans le contexte de reconnaissance faciale, une paire positive sera constituée de deux images de la même personne, et une paire négative sera constituée d'images de deux personnes différentes.

Au cours de l'entraînement, le modèle est optimisé pour que les images des paires positives se retrouvent proches dans l'espace des *features*, et les images des paires négatives se retrouvent éloignées (voir Figure 1.5).

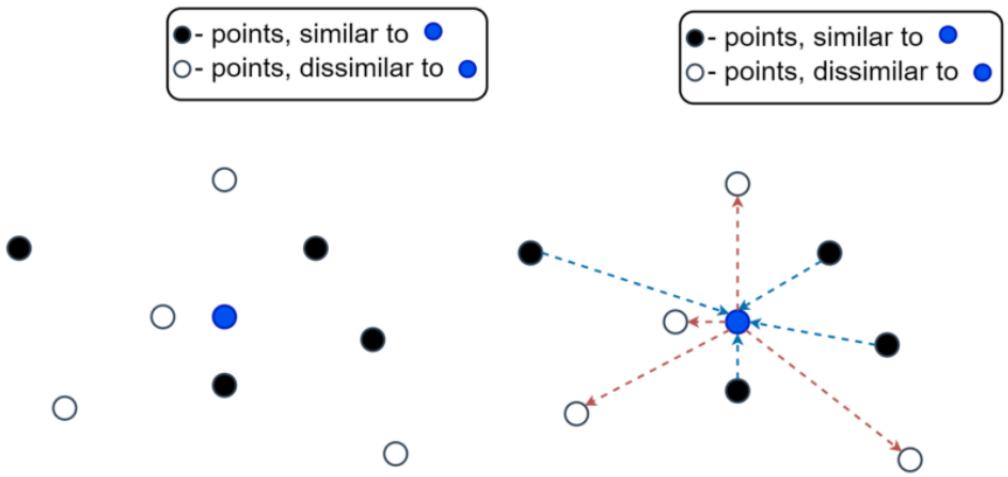


FIGURE 1.5 – Fonctionnement de la *contrastive loss* [23]

Les réseaux siamois peuvent également être utilisés pour l'apprentissage multi-tâches. Une partie du réseau permet d'extraire des *features* globales, tandis que d'autres parties du réseau sont spécifiques à chaque tâche (classification, segmentation, reconstruction, etc.). L'objectif est de contraindre le réseau de neurones à apprendre une représentation des données qui soit moins biaisée par le type de tâche et donc plus facilement généralisable [24].

L'architecture siamoise pour la comparaison d'échantillons de données appariées est celle que nous avons utilisée pour réaliser le travail présenté dans ce mémoire.

1.2.3 Auto-encodeurs et Encodeurs-décodeurs

Les auto-encodeurs sont des architectures conçues pour apprendre une représentation "compressée" des données d'entrée. Lors de la phase d'apprentissage, le réseau apprend à réduire la dimension des données (partie encodeur), puis à reconstituer les données originales à partir de cette représentation compressée (partie décodeur). Autrement dit, le réseau apprend une approximation de la fonction identité. L'avantage de ce type d'architecture est qu'elles sont entraînées de façon non supervisée, ce qui signifie qu'il n'est pas nécessaire d'utiliser des données labellisées.

Comme pour les réseaux siamois, les couches qui composent l'auto-encodeur peuvent être de type *fully-connected*, convolutives (exemple Figure 1.6), ou autres.

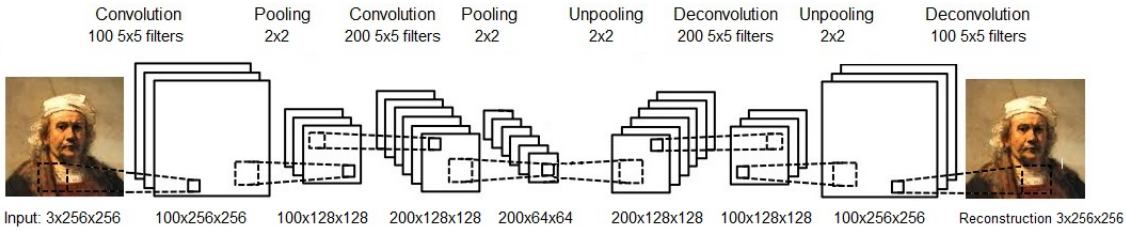


FIGURE 1.6 – Architecture de l'auto-encodeur convolutif utilisé par [25]

Les réseaux de type encodeurs-décodeurs sont également utilisés pour réaliser automatiquement des transformations sur les données d'entrée. Dans le cas d'analyse d'images, l'un des réseaux les plus populaires est U-Net [26], qui est utilisé notamment pour la segmentation sémantique.

1.2.4 Réseaux antagonistes génératifs

Les *Generative Adversarial Networks* (GAN) ont été présentés en 2014 par Goodfellow *et. al.* [27], puis l'architecture a été modifiée pour donner les *Deep Convolutive GAN* (DCGAN) en 2015 [28], adaptés à la génération d'images. Le principe de ces modèles est de générer des données artificielles qui suivent une certaine distribution (par exemple générer des images de visages, d'animaux, ou de pièces d'un maison).

L'optimisation de ces modèles peut se faire de façon directe, en utilisant une fonction de coût pour comparer la distribution des échantillons générés à celle des échantillons véritables, ou de façon indirecte, en utilisant un réseau discriminateur dont le but est de reconnaître les échantillons générés (voir Figure 1.7).

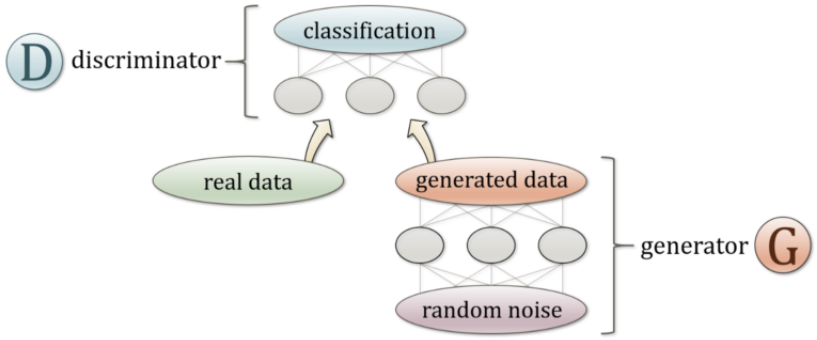


FIGURE 1.7 – Architecture simplifiée d'un réseau de type GAN [29]

Dans cette deuxième approche, le générateur et le discriminateur s'affrontent comme dans un jeu à deux joueurs. Le générateur est optimisé pour maximiser l'erreur de classification du discriminateur, et le discriminateur est optimisé pour minimiser son erreur de classification. Ainsi, à chaque itération les échantillons générés sont plus difficiles à distinguer des vrais échantillons.

Il existe deux principales variantes de l'architecture GAN (voir Figure 1.8). La première est le GAN conditionnel [30], qui incorpore une contrainte sur la sortie du générateur. Ce type de réseau est utilisé pour de l'augmentation de données, de la colorisation d'images, ou de la super-résolution. La seconde est le GAN cyclique [31], où deux générateurs et deux discriminateurs sont entraînés simultanément. Ces modèles sont utilisés pour le transfert de style ou le transfert de domaine (par exemple transformer des images de chevaux en images de zèbres).

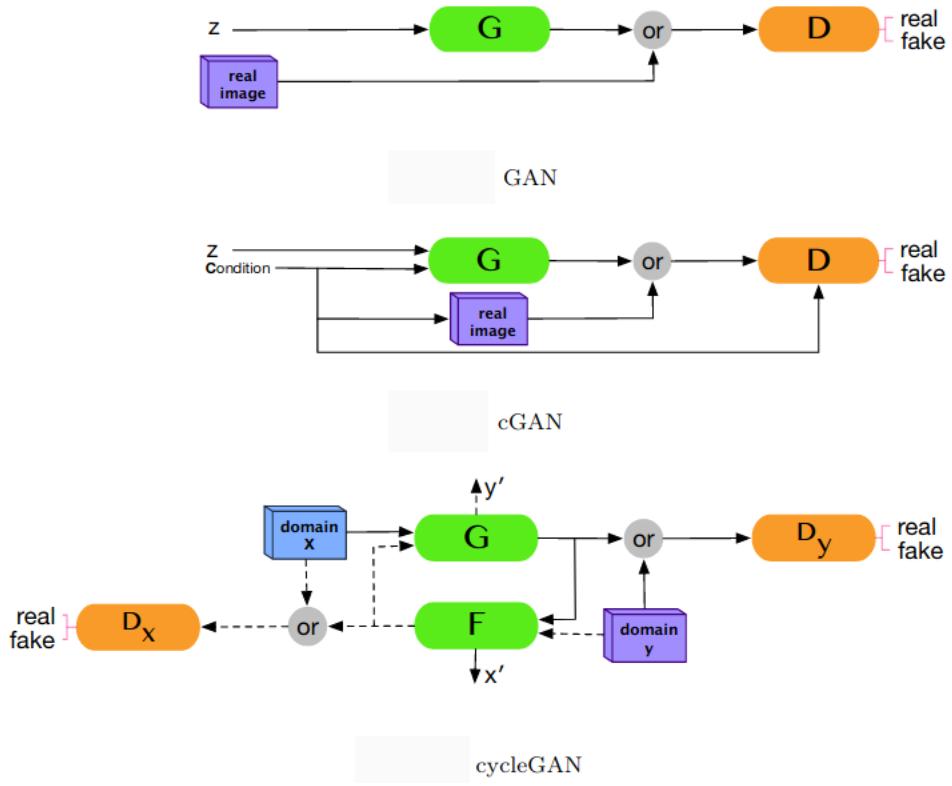


FIGURE 1.8 – Principales architectures de type GAN (figure modifiée issue de [10])

1.2.5 Réseaux transformers

Les réseaux *transformers* ont été créés pour des tâches de traitement du langage [32], en tant que stratégie alternative aux RNN pour le traitement de séquences temporelles. Contrairement aux RNN où les éléments de la séquence sont traités l'un après l'autre, les réseaux *transformers* utilisent directement l'entièreté de la séquence.

Cette architecture (voir Figure 1.9) est composée de deux gros blocs : une série d'encodeurs suivie par une série de décodeurs. De plus, elle utilise deux *inputs* : une séquence pour l'encodeur et une séquence appariée pour le décodeur, et produit un unique élément en sortie. Dans le cas d'une tâche de traduction du français vers l'anglais, l'encodeur recevra la phrase en français, le décodeur recevra la traduction correcte (vérité terrain) en anglais, et le réseau prédira en plusieurs étapes une traduction pour chaque mot de la phrase en français. A la phase d'inférence, l'*input* de l'encodeur sera uniquement constitué des mots déjà traduits à l'étape précédente.

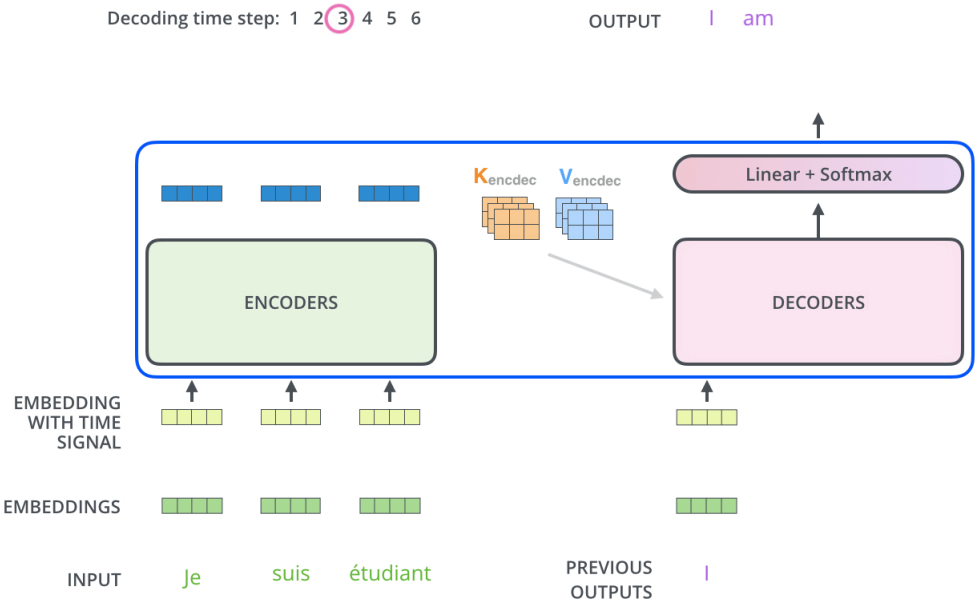


FIGURE 1.9 – Vision simplifiée de l'architecture d'un réseau *transformer* [33]

Les deux principes sur lesquels repose le *transformer* sont les suivants :

— L'auto-attention (*self-attention*) : chaque encodeur et décodeur possède une couche de *self-attention*, dont le but est de produire une représentation de la séquence d'entrée qui prenne en compte les relations sémantiques entre les différents éléments. Pour cela, un score est attribué à chaque élément vis-à-vis de chacun des autres éléments de la séquence. La somme de ces scores pour chaque élément donne une représentation de son importance relative dans la séquence d'entrée. Pour les encodeurs la séquence entière est utilisée, alors que pour les décodeurs seuls les éléments

déjà prédits par le réseau sont utilisés (le reste de la séquence est masqué).

— L'encodage des positions (*positional encoding*) : avant le premier encodeur du réseau, un vecteur qui encode l'ordre des éléments dans la séquence est ajouté à chaque élément. Ainsi si un élément apparaît deux fois il n'aura pas la même représentation.

Les réseaux *transformers* ont également été adaptés pour des tâches de traitement d'image, en particulier pour pallier à certains problèmes propres aux réseaux convolutifs. En effet, à part en utilisant des *kernels* de grande taille, les opérations de convolution ne permettent pas de modéliser des dépendances lointaines entre différents éléments d'une image. Les *transformers* peuvent donc être utilisés en complément ou à la place des couches de convolution. Dans ce cas, au lieu d'utiliser une image entière comme *input*, on utilise une séquence de patchs avec un encodage des positions en 2D (voir Figure 1.10). Ces modèles ont notamment été appliqués à des tâches de génération d'images [34], de détection d'objets [35], et de reconnaissance d'images [36].

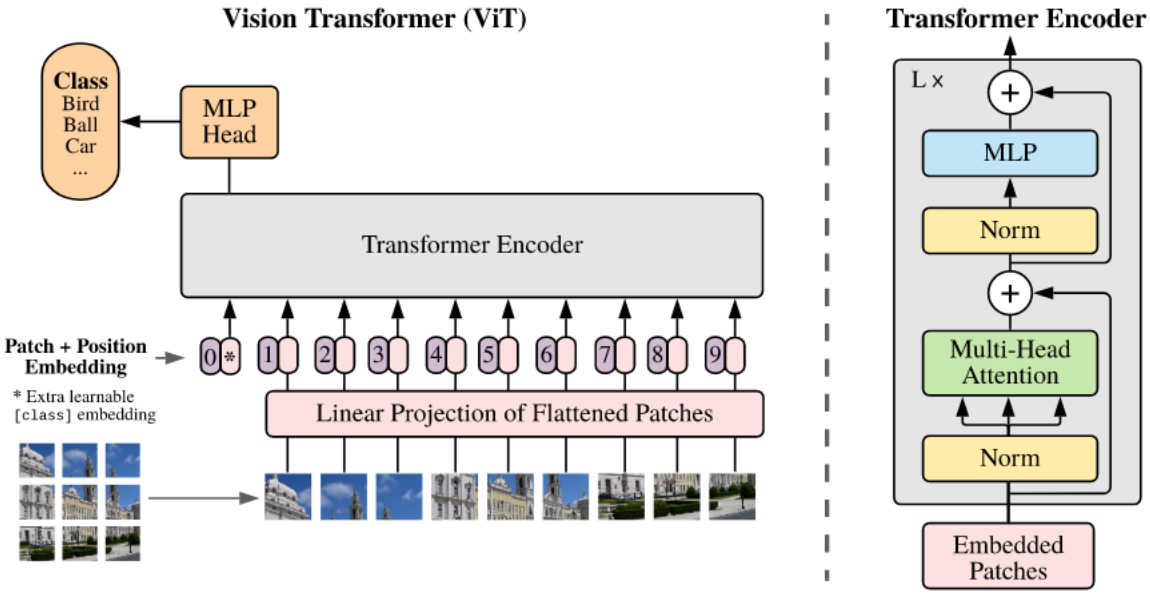


FIGURE 1.10 – Architecture du modèle Vision Transformer (ViT) [36]

1.2.6 Réseaux récurrents

Principe

Les réseaux récurrents (*Recurrent Neural Networks* (RNNs)) ont été inventés par David Rumelhart en 1986 [37], comme modèle adapté aux données séquentielles (avec un aspect temporel). Le principe de ce modèle est que, à chaque itération, l'entièreté de la séquence temporelle est traitée par chaque couche récurrente. Une couche récurrente peut être représentée sous forme repliée, où une connection cyclique est représentée, ou sous forme dépliée qui montre comment l'aspect temporel des données d'entrées est géré (Figure 1.11). Dans la forme dépliée, l'ensemble des opérations d'une couche L à un instant t est nommé cellule RNN N_t.

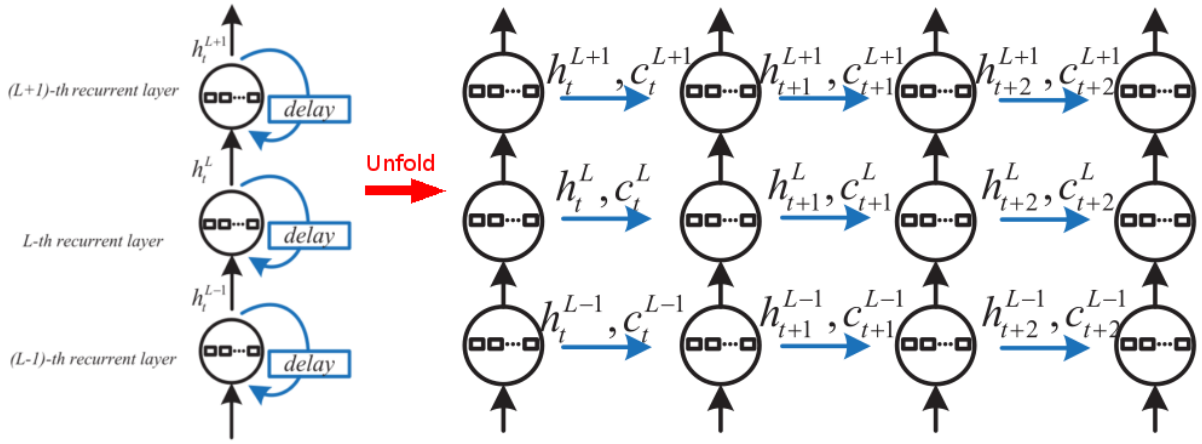


FIGURE 1.11 – Représentation repliée et dépliée d'une suite de couches RNN (image modifiée depuis [38])

Chaque cellule est liée à la cellule suivante de la même couche, ainsi la valeur de sortie h_t de la cellule N_t dépend à la fois de la donnée d'entrée courante x_t et de la valeur de sortie h_{t-1} de la cellule N_{t-1}. On a donc $h_t = \sigma\left(W_x x_t + W_h h_{t-1} + b\right)$, où W_x et W_h sont les matrices de poids de la couche récurrente, et b est la matrice des biais. Cette architecture permet donc en théorie de conserver de l'information tout au long d'une séquence temporelle. Cependant, l'ajout d'une nouvelle donnée à chaque pas de temps "écrase" l'information provenant des pas de temps précédents, ce qui rend l'architecture RNN originelle peu adaptée pour les dépendances à long-terme [39], et justifie l'introduction des variantes présentées dans les deux sous-sections ci-après.

Réseaux à mémoire à court et long terme

Une version améliorée des RNNs, appelée réseaux à mémoire à court et long terme (*Long Short Term Memory* (LSTMs)) a été proposée par Hochreiter et Schmidhuber en 1997 [40]. Cette architecture introduit l'état de la cellule (*cell state*) c_t, qui est également propagé au cours du temps. L'état est actualisé à l'aide de trois opérations, appelées "portes" :

— La porte "oubli" (*forget*) $f()$ utilise l'entrée courante x_t et la sortie de la cellule précédente h_{t-1}, et calcule un nouvel état c_t. Cela revient à supprimer une partie de l'information provenant de plus tôt dans la séquence.

— La porte "entrée" (*input*) $i()$ utilise l'entrée courante x_t, la sortie de la cellule précédente h_{t-1}, et une valeur d'état temporaire $\tilde{c}_t$ pour actualiser l'état courant c_t. Cela revient à ajouter des informations provenant de l'instant t.

— La porte "sortie" (*output*) $o()$ utilise l'entrée courante x_t, la sortie de la cellule précédente h_{t-1}, et le nouvel état après actualisation c_t, pour calculer la sortie de la cellule courante h_t. Cela revient à utiliser l'état de la cellule pour filtrer une partie de l'information avant de la transmettre à la cellule suivante.

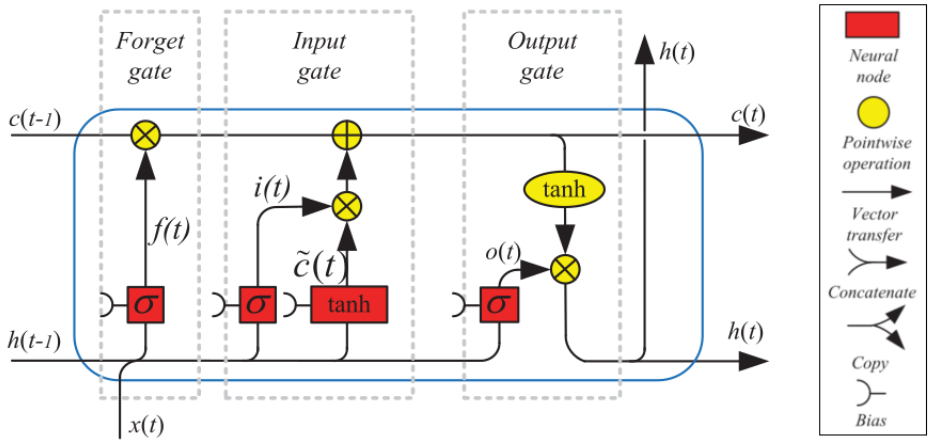


FIGURE 1.12 – Architecture d'une cellule LSTM [38]

A la fin des calculs, une copie de h_t est également envoyée à la cellule $t+1$ de la même couche, de même que l'état c_t. Aujourd'hui, les réseaux LSTM sont majoritairement utilisés, au détriment des simples RNN, pour divers problèmes comme la reconnaissance de parole, la modélisation de son, la prédiction de trajectoires, ou l'encodage de phrases en vecteurs (*sentence embedding*) [38].

Réseau de neurones récurrents à portes

L'une des variantes du RNN les plus récentes est le réseau de neurones récurrent à portes (*Gated Recurrent Unit* (GRUs)), qui a été présenté en 2014 [41]. La cellule GRU (voir Figure 1.13) combine les portes entrée et oubli en une porte "actualisation" (*update*) $z()$, fusionne l'état avec la sortie de la cellule h_t, possède une nouvelle fonction appelée "réinitialisation" (*reset*) $r()$, et n'a pas de porte sortie. L'avantage de cette architecture est qu'elle a des performances égales ou supérieures aux LSTMs, mais nécessite moins de paramètres, ce qui réduit à la fois les temps de calcul et les risques de sur-apprentissage.

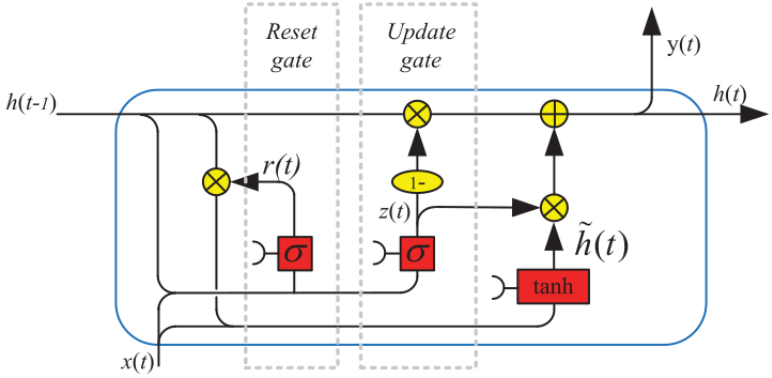


FIGURE 1.13 – Architecture d'une cellule GRU [38]

Application à l'analyse de séquences d'images

L'analyse de séquences d'images pose un problème spécifique : d'une part chaque image contient une information spatiale, et d'autre part la séquence d'images contient une information temporelle. Il faut donc que le modèle utilisé soit capable de résumer ces deux types d'information dans un vecteur de *features*, avant l'utilisation d'un classifieur.

Comme nous l'avons vu, une couche RNN donne en sortie une séquence contenant le même nombre de pas de temps que la séquence d'entrée. Il est ensuite possible de résumer l'information temporelle en appliquant une opération de *pooling* dans le temps. On peut également choisir de ne récupérer que le dernier pas de temps de la séquence de sortie, car elle contient théoriquement des informations résiduelles provenant de l'ensemble de la séquence. Enfin, les RNN bi-directionnels ont été créés dans l'optique d'équilibrer l'extraction d'information au travers de la séquence d'entrée [42]. Une couche RNN bi-directionnelle est simplement constituée de deux couches RNN indépendantes, l'une traitant l'entrée dans le sens chronologique, et l'autre dans le sens inverse. Les sorties de ces deux RNN sont ensuite fusionnées, par concaténation, somme, ou moyennage par exemple.

Une extension des RNN classiques a été présentée par Xinjiang *et. al.* [43]. Dans cet article, les auteurs s'intéressent à la prédiction de l'intensité de pluie sur des régions données. Pour résoudre ce problème de prédiction spatio-temporelle, ils utilisent une variante de la cellule RNN, dans laquelle les multiplications matricielles avec les matrices de poids sont remplacées par des opérations de convolution. Cette nouvelle architecture leur a donné de meilleurs résultats que les RNN classiques, ce qui montre l'utilité d'intégrer une approche convolutive pour le calcul de *features* spatiales, dans le cas où ces *features* sont aussi importantes que les informations temporelles.

1.3 Apprentissage par transfert

1.3.1 Principe

Cette approche consiste à utiliser un modèle pré-entraîné sur une grande quantité de données, qui peuvent différer dans leurs distributions statistiques des données sur lesquelles le réseau sera appliqué *in fine*. Dans un deuxième temps, les paramètres des couches du nouveau modèle sont initialisés avec ceux du réseau pré-entraîné, et on réalise un second entraînement avec les données de la nouvelle tâche à accomplir. Les paramètres de certaines couches sont fixés, et ne sont plus affectées par la rétro-propagation, afin de ré-entraîner uniquement les couches souhaitées. En général, cela nécessite moins de temps de calcul pour l'entraînement pour la tâche finale, permet au réseau de converger plus facilement qu'avec une initialisation aléatoire [44], et peut permettre la convergence de l'apprentissage même en présence insuffisamment d'exemples d'apprentissage dans le domaine-cible.

La difficulté repose sur le choix du jeu de données pour le pré-entraînement : il est par exemple possible de tirer parti de la grande taille des jeux de données d'images naturelles disponibles en libre accès (par exemple la base de données ImageNet [45]), mais il faut se demander si les caractéristiques de ces images sont assez semblables à celles du nouveau jeu de données, pour que le transfert d'apprentissage soit efficace.

De même, lors du ré-entraînement (*fine-tuning*), il faut choisir quelles couches seront entraînées avec les nouvelles images. Dans un réseau de neurones convolutif, les premières couches extraient des descripteurs génériques, comme des contours, et les couches profondes extraient des caractéristiques plus spécifiques aux images. L'approche privilégiée est donc souvent de ré-entraîner uniquement les dernières couches de convolution et les couches du classifieur.

On parle de transfert de domaine dans le cas où les deux jeux de données (appelés source et cible) diffèrent significativement dans leurs distributions statistiques respectives, par exemple dans le cas de données médicales lorsque l'on a des images provenant de méthodes d'acquisition différentes. La *review* de Choudhary *et. al.* [46] présente deux grands types de stratégies utilisées pour le transfert de domaine (voir Figure 1.14) :

— Transformation du domaine : les données du domaine source sont "traduites" dans le domaine cible. Pour cela, des réseaux génératifs sont en général utilisés.

— Transformation de l'espace des *features* : les données du domaine source et du domaine cible sont projetées dans un espace commun. Le but de l'entraînement est d'extraire des informations liées à la tâche et non aux spécificités du domaine.

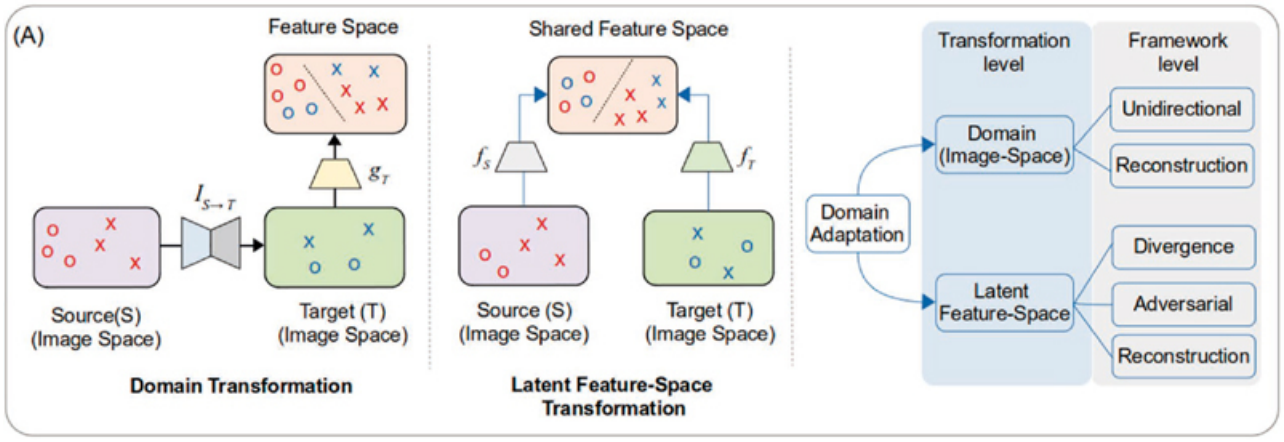


FIGURE 1.14 – Stratégies pour le transfert de domaine [46]

1.3.2 Apprentissage auto-supervisé

Dans le cas où beaucoup de données dont disponibles mais peu sont étiquetées ("labellisées"), ce qui est le cas si le processus d'acquisition de vérité terrain est long ou coûteux (vérification manuelle de chaque donnée, concertation avec des experts du domaine comme des médecins par exemple, etc.), il est possible de passer par une étape d'apprentissage auto-supervisé.

Les réseaux de type auto-encodeurs sont particulièrement utiles pour cette approche. En effet, dans un premier temps, l'apprentissage de filtres pour l'extraction de *features* se fait de façon non supervisée. Dans un second temps, la partie décodeur est remplacée par un classifieur, et le modèle est ré-entraîné pour la tâche souhaitée, tout en fixant les poids de la partie encodeur.

Il est aussi possible de pré-entraîner un modèle à l'aide d'une tâche "prétexte", pour laquelle on peut obtenir une vérité terrain facilement, par exemple la colorisation d'une image, le positionnement de patches à partir d'un patch central [47], ou la reconstruction d'une partie masquée d'une image [48].

1.4 Évaluation des modèles

Après entraînement, la performance des modèles est mesurée à l'aide de plusieurs métriques. Par exemple, dans le cas d'une tâche de classification binaire (positifs *versus* négatifs), on peut construire une matrice de confusion, qui représente la répartition des données d'entrée en fonction de leur classe véritable (vérité terrain) et de leur classe prédite par le modèle (voir Table 1.1). On définit ainsi comme dans toutes les études statistiques la notion de Vrais Positifs (*True Positives (TP)*), Vrais Négatifs (*True Negatives (TN)*), Faux Positifs (*False Positives (FP)*), et Faux Négatifs (*False Negartives (FN)*).

		Classe Véritable	
		Positif	Négatif
Classe Prédite	Positif	*TP*	*FP*
	Négatif	*FN*	*TP*

TABLE 1.1 – Matrice de confusion

On définit ensuite les métriques suivantes (Table 1.2) :

Métrique	**Définition**	**Formule**
Taux de Faux Positifs (*False Positive Rate (FPR)*)	%age d'éléments incorrectement prédits positifs parmis les éléments négatifs	$\frac{FP}{N} = \frac{FP}{FP+TN}$
Rappel (*Recall*) = Taux de Vrais Positifs (*True Positive Rate (TPR)*)	%age d'éléments correctement prédits positifs parmis les éléments positifs	$\frac{TP}{P} = \frac{TP}{TP+FN}$
Précision (*Precision*)	%age d'éléments correctement prédits positifs parmis les éléments prédits positifs	$\frac{TP}{TP+FP}$
Score F1 (*F1 Score*)	Ratio entre la Précision et le Rappel	$2\frac{Precision*Recall}{Precision+Recall}$
Exactitude (*Accuracy*)	%age de prédictions correctes	$\frac{TP+TN}{TP+TN+FP+FN}$

TABLE 1.2 – Métriques calculées à partir de la matrice de confusion

Enfin, une fois que le modèle a été choisi, il est possible de faire varier les hyperparamètres du modèle pour trouver ceux qui donnent la meilleure séparation entre les classes. On obtient ainsi une courbe ROC (*Reciver Operating Characteristics*), qui montre la variation du Taux de Vrais Positifs en fonction du Taux de Faux Positifs. L'aire sous la courbe (*Area Under Curve* (AUC)) résume cette courbe en donnant une mesure de la capacité du modèle à discriminer correctement entre les deux classes [49] (voir Figure 1.15).

Par exemple, si l'on utilise la *cross-entropy*, on pourra faire varier entre 0 et 1 le seuil de probabilité associé à la prédiction de la classe positive, et ainsi trouver le seuil de probabilité qui permettra la meilleure séparation entre positifs et négatifs.

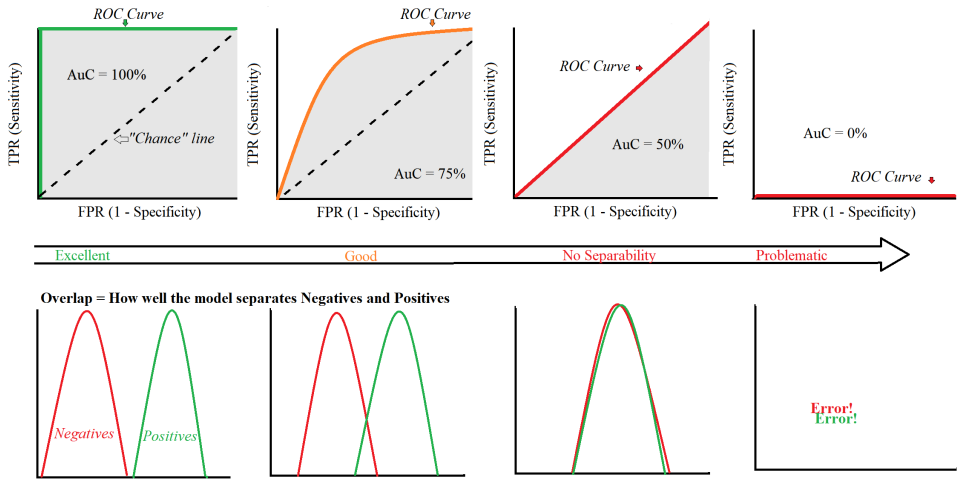


FIGURE 1.15 – Interprétation de la courbe ROC d'un modèle pour une classification binaire [50]

On qualifie très souvent les réseaux de neurones de "boîtes noires", dont il est très difficile d'interpréter les mécanismes sous-jacents. Différentes techniques ont été présentées pour mieux comprendre ces mécanismes. Par exemple pour mieux comprendre ce qu'apprennent les couches intermédiaires, Hosseini *et. al.* [51] présentent une projection *t-SNE* [52] de leurs images à chaque couche du réseau. Cet algorithme de réduction de dimensions permet de visualiser la discrimination de chaque classe à travers les couches du réseau (voir Figure 1.16).

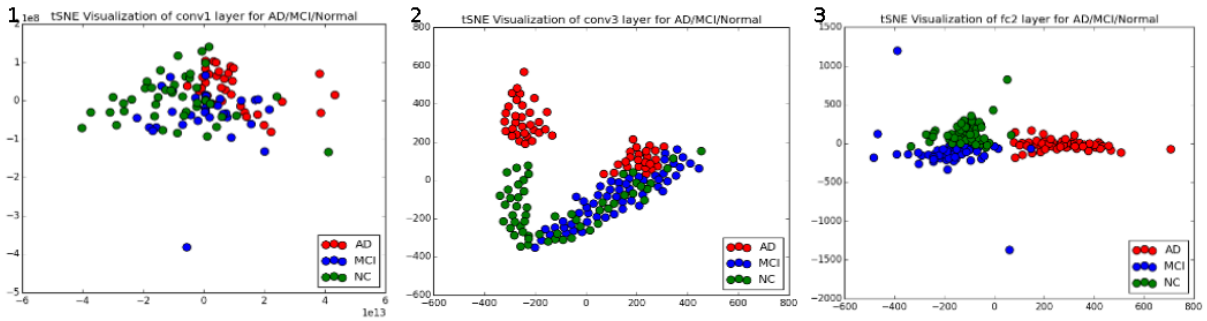


FIGURE 1.16 – [51]

Après cette présentation des différents concepts liés au domaine des réseaux de neurones profonds, nous présenterons dans le prochain chapitre les spécificités de leur application dans le cadre de l'analyse de données bio-médicales, et en particulier pour de l'analyse de données d'imagerie médicale associées à des données cliniques quantitatives et qualitatives.

Chapitre 2

Application des modèles d'apprentissage profond aux données biomédicales

Dans ce chapitre, nous présenterons tout d'abord les caractéristiques des données biomédicales, et les limites propres à leur utilisation pour de l'apprentissage automatique. Nous dresserons ensuite un état de l'art des principaux modèles utilisés pour l'analyse des IRM structurales et fonctionnelles. Nous présenterons également les travaux focalisés sur l'utilisation combinée de plusieurs modalités de données. Enfin, nous discuterons des stratégies utilisables pour faciliter l'apprentissage de ces modèles et améliorer leurs performances.

2.1 Caractéristiques et limites des données biomédicales

Les images d'organes sont acquises avec diverses techniques : IRM structurale, fonctionnelle, PET scan, ... Ces types d'imagerie donnent chacune des informations différentes, utilisables pour caractériser des pathologies spécifiques (maladies neurologiques, cancers, *etc.*). Elles ont également des caractéristiques spécifiques, qui doivent être prises en compte lors de la conception de modèles pour l'apprentissage profond. Dans la suite de cette section, nous décrirons en particulier l'usage de ces techniques d'imagerie pour l'étude du cerveau, étant donné qu'il s'agit de notre organe d'intérêt dans le cadre de ces travaux.

2.1.1 IRM structurales

L'IRM structurale est utilisée pour obtenir des informations sur la forme et la taille des régions du cerveau, ainsi que pour repérer des tissus anormaux ou des changements de composition dans les tissus [53].

Sur les images, les tissus ont des nuances de gris différentes en fonction de leur teneur en eau, ce qui permet d'identifier les différents types de tissus, par exemple la matière grise qui est composée de 80 % d'eau, et la matière blanche qui est composée de 70% d'eau (voir Figure 2.1).

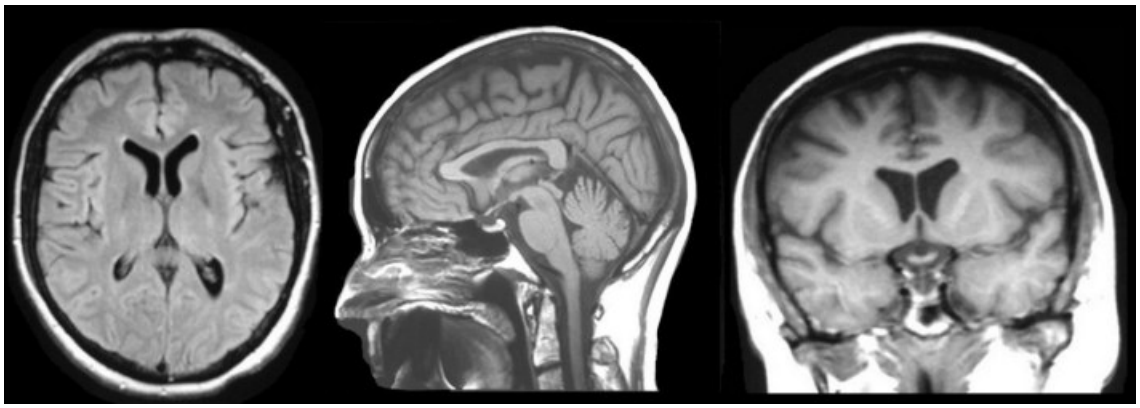


FIGURE 2.1 – Coupe axiale, sagittale, et coronale d'une IRM de cervau humain (Source : [54])

Les images structurales sont en 3D volumique, c'est-à-dire composées d'une pile d'images 2D obtenues lorsque le cerveau est scanné. Bien que chaque "tranche" de cerveau puisse être analysée en tant que telle, l'ensemble des informations morphologiques ne peut être compris que grâce à une analyse prenant en compte leur aspect tridimensionnel.

2.1.2 IRM fonctionnelle

L'IRM fonctionnelle (IRMf) ajoute une dimension temporelle à l'IRM structurale. Ce sont donc des données en 4 dimensions, qu'il faut soit réduire, soit traiter comme une série temporelle d'images 3D.

Cette technique consiste le plus souvent à cartographier les niveaux d'activité cellulaire des zones du cerveau d'un sujet lorsqu'il est au repos (*resting-state*), effectue une tâche (par exemple en regardant une image, ou en effectuant un mouvement), ou est exposé à un stimulus [53].

Les zones les plus actives étant caractérisées par un plus fort approvisionnement en oxygène, il est ainsi possible de visualiser la teneur en oxygène sous la forme de variations de couleurs, avec le signal BOLD (*Blood Oxygen Level Dependant*) (voir Figure 2.2). En fonction du contexte de l'IRM fonctionelle, les résultats obtenus seront très différents, ce qui doit être pris en compte durant l'analyse.

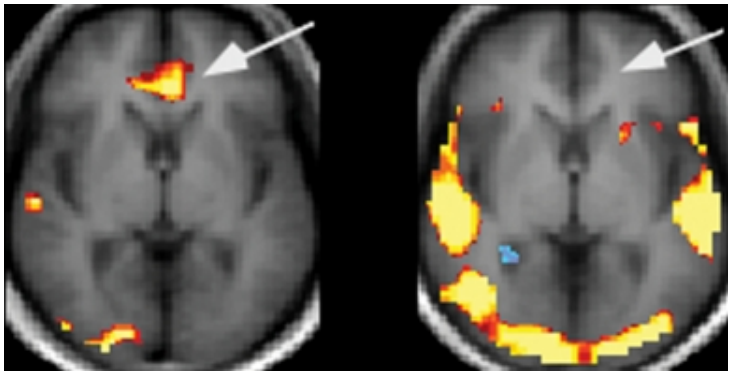


FIGURE 2.2 – Exemple d'utilisation de l'IRM fonctionnelle pour l'étude de la réponse neurologique à des stimuli en rapport avec la cocaïne (sujet dépendant à gauche, sujet sain à droite) [53]

2.1.3 Données cliniques

Les données cliniques acquises durant un suivi de patient sont de natures variées. Les données les plus basiques sont des données démographiques comme l'âge, le sexe biologique, l'origine ethnique, *etc.*. Des prélèvements biologiques de molécules d'intérêt peuvent également être effectués. Dans le cas d'études portant sur des pathologies neurologiques, les médecins compilent également de nombreux scores cognitifs, qui permettent de mesurer l'avancée des maladie. Ces scores peuvent être basés sur des réponses à des questionnaires ou auto-questionnaires, ou sur une performance pour la réalisation de tâches diverses. De plus, les antécédents médicaux ou les traitements médicamenteux en cours sont souvent inclus également, car ils apportent des informations sur l'historique du patient. Enfin, dans le cas où la pathologie est liée au patrimoine génétique, les allèles des gènes impliqués seront également des données importantes.

Ainsi, toutes ces données, que nous appelons dans ce mémoire "données cliniques" forment donc un mélange hétérogène de données quantitatives et qualitatives, et n'ont pas *a priori* la même importance pour un problème donné.

2.1.4 Pré-traitements classiques

Certaines étapes de pré-traitement sont caractéristiques de l'imagerie cérébrale :

— Recalage de l'image par rapport à une référence (*image registration*) [55, 56].

— Extraction des tissus appartenant uniquement au cerveau (*skull stripping*) [55, 57, 56].

— Correction des artefacts liés aux mouvements lors de l'acquisition [58].

Il est également nécessaire de normaliser les images avant l'entraînement d'un réseau de neurones, en général par centrage réduction des données [59]. Afin de réduire la taille des images d'entrée, il est aussi possible de supprimer une partie des images qui constituent chaque volume [55], ou d'extraire des volumes d'intérêt autour de régions identifiées à l'avance [56]. Ces régions sont identifiées à partir d'atlas du cerveau humain.

Dans le cas des données cliniques, d'autres pré-traitements sont nécessaires :

— Sélection des variables d'intérêt.

— Normalisation des données quantitatives.

— Choix d'une stratégie pour gérer les données manquantes (voir paragraphe suivant).

2.1.5 Limites

Travailler avec des données biomédicales pose d'autres problèmes dans le contexte de l'apprentissage automatique. Les études cliniques sont en général réalisées sur de petites cohortes, même si de plus grande bases de données se mettent en place depuis quelques années. La question de l'étiquetage des données avec une vérité terrain se pose également : cette vérité terrain peut être un diagnostique donné par un expert, mais au cas où elle est indisponible ou incomplète il est nécessaire d'en créer une à partir des seules informations disponibles. Un des problèmes principaux de l'apprentissage profond appliqué aux données biomédicales est donc le faible nombre de bases de données de grande taille, correctement annotées et labellisées [60].

Dans le cas d'une étude longitudinale, c'est-à-dire d'un suivi de patients dans le temps pendant plusieurs mois à plusieurs années, la dimension temporelle à long terme doit également être prise en compte de façon appropriée. Pour cela, on peut traiter l'ensemble des pas de temps comme une série temporelle, ou évaluer les changements entre deux pas de temps choisis. Enfin, un des problèmes caractéristiques des études cliniques est la possibilité d'avoir des données manquantes. Il faut donc choisir une stratégie pour les inférer ou s'en accommoder, et pour prendre en compte les individus qui sortent de l'étude.

Little & Rubin [61] définissent trois types de données manquantes :

— Données manquantes de manière complètement aléatoire, lorsque la probabilité d'avoir une donnée manquante n'est liée ni à une variable indépendante,*i. e.* variable manipulée par l'expérimentateur), ni à une variable dépendante,*i. e.* variable

qui subit l'effet des variables indépendantes (par exemple un problème sur une machine lors de l'acquisition).

— Données manquantes aléatoirement, lorsque la probabilité d'avoir une donnée manquante peut être liée à des variables dépendantes ou indépendantes, mais pas à l'issue de l'essai clinique en elle-même (par exemple un sujet qui quitte l'étude, en raison d'une caractéristique connue, comme son âge).

— Données manquantes par omission prévisible, lorsque la probabilité d'avoir une donnée manquante est liée à l'issue de l'essai clinique (par exemple si l'état de santé du sujet est trop mauvais pour qu'il puisse passer les tests médicaux).

Plusieurs stratégies existent pour faire face à ce problème [62]. La première est simplement de filtrer le jeu de données, et de ne conserver que les sujets dont les données sont complètes. Cette option pose deux problèmes : d'une part elle conduit à un jeu de données significativement plus petit, et d'autre part elle entraîne un bais lorsque les données manquantes écartées ne sont pas de type "complètement aléatoires". L'autre stratégie est de remplacer les valeurs manquantes par des valeurs plausibles, à l'aide d'une méthode d'imputation (imputation simple ou multiple, maximum de vraisemblance, ...).

A titre d'exemple l'impact de la méthode d'imputation sur la classification de patients de la base de données Alzheimer's Disease Neuroimaging Initiative a été étudiée [63]. Dans cet article, les auteurs comparent six méthodes d'imputation (zéro, moyenne, médiane, moyenne windsorisée, k plus proches voisins (kNN), et algorithme espérance-maximisation (EM), à l'utilisation d'un jeu de données réduit sans valeurs manquantes. Leurs conclusions montrent que les classifieurs étudiés (SVM et forêts d'arbres décisionnels) sont plus performants et plus robustes lorsqu'ils ont été entraînés sur le jeu de données complet. Les auteurs notent également l'importance de la distribution statistique des données pour le choix de la méthode d'imputation. Enfin, ils soulignent l'intérêt de développer des algorithmes capables de gérer les données manquantes, pour limiter la nécessité d'imputer des valeurs.

Enfin, dans le cas de l'étude du cerveau, il a été montré qu'il existe une plus grande variabilité morphologique entre deux sujets donnés, qu'entre le même sujet avant et après une maladie neurologique. Il est donc nécessaire de privilégier l'étude de l'évolution morphologique des sujets, plutôt que la comparaison des sujets à un atlas ou modèle de référence [64].

2.2 Analyse des IRM structurales

Plusieurs *reviews* s'intéressent à l'utilisation de l'apprentissage profond dans le domaine neuro-médical [65, 66, 60]. Concernant l'étude d'IRM, quatre grands domaines d'application sont recensés [65] :

— La détection d'images, qui permet d'identifier des tissus d'intérêt. Par exemple, les auteurs de [67] utilisent un DNN pour la détection de nodules sur des IRM de poitrine, ou de lésions sur des mammographies. Les auteurs de [68] utilisent un CNN 3D pour la détection de micro-saignements cérébraux à partir d'IRM.

— Le recalage d'images (*image registration*), qui a pour but de faire correspondre et superposer deux images ou plus, qui peuvent avoir été pris à des moments différents, avec des méthodes d'acquisition différentes, ou dans des conditions expérimentales différentes). Par exemple, les auteurs de [69] présentent une méthode de recalage pour les IRM de cerveau.

— La segmentation d'image, qui permet de séparer différents tissus les uns des autres. Le réseau le plus connu pour cette application est U-Net [26], mais aussi sa variante 3D V-Net [70]. Ainsi, par exemple, Zhou *et. al.* [71] utilisent l'architecture U-Net 3D pour la segmentation de tumeurs dans le cerveau à partir de plusieurs modalités d'IRM structurale (FLAIR, T1, T2, T1 avec augmentation de contraste). Dans un second article [72], les mêmes auteurs utilisent également une variante de l'architecture U-Net pour la segmentation de tumeurs de cerveau, cette fois en utilisant les corrélations entre les modalités IRM pour être plus robustes à une éventuelle modalité manquante.

— La classification d'images (supervisée ou non), qui a pour but de grouper des images selon leurs caractéristiques ou d'affecter chaque image à des classes pré-définies, et pour laquelle nous présenterons plusieurs exemples de tâches de classification (supervisée ou non) dans la suite de cet état de l'art.

Dans le domaine de l'analyse d'image de cerveaux humains par apprentissage profond, il existe un grand nombre d'articles sur la maladie d'Alzheimer, dans lesquels l'objectif est de catégoriser les patients sains (NC), les patients atteints du stade précurseur (MCI), et les patients atteints de la maladie d'Alzheimer (AD) [66]. Toutes ces études sont basées sur des images IRM issues de la base ADNI (*Alzheimer's Disease Neuroimaging Initiative*) [5]. La comparaison de ces articles fait ressortir les différentes approches et stratégies utilisées (voir Table 2.1 ci-dessous). Les bases de données utilisées étant différentes d'une méthode à l'autre, nous avons volontairement omis de comparer leurs performances dans la Table 2.1. En revanche, nous pouvons donner un ordre d'idée des performances d'*accuracy* rapportées pour ces méthodes, comprises entre 79% [57] et 96,85% [55].

Ref	Données	Prétraitement des données	Architecture
[51]	30 sujets pour le pré-entraînement ; 210 sujets ADNI dataset (70 AD, 70 MCI, 70 NC) pour la tâche finale	données déjà prétraitées	CNN 3D basé sur un AutoEncodeur Convolutif
[59]	ADNI dataset (755 AD, 755 MCI, 755 NC)	normalisation, centrage réduction	CNN 3D basé sur un AutoEncodeur
[55]	ADNI dataset	correction du mouvement, skull stripping, flou Gaussien, filtre passe bas, recalage, suppression des 10 premières slices	CNN 2D (LeNet)
[57]	ADNI dataset (50 AD, 120 MCI, 61 NC)	Skull stripping, normalisation	CNN 3D (VGG, ResNet)
[73]	ADNI dataset (900 sujets)	corrections, normalisation, skull stripping, sélection de slices	CNN 2D (VGG16)

TABLE 2.1 – Comparaison des approches d'apprentissage profond utilisées dans l'étude d'images IRM pour la maladie d'Alzheimer

L'architecture utilisée classiquement est le réseau de neurones convolutif (voir Section 1.2.1) . Les opérations successives de convolution conservent les caractéristiques les plus importantes de l'image. Les réseaux CNN profonds peuvent être entraînés de manière *end-to-end*, où le réseau apprend automatiquement les *features* les plus utiles pour la tâche finale (classification par exemple). Ainsi, elles peuvent remplacer le calcul de descripteurs manuels. Cependant, pour le même type d'images d'entrée, différents formats peuvent être utilisés pour entraîner les réseaux : images 2D [55, 73], images 3D [51, 59, 57, 56], régions d'intérêt sélectionnées à partir d'atlas de cerveau humain [56].

L'intérêt principal d'utiliser des entrées à deux dimensions est la réduction de la charge mémoire et du temps de calcul nécessaires durant l'entraînement, mais cela a pour inconvénient de perdre une partie de l'information originale, par rapport à l'information tridimensionnelle contenue dans les images. En effet, en traitant les *slices* (2D) une par une, le modèle n'a qu'une information partielle sur la morphologie des différentes régions du cerveau.

L'utilisation de volumes d'intérêt permet de conserver cette information, mais nécessite d'identifier préalablement les zones impliquées dans la maladie. À noter qu'une approche 2D+ utilisant les projections selon les trois axes a été utilisée par Aderghal *et. al.* [74].

2.3 Analyse des IRM fonctionnelles

L'étude des IRM fonctionnelles de repos permet d'analyser la connectivité fonctionnelle, qui peut présenter des altérations dans des cas de troubles neurologiques (autisme, dépression, schizophrénie). Dans le domaine du *machine learning*, on distingue deux approches pour l'étude de cette connectivité : une approche non-supervisée, dont le but est de comprendre l'organisation fonctionnelle du cerveau sain, et une approche supervisée, dont le but est d'utiliser les schémas de connectivité pour faire des prédictions à l'échelle des individus (diagnostique ou pronostique).

Ainsi, les approches non-supervisées visent à délimiter des sous-unitées fonctionnelles dans le cerveau, à la manière d'un atlas pour la partie structurelle. Pour cela, l'information en 4 dimensions est décomposée en une superposition linéaire d'unités distinctes dans l'espace mais qui présentent des dynamiques similaires dans le temps. Parmi les méthodes utilisées, on peut nommer le *clustering* hiérarchique, le partitionnement en k-moyennes (*k-means*), et l'Analyse en Composantes Indépendantes (ACI) [75]. Une *review* datant de 2018 recense les différents modèles d'apprentissage profond utilisés pour analyser des IRM fonctionnelles [76].

L'architecture de type réseau de neurones récurrents convolutifs (Conv-RNN) [43] a été utilisée dans plusieurs études portant sur l'analyse d'IRM fonctionnelles. Dans [77], les auteurs utilisent un réseau LSTM avec des convolutions 1D sur des IRMf, après segmentation de 236 régions d'intérêt basées sur un atlas. Cette approche a été utilisée pour prendre en compte les variabilités individuelles, au lieu d'utiliser une méthode basée sur des modèles statiques de connectivité. De plus, les modèles de connectivité se basent sur des corrélations temporelles pour grouper les régions du cerveau selon leurs activités, mais cela ne prend pas entièrement en compte l'évolution de chaque région au cours du temps.

Les Conv-RNN sont également très utilisés pour la classification d'individus possédant un Trouble du Spectre Autistique (TSA), à l'aide de la base de données publique ABIDE [78]. Dans leur article [79], les auteurs implémentent une architecture complexe qui utilise à la fois des CNN 3D et un LSTM convolutif 3D pour extraire des *features* spatiales et temporelles à partir des IRMf (CNN 3D $\rightarrow$ Conv3D-LSTM $\rightarrow$ CNN 3D $\rightarrow$ *pooling* dans le temps).

2.4 Apprentissage multimodal

Dans le cas du suivi médical d'un patient, différentes modalités de données sont généralement acquises au cours du temps : imagerie, mesures biologiques, et autres données cliniques. Ces données sont complémentaires et corrélées, et constituent pour un médecin différents "angles de vue" sur la maladie. Il est donc théoriquement possible de concevoir un modèle d'apprentissage profond capable d'utiliser conjointement ces différentes modalités et d'en extraire des corrélations. De plus, l'utilisation de plusieurs modalités peut dans certains cas pallier à la présence de bruit ou de données manquantes dans certaines modalités.

De nombreuses études utilisent différentes modalités d'imagerie dans le domaine médical, mais très peu utilisent à la fois des données d'imagerie et des données cliniques. Ces travaux sont répertoriés dans la Table 2.2 ci-après, et **ne sont pas limités à l'analyse de pathologies neurologiques**.

Ref	Modalités	Extraction descripteurs	Architecture	Avantages	Limites
[80]	imagerie (IRM + PET), scores cognitifs, mesures biologiques	Mesures morphométriques	Multi-Kernel SVM	Identification de l'importance chaque *kernel* / modalité sur la classification ; Ajout facile d'une nouvelle modalité au modèle ; Meilleure performance qu'utiliser une SVM sur la concaténation des descripteurs	Besoin de pré-traiter les données et de calculer des descripteurs manuellement ; En fonction de l'avancée de la maladie, les changements ne sont pas toujours détectables par toutes les modalités
[81]	imagerie (IRM), scores cognitifs, scores cliniques	Mesure de connectivité, lésions dans la matière blanche, mesures sur la matière grise, diffusion des molécules d'eau, mesure d'aire sur la moelle épinière	Multi-Kernel SVM	Identification de l'importance chaque *kernel* / modalité sur la classification	Données provenant d'un seul centre, donc possible biais d'acquisition ; Vérité terrain dépendante de l'annotation manuelle d'un expert, qui peut s'être trompé
[82]	imagerie (5 modalités différentes), textes libres, données démographiques	CNNs 2D pour les images, mots clés et ontologie médicale pour les textes libres	Forêt d'arbres décisionnels	Identification de l'importance chaque vecteur de descripteurs / modalité sur la classification ; Modèle facilement interprétable par un humain	Grande importance du choix des descripteurs
[83]	imagerie (cervigrame), données cliniques (scores, mesures biologiques, données démographiques)	CNN 2D pour les images, puis réduction de dimension	DNN avec fusion précoce de deux entrées	Descripteurs extraits automatiquement ; Apprentissage des corrélations entre les modalités	Choix des hyperparamètres par expérimentation
[84]	imagerie (IRM), données cliniques (scores, mesures biologiques, données démographiques)	Mesure de la connectivité, mesures d'épaisseur corticale et de volumes sous-corticaux	DNN avec fusion tardive de trois sous-réseaux correspondant aux trois modalités	Apprentissage des corrélations entre les modalités	Choix des hyperparamètres par expérimentation
[85]	imagerie (IRM), données cliniques (scores et facteurs de risque)	Mesures d'épaisseur corticale et de volumes sous-corticaux	DNN avec fusion progressive des modalités	Etude de l'évolution d'un patient à l'aide d'une paire de visites médicales	Choix des hyperparamètres par expérimentation, N'utilise pas l'IRM en tant que telle
[86]	imagerie (IRM), données cliniques (scores, facteurs de risque, prélèvements biologiques)	Mesures d'épaisseur corticale et de volumes sous-corticaux	RNN avec fusion tardive des modalités	Etude de l'évolution d'un patient à l'aide d'une séquence de visites médicales	Choix des hyperparamètres par expérimentation, N'utilise pas l'IRM en tant que telle

TABLE 2.2 – Etudes utilisant l'apprentissage multimodal avec des données d'imagerie et des données cliniques

Comme expliqué précédemment, les données ont des dimensions et des caractéristiques différentes selon leur modalité. La question de la fusion des modalités est discutée dans la *review* de Ramachandram *et. al.* [87]. En particulier les auteurs font la différence entre la fusion précoce des modalités (par concaténation ou Analyse en Composante Principale, avant utilisation du modèle), la fusion tardive (par fusion des décisions prises par des modèles indépendants pour chaque modalité), et la fusion intermédiaire (par exemple par utilisation de réseaux de neurones dans lesquels les premières couches sont spécifiques à chaque modalité puis se rejoignent au niveau d'une couche intermédiaire) (Voir Figure 2.3).

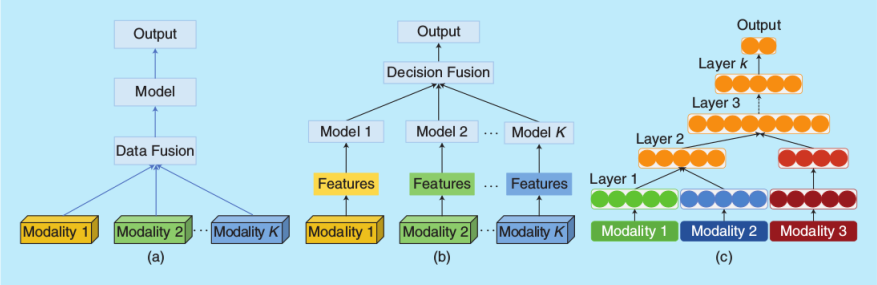


FIGURE 2.3 – Stratégies de fusion de données multimodales : (a) Fusion précoce ; (b) Fusion tardive ; (c) Fusion intermédiaire [87]

Hinrichs *et. al.* [80], Eshaghi *et. al.* [81] (voir Figure 2.4) utilisent des méthodes similaires, consistant à calculer des descripteurs ou autres informations à partir des données d'imagerie, puis à fusionner ces données avant l'utilisation d'un classifieur (Machine à Vecteur de Support Multi-Kernel). Il s'agit ici d'une **fusion précoce** des données.

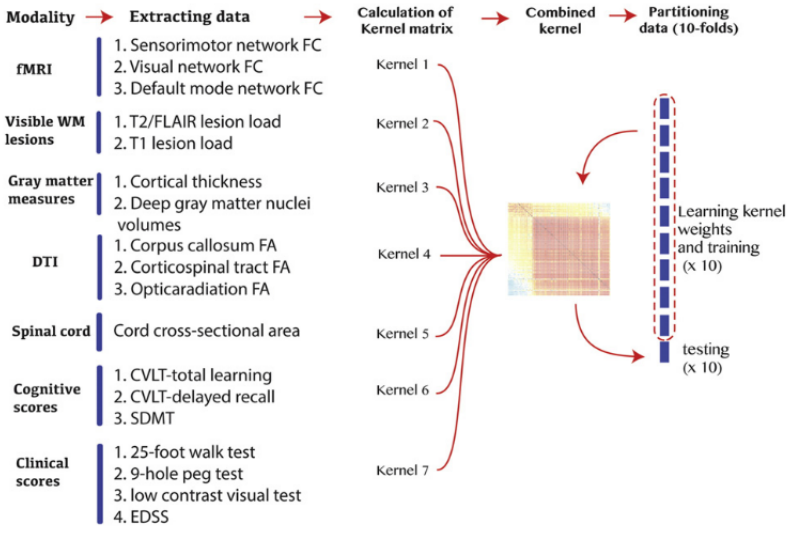


FIGURE 2.4 – Stratégie d'apprentissage multimodal utilisée par [81]

La méthode de Galveia *et. al.* [82] (voir Figure 2.5), dans le domaine de l'ophtalmologie, utilise aussi une **fusion précoce** : après avoir extrait des descripteurs à partir d'images et de textes libres, les modalités sont fusionnées puis traitées par une forêt d'arbres décisionnels (*random forest*).

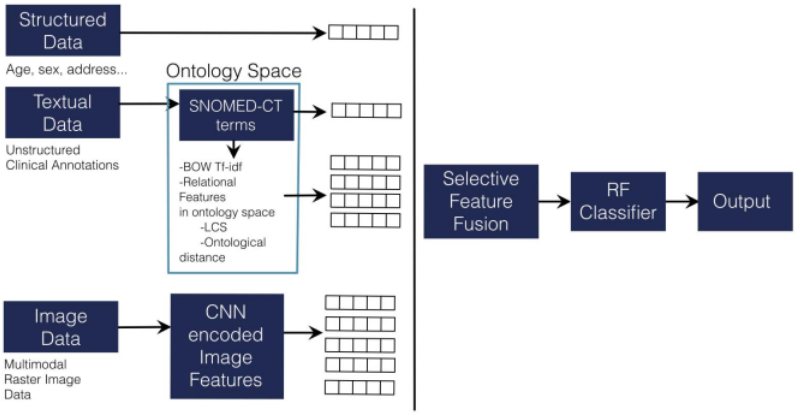


FIGURE 2.5 – Stratégie d'apprentissage multimodal utilisée par [82]

Xu *et. al.* [83] (voir Figure 2.6) utilisent une stratégie de **fusion intermédiaire**, pour le diagnostique de cancer du col de l'utérus. Les auteurs utilisent un unique réseau de neurones qui prend en entrée simultanément les images (cervigrammes) et les données cliniques. Les images passent par une suite de couches de convolution, tandis que les données cliniques passent par une suite de couches entièrement connectées. Enfin, les deux sous-réseaux sont fusionnés dans un réseau de neurones classique (entièrement connectés) pour la classification.

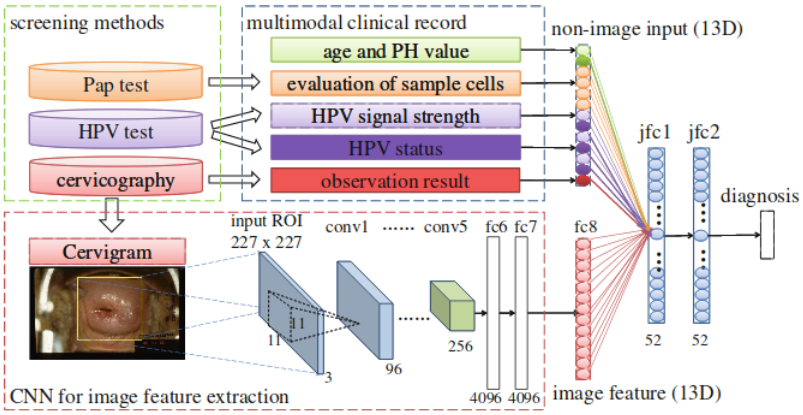


FIGURE 2.6 – Stratégie d'apprentissage multimodal utilisée par [83]

L'approche de Van der Burgh *et. al.* [84] (voir Figure 2.7) est aussi basée sur des réseaux de neurones, mais il s'agit ici d'un **mélange entre une fusion tardive et intermédiaire**. Ce modèle est utilisé pour l'étude de la maladie de Charcot. Chaque modalité est traitée par un réseau indépendant, qui renvoie une prédiction et une probabilité associée. Pour obtenir une prédiction finale, ces probabilités d'appartenance à chaque classe sont traitées simultanément par un dernier réseau entièrement connecté.

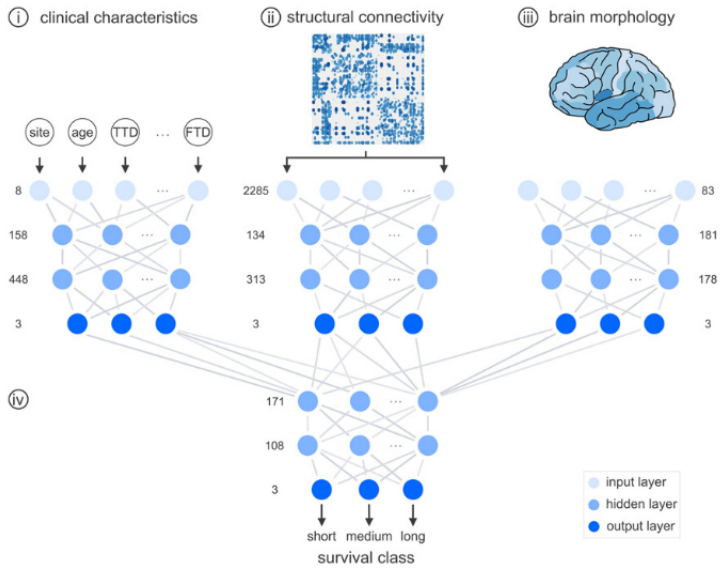


FIGURE 2.7 – Stratégie d'apprentissage multimodal utilisée par [84]

L'utilisation de plusieurs *kernels*, dans les méthodes de Hinrichs *et. al.* [80] et Eshaghi *et. a.* [81],permet d'étudier et d'interpréter facilement l'importance de chaque modalité ou sous-modalité sur le résultat de la classification, car les poids associés aux *kernels* varient en fonction de leur importance. Cependant, les différents *kernels* sont combinés linéairement, donc il n'est pas possible de déduire des corrélations entre les modalités [80, 81]. Au contraire, les architectures de Xu *et. al.* [83] et Van *et. al.* [84] présentent toutes les deux une partie d'apprentissage conjoint entre les différentes modalités. Cela permet au réseau d'apprendre également des informations de corrélations entre modalités. L'approche utilisant une fusion intermédiaire semble donc plus adaptée à notre problématique. Cependant, aucune des approches présentées précédemment n'utilise d'images en 3D. Ainsi, nous proposerons dans la suite de ce manuscrit un modèle multimodal avec fusion intermédiaire, dont l'une des modalités sera l'imagerie médicale en 3D volumique (IRM de cerveau).

Deux modèles de l'état de l'art ont été proposés contemporainement à ces travaux de thèse. Ces deux modèles sont décrits dans les paragraphes suivants, et seront comparés à notre approche dans les chapitres suivants.

Le premier modèle est le modèle *Longitudinal Siamese Network* (LSN), proposé par Bhagwat *et. al.* en 2018 [85]. Le modèle que les auteurs proposent (Figure 2.8) est multimodal. La première partie de leur architecture est un réseau siamois prenant en entrée les données issues d'IRM acquises à l'inclusion d'un patient et à une visite de suivi. Ces données sont les valeurs d'épaisseur corticale moyenne pour 78 régions d'intérêt dans le cerveau du patient. Après une suite de couches entièrement connectées, les sorties des deux branches sont réunies par concaténation. Ce résultat est ensuite multiplié par un vecteur de caractéristiques calculé à partir d'une information génomique (nombres d'allèles APOE4, qui est un facteur de risque connu pour la maladie d'Alzheimer). Enfin, un attribut clinique, le score MMSE, est concaténé à ce résultat, suivi par une dernière couche entièrement connectée, conduisant à la prédiction finale.

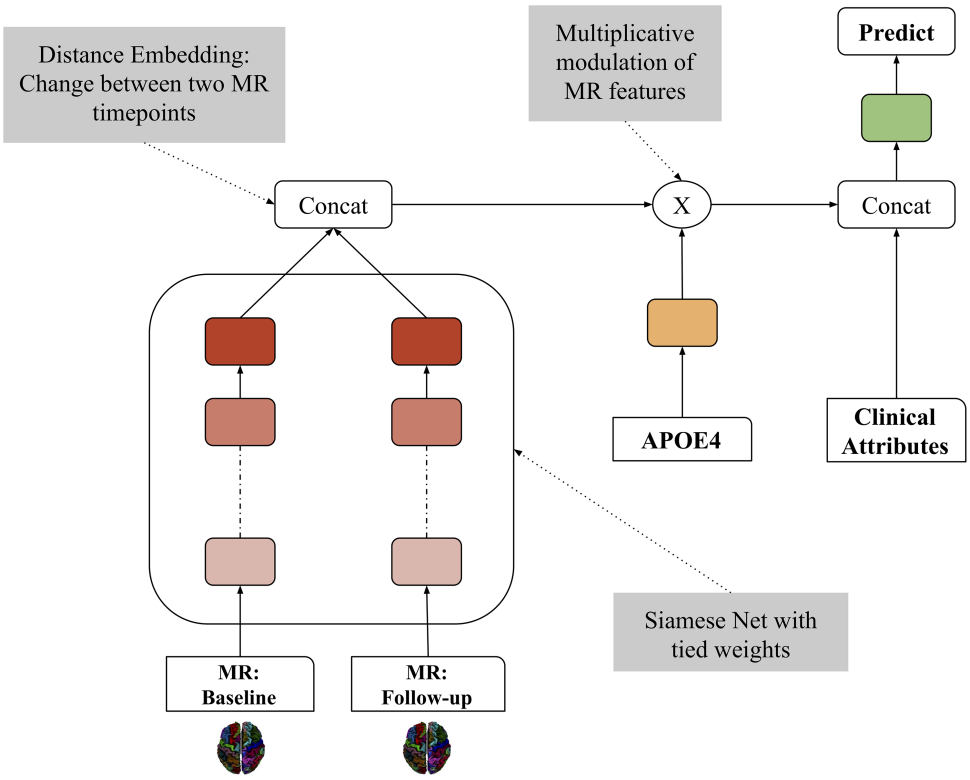


FIGURE 2.8 – Représentation simplifiée de l'architecture LSN proposée par Bhagwat *et. al.* [85]. *Baseline* : inclusion du patient, *follow-up* : visite de suivi

Avec un jeu de données de 1116 sujets ADNI, le modèle LSN obtient une *accuracy* de 0.94 et une AUC de 0.99, mais comporte selon notre analyse plusieurs limites :

— D'une part, un seul score cognitif, le score MMSE est utilisé en tant que modalité clinique (voir code fourni par les auteurs, à cette adresse : *https ://github.com/CoBrALab/LSN*). De plus, ce score est le score ayant été utilisé pour créer les deux classes de patients (vérité terrain), ce qui signifie que l'inclure en tant que donnée risque de permettre l'apprentissage par le modèle LSN de la vérité-terrain, donnant ainsi une évaluation des performances très optimiste.

— D'autre part, les IRM ne sont pas utilisées en tant que telles (images en 3D volumique), mais sont d'abord segmentées en 78 régions d'intérêt (ROI) basées sur un atlas, puis pour chaque ROI l'épaisseur corticale est mesurée, et ce sont ces valeurs qui servent de données pour la modalité IRM. Cette étape de calcul d'information est chronophage, mais surtout peut introduire des biais dus à de possibles erreurs de segmentation des zones du cerveau de chaque sujet. En effet, la segmentation 3D d'IRM de cerveau est bien connue pour être un problème difficile en traitement d'images, et les éventuelles erreurs de segmentation peuvent amener à des erreurs de pronostique. De plus, les données d'imagerie sont résumées par un vecteur (1D), ce qui implique une perte d'information tridimensionnelle d'une part, et d'autre part réduit drastiquement la quantité d'information disponible, en remplaçant une image 3D entière par un vecteur de seulement 78 valeurs.

— Les deux branches des réseaux siamois sont réunies par une couche nommée par les auteurs "*distance embedding*", qui consiste en une concaténation des résultats de chaque branche. Il nous a semblé plus cohérent de calculer une différence entre les résultats des deux branches, plutôt que de faire une concaténation, dans la mesure où le but de cette architecture est de modéliser les différences entre la première et la deuxième visite médicale.

— Enfin, il n'y a pas de couches supplémentaires après la fusion des modalités, ce qui ne permet pas de faire un apprentissage conjoint, qui permettrait d'extraire des informations prenant en compte les corrélations entre modalités.

Nous comparerons notre modèle au modèle LSN dans le Chapitre 4, Section 4.

L'autre approche récente que nous avons mentionné est une approche utilisant les RNN. Elle a été publiée [86] en 2019 par Lee *et. al.*, dans le cadre de la prédiction de l'évolution de patients atteints de la maladie d'Alzheimer. Leur modèle, nommé MildInt (voir Figure 2.9), est un réseau multimodal, composé d'un sous-module RNN par modalité. Quatre modalités différentes sont utilisées : mesures calculées à partir des IRM de cerveaux des sujets, scores cognitifs, données démographiques, et dosages de molécules présentes dans le fluide cérébro-spinal. Chaque sous-module est composé d'une simple couche de type *Gated Recurrent Unit* (GRU) [41], et prend en entrée une séquence de données, correspondant à une suite de visites médicales. On notera que pour la modalité IRM, seules les valeurs de la première visite médicale sont utilisée, et non la séquence entière de visite.

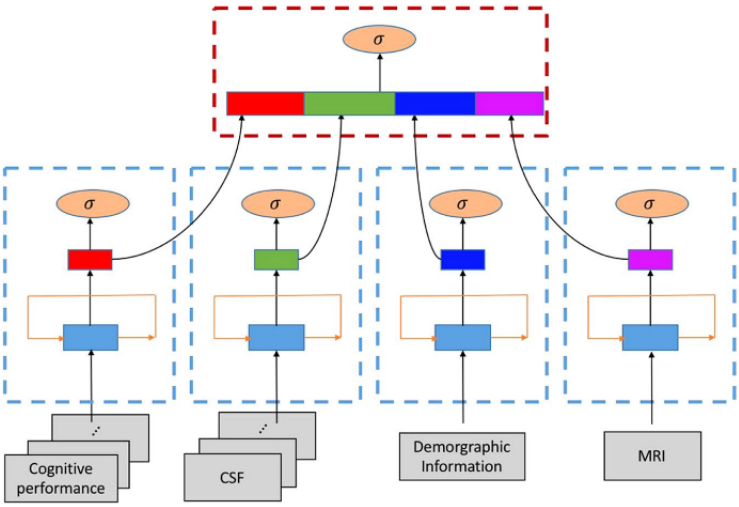


FIGURE 2.9 – Architecture du réseau RNN multimodal MildInt (figure issue de [86])

Dans cet article de Lee *et. al.*, l'entraînement du modèle MildInt est fait en deux temps : tout d'abord chaque sous-module est entraîné indépendamment, puis les poids des sous-modules respectifs sont utilisés pour initialiser le modèle multimodal, qui est ensuite ré-entraîné.

Le jeu de données utilisé est constitué uniquement de sujets de la base ADNI, qui ont été diagnostiqués MCI (stade précédant la maladie d'Alzheimer) à l'inclusion. Ces patients sont répartis en deux classes : ceux qui restent stables, et ceux qui déclinent et sont finalement diagnostiqués Alzheimer. Ainsi, sur un jeu de données de 539 sujets, les auteurs obtiennent une *accuracy* de 79%, et un F1 score de 0.80.

Nous comparerons notre modèle au modèle MildInt dans le Chapitre 5, Section 5.1.

2.5 Stratégies pour l'amélioration de l'apprentissage

Afin de pallier aux problèmes liés aux jeux de données de petite taille, et notamment le sur-apprentissage, voir Section 1.1.4), certains travaux s'orientent vers le transfert d'apprentissage (voir Section 1.3), comme détaillé dans la Table 2.3).

Ref	**Données**	**Tâche**	**Architecture**	**Transfert d'apprentissage**
[59]	ADNI (755 AD, 755 MCI, 755 NC)	Classification supervisée	CNN 3D basé sur un autoencodeur	patchs extraits d'images du jeu de données
[88]	ADNI (755 AD, 755 MCI, 755 NC)	Classification supervisée	CNN 2D basé sur un autoencodeur	patchs extraits d'images du jeu de données, ou patchs extraits d'images naturelles
[51]	30 sujets du *challenge CAD-Dementia* [89] pour le pré-entraînement ; 210 sujets ADNI (70 AD, 70 MCI, 70 NC) pour la tâche finale	Classification supervisée	CNN 3D basé sur un autoencodeur convolutif	autre jeu de données
[90]	280 sujets de la cohorte RUN DMC [91] pour le pré-entraînement ; nombre variable pour la tâche finale	Segmentation	CNN 2D	même patients, mais scans acquis avec un protocole différent
[92]	ADNI (145 AD, 172 NC)	Classification supervisée	CNN 3D basé sur un autoencodeur	patchs extraits d'images du jeu de données

TABLE 2.3 – Stratégies de transfert d'apprentissage utilisées avec des jeux de données d'imagerie cérébrale

La majorité des approches proposées utilisent un auto-encodeur pour l'étape de pré-entraînement (approche auto-supervisée). La difficulté repose ici sur le choix du jeu de données pour le pré-entraînement : il est possible de tirer parti de la grande taille des jeux de données d'images naturelles disponibles en libre accès mais il faut se demander si les caractéristiques de ces images sont assez semblables à celles de l'imagerie cérébrale pour que le transfert d'apprentissage soit efficace.

A l'inverse, choisir des images de cerveau issues d'une étude similaire, supprime ce problème, car les descripteurs extraits par le réseau constituent des bio-marqueurs de la maladie [51], mais il y a un risque que le modèle soit trop spécifique au jeu de données de pré-entraînement.

Une autre approche auto-supervisée est l'utilisation d'une tâche annexe (ou prétexte), en complément de la tâche principale. Par exemple, les auteurs de [93] pré-entraînent un modèle pour la restauration d'images corrompues (plusieurs *patchs* sont intervertis dans chaque image) puis utilisent ce modèle pour des tâches de classification, localisation et segmentation après *fine-tuning* (voir Figure 2.10). Les auteurs de [94] utilisent une méthode similaire, dans laquelle les *patchs* sont intervertis dans l'axe de la profondeur, pour des images en 3D.

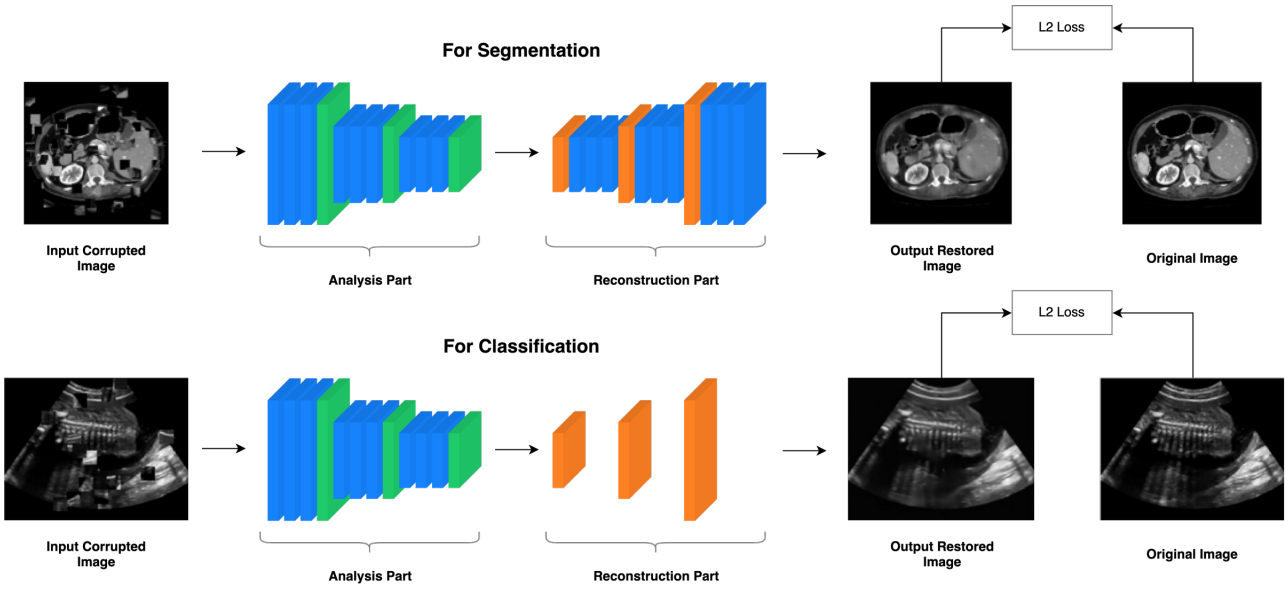


FIGURE 2.10 – Architectures utilisées pour les tâches prétexte de restauration d'images corrompues [93]

Il est également possible d'utiliser le transfert de domaine dans le cas d'images biomédicales [46]. On peut ainsi profiter d'un apprentissage sur une modalité d'imagerie facilement disponible, puis transférer cette connaissance pour une autre modalité, en restant sur des images du même organe. Un exemple est le transfert d'images de poumons obtenues par rayons X vers des images de scanner, pour le diagnostique de Covid-19, en utilisant une technique de *transfer learning* [95]. Le transfert de domaine peut également être fait en restant dans la même modalité, mais en ciblant une autre population de sujets (voir Figure 2.11 pour divers exemples).

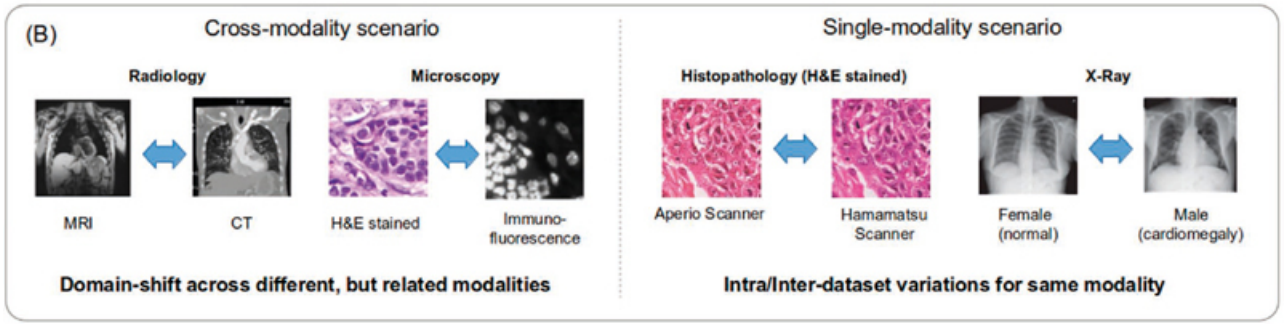


FIGURE 2.11 – Exemples d'application de transfert de domaine avec des données biomédicales [46]

2.6 Synthèse

Dans ce chapitre, nous avons présenté un panel d'architectures et de stratégies pouvant répondre à notre problématique de classification de données bio-médicales multimodales.

Cet état de l'art nous a permis d'orienter nos travaux d'une part vers la fusion intermédiaire des modalités, avec l'utilisation de sous-modules adaptés à chaque modalité, et d'autre part vers l'utilisation de réseaux de neurones convolutifs permettant de traiter des images 3D directement (sans extraction préalable de régions d'intérêt ou projection dans un espace 2D), afin d'éviter au maximum la perte d'information en amont de l'utilisation du modèle.

Dans le chapitre suivant, nous présenterons la mise en place de ces sous-modules, et les choix de type d'opérations à utiliser (convolutions 3D par exemple) pour extraire le maximum d'information de nos données. Nous expliciterons également notre utilisation des réseaux siamois pour la comparaison de l'état de santé des sujets entre deux points de temps.

Deuxième partie

Conception et implémentation de modèles multimodaux pour l'étude de maladies neuro-dégénératives

Chapitre 3

Méthodes proposées : 3DSiameseNet et Multimodal3DSiameseNet, et résultats préliminaires

Dans ce chapitre, nous présenterons dans un premier temps la conception de notre modèle de réseaux de neurones profonds pour la prédiction du déclin cognitif de patients suivis pour la maladie d'Alzheimer. Cette prédiction se fait à partir de deux visites médicales, d'abord en utilisant uniquement leurs IRM structurales de cerveau : modèle **3DSiameseNet** (travaux présentés à la conférence ICPRS 2019 [96]), puis en ajoutant les données cliniques (scores cognitifs et facteurs de risques) comme modalités supplémentaires : modèle **Multimodal3DSiameseNet** (travaux présentés à la conférence IPTA 2020 [97]).

En étudiant l'état de l'art concernant les études de maladies neuro-négénératives en *deep learning*, on peut constater que les réseaux de neurones sont utilisés le plus souvent pour classer les patients en deux classes : malades et sains. Cette classification repose généralement sur l'analyse de données patients (mesures cliniques et/ou imagerie) obtenues lors d'une unique visite médicale. Il est cependant aussi intéressant d'obtenir un pronostique sur l'avancée de la maladie, pour pouvoir identifier les sujets les plus à risque. Notre stratégie a été d'utiliser un réseau de type siamois pour classer les sujets en deux classes (Stable / Déclin sur le long terme), selon l'évolution de leur maladie neuro-dégénérative entre leur inclusion dans l'étude (*baseline*) et une visite de suivi (*follow-up*).

Comme décrit dans l'état de l'art, les réseaux siamois sont souvent utilisés pour classifier des paires d'*inputs* similaires ou différents. Ils peuvent également être considérés comme un outil pour évaluer le degré de différence entre deux entrées, et produire un résultat de classification en fonction de cette différence. Dans notre cas, si l'état d'un patient est sensiblement différent entre une visite et la suivante, nous pourrons donc identifier un déclin cognitif. Nous définissons ainsi cette problématique comme une tâche de classification entre patients stables et patients en déclin cognitif.

3.1 Cas d'application : la maladie d'Alzheimer

Afin de tester notre modèle, nous avons choisi de nous intéresser au cas de la maladie d'Alzheimer. D'un point de vue morphologique, cette pathologie est caractérisée par la destruction de neurones et de synapses dans les régions corticales et sous-corticales (notamment par l'action de deux molécules : le peptide beta-amyloïde et la protéine tau), résultant en une atrophie du lobe temporal, du lobe pariétal, et d'une partie du cortex frontal et du gyrus cingulaire [98] (voir Figure 3.1). Cette dégénérescence conduit à un déclin des facultés cognitives, en particulier concernant des problèmes de mémoire à court-terme, de désorientation, et de comportement [99].

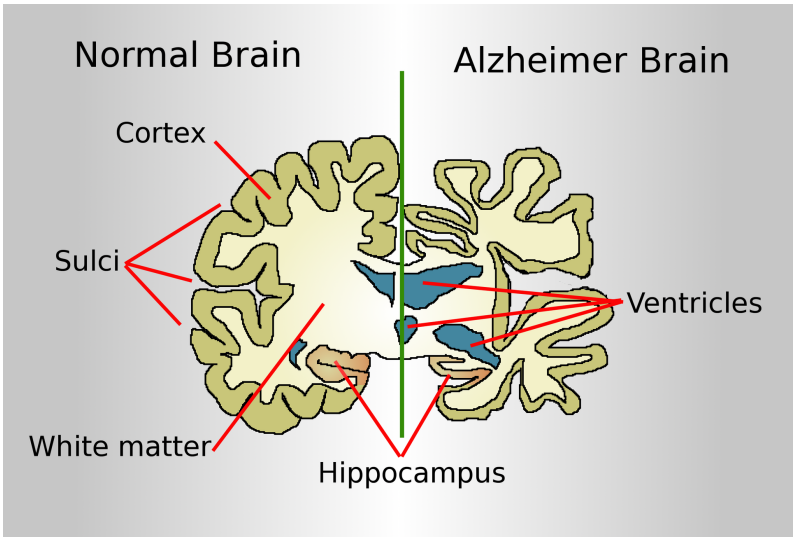


FIGURE 3.1 – Représentation schématique des zones du cerveau affectées par la maladie d'Alzheimer (Source : Wikipédia, auteur : Garrondo, domaine public)

La base de données publique ADNI [5], basée sur une étude américaine multicentrique nommée *Alzheimer's Disease Neuroimaging Initiative*, ayant pour but d'étudier l'évolution de la maladie d'Alzheimer, est un bon cas d'application :

- Maladie neurodégénérative
- Etude longitudinale : visite médicale tous les 6 mois
- Multimodalité : IRM, PET scan, données génétiques, données démographiques, prélèvements et mesures biologiques, scores neurocognitifs
- Larges cohortes : 800 sujets pour la cohorte originale (ADNI-1)
- Très souvent utilisée dans l'état de l'art

Dans cette base de données, les individus ont un diagnostique médical tous les six mois : maladie d'Alzheimer (*Alzheimer's Disease* : AD), trouble cognitif léger (*Mild Cognitive Impairement* : MCI), ou témoin sain (*Normal Control* : NC). La cohorte ADNI-1 est composée de 200 sujets NC, 400 sujets MCI, et 200 sujets AD. Ces diagnostiques sont ponctuels et ne donnent pas d'éléments sur l'évolution de la maladie chez ces patients (par exemple, un patient peut avoir connu un déclin cognitif au cours de l'étude, et pourtant rester dans la classe MCI durant toute la durée de l'étude).

3.2 Qualification de l'évolution de la maladie à partir des IRM

Dans cette section, nous allons présenter notre premier modèle, appelé 3DSiameseNet, qui est un réseau siamois utilisant des convolutions 3D pour la prédiction du déclin cognitif à partir de deux IRM d'un même sujet. Ces travaux ont été présentés à la conférence ICPRS 2019 [96], et le code associé est disponible à cette adresse : *https ://github.com/CeciliaOstertag/3D-SiameseNet*).

3.2.1 Étiquetage des données

Dans la majorité de l'état de l'art les réseaux de neurones sont utilisés pour diagnostiquer les patients selon les trois groupes classiques (AD : maladie d'Alzheimer, MCI : déficit cognitif léger, et NC : contrôles). Cette classification repose sur l'analyse des données patient (mesures cliniques ou imagerie) obtenues lors d'une unique visite médicale.

Afin de disposer d'une vérité-terrain concernant l'évolution dans le temps des patients ("Stable" ou "Déclin"), et pouvoir comparer notre modèle au modèle de référence proposé par Bhagwat *et. al.* [85] (*Longitudinal Siamese Network* : LSN), nous avons choisi de reprendre la méthode utilisée par Bhagwat *et. al.* pour étiqueter chaque patient.

Plus précisément, l'indice clinique *Mini Mental State Evaluation* (MMSE) est utilisé sur une partie des patients pour un *clustering* hiérarchique avec l'algorithme de Classification Ascendante Hiérarchique utilisant la distance euclidienne et la méthode de Ward. Ce score de MMSE varie entre 30 (valeur maximale, attendue pour un individu sain), et 0 (valeur minimale).

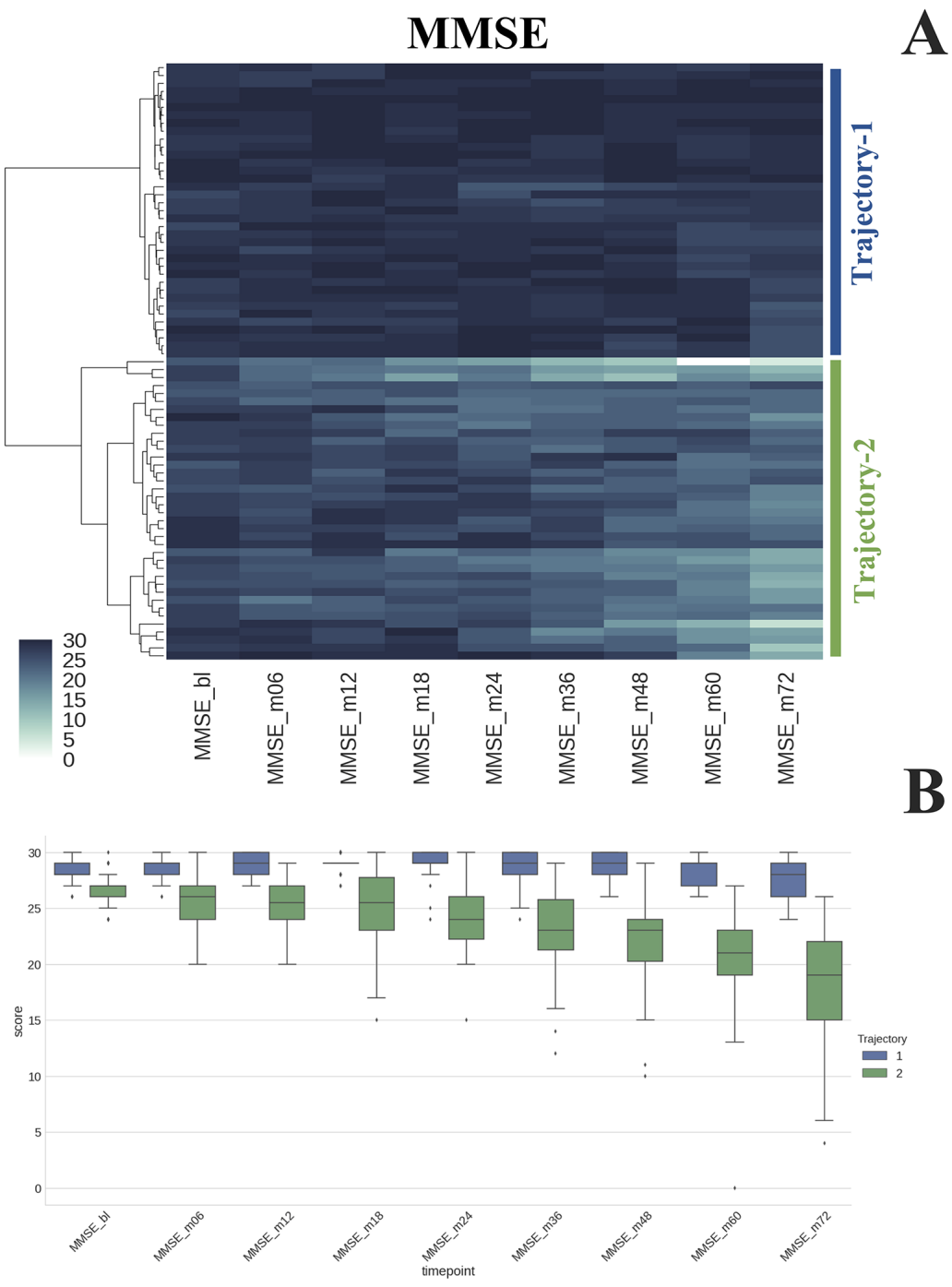


FIGURE 3.2 – Groupement des sujets ADNI en fonction de l'évolution du score MMSE au cours du temps (figure issue de Bhagwat et. al [85]). A : résultats du *clustering* (une ligne = 1 sujet) ; B : trajectoires des deux *clusters*). Sujets stables en bleu, sujets en déclin en vert.

Deux *clusters* sont identifiables (Figure 3.2 A ci-avant), en fonction de la rapidité avec laquelle les capacités cognitives des sujets varient. Les trajectoires de ces *clusters* (Figure 3.2) B) sont ensuite utilisées pour labelliser le reste des sujets, en fonction de la distance euclidienne de la MMSE des sujets à la MMSE moyenne des *clusters*, en se basant sur au moins trois visites médicales sur une durée de 72 mois. Nous obtenons donc deux classes : Stable et Déclin. Le tableau suivant montre la répartition des sujets dans ces deux classes, en fonction du diagnostique reçu lors de la visite d'inclusion :

	AD	NC	MCI	Total
Stable	2	213	153	368
Déclin	159	2	211	372

TABLE 3.1 – Répartition des sujets ADNI dans les classes Stable et Déclin, créées d'après l'évolution du score MMSE, en fonction du diagnostique reçu lors de la visite d'inclusion

On voit que, si les sujet souffrant d'Alzheimer ("AD") à l'inclusion appartiennent essentiellement à la classe "Déclin", et que les sujets sains ("NC") à l'inclusion appartiennent généralement à la classe "Stable", la population des sujets souffrant d'un handicap cognitif léger à l'inclusion ("MCI") peuvent soit rester stables, soit connaître un déclin cognitif lors de l'étude.

3.2.2 Notre modèle : 3DSiameseNet

Le modèle que nous proposons est un réseau siamois, comme le modèle LSN de Bhagwat *et. al.* [85] (voir Etat de l'art, Section 2.4), qui prend en entrée les données de deux visites médicales d'un même sujet. Contrairement au LSN, qui utilise des mesures issues des IRM originelles, calculées après segmentation selon un atlas de cerveau humain, notre modèle prend en entrée les IRM en tant qu'images en 3D volumique, et utilise des convolutions 3D pour calculer des *features* représentant l'évolution de l'état de santé du sujet, pour enfin prédire son déclin cognitif. L'architecture de notre modèle est détaillée dans la Figure 3.4, en page 70.

Utilisation d'opérations 3D

Le risque d'introduction d'erreurs est important si l'on passe par une étape de segmentation du cerveau en zones d'intérêt. Nous voulons donc créer une architecture capable de traiter les IRM de cerveau, non seulement en 3D, mais aussi en traitant l'image directement à partir de l'acquisition.

Dans le modèle que nous proposons, chaque branche du réseau siamois est un réseau de neurones convolutif utilisant des filtres 3D. Nous avons décidé de ne pas utiliser une approche par *patchs*, qui conduit également à une perte d'information, mais de traiter les images IRM entières. Ainsi, chaque filtre de convolution 3D opère sur les pixels provenant de *slices* successives, dans les trois axes, du cerveau du sujet. Les opérations de convolution

3D nécessitant des ressources mémoire importantes, nous avons dû faire des choix lors de l'implémentation de notre modèle, en terme de taille de *kernels* d'une part, et de nombre de couches de convolution d'autre part :

— Nous avons utilisé des filtres de taille $3 \times 3 \times 3$ pixels pour les opérations de convolution, avec un pas de 1 pixel dans les trois dimensions, avec l'hypothèse qu'un *kernel* de petite taille permettrait de prendre en compte les changements morphologiques subtils chez les patients.

— Pour les opérations de *pooling*, nous utilisons des filtres de taille $2 \times 2 \times 2$, avec un pas de 2 pixels dans les trois dimensions. Cette réduction de dimension par *pooling* est faite en prenant la moyenne des valeurs pour éviter un biais vers la détection de contours, comme cela peut être le cas en prenant le maximum des valeurs.

— Nous utilisons un faible nombre de filtres (16 ou 32) dans les couches de convolution.

Nous avons ainsi construit un réseau convolutif comme une succession de "blocs de convolution" (voir Figure 3.3). Chaque bloc est composé d'une couche de convolution, d'une couche de *pooling*, d'une couche de normalisation par *batch*, et d'une couche d'activation de type *Leaky ReLU*. Pour chaque bloc, le nombre de filtres de la couche de convolution est variable.

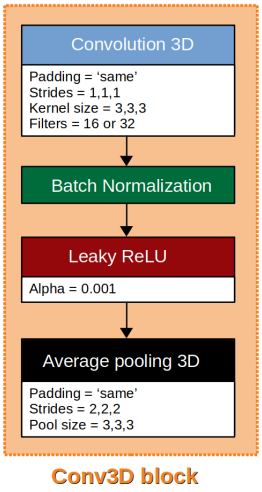


FIGURE 3.3 – Bloc de convolution élémentaire : 1 couche Conv 3D, 1 couche de normalisation par *batch*, 1 *Leaky ReLU* , 1 couche de *pooling* par moyennage en 3D.

Dans cette première version de notre architecture, appelée 3DSiameseNet, ce bloc entier est répété trois fois, suivi d'une quatrième sans la couche de *pooling*. L'ensemble de ces quatre blocs constitue chacune des branches du réseau siamois.

Réunion des deux branches siamoises

Pour rassembler les deux branches siamoises, nous avons choisi de calculer la différence entre les deux sorties, afin de mieux représenter l'évolution entre les deux entrées du réseau qu'en utilisant une concaténation. Ce résultat, stocké dans une matrice 3D, est aplati (*flattened*) dans un vecteur à une dimension.

Ce vecteur 1D passe ensuite successivement par trois couches *Fully Connected*, avec une activation de type *Leaky ReLU* pour les deux premières, et une Softmax pour la couche finale. Un *Dropout*, qui supprime aléatoirement 50% des entrées, est appliqué avant cette dernière couche, pour réduire le sur-apprentissage. Chaque couche *Fully Connected* est suivie d'une normalisation par *batch*, comme pour les blocs de convolution.

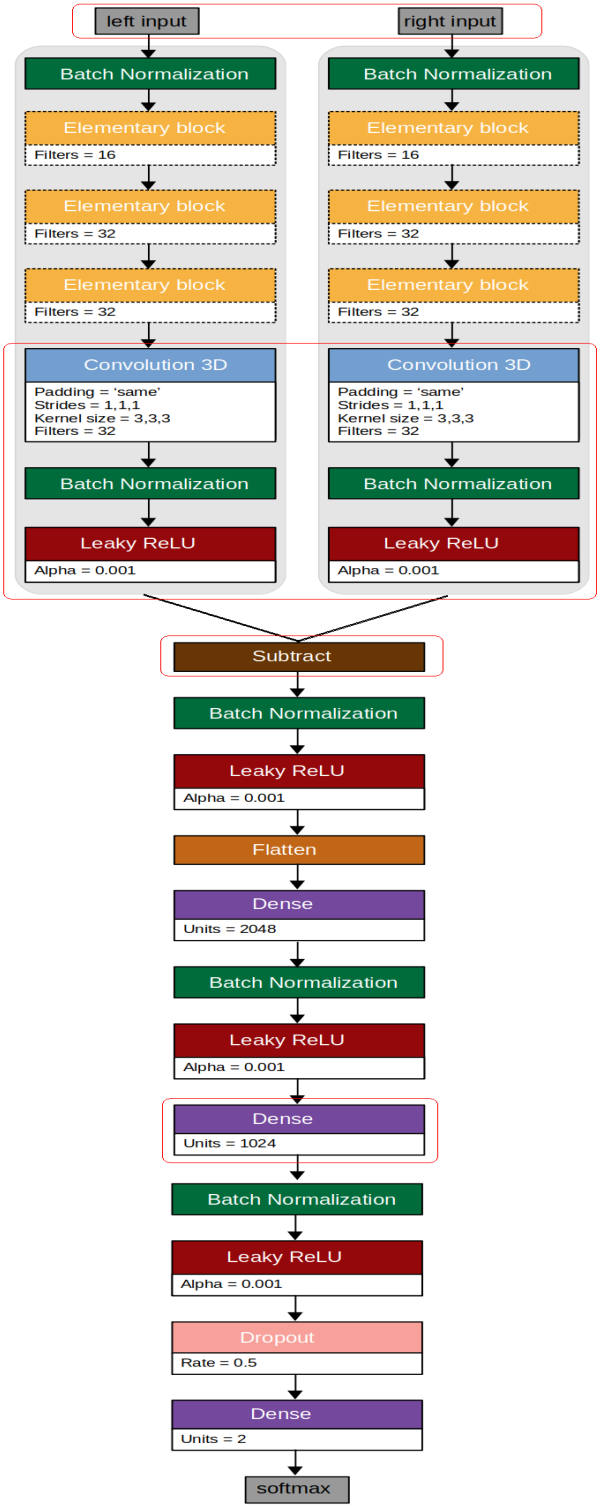


FIGURE 3.4 – Architecture du réseau 3DSiameseNet, prenant une paire de données, correspondant à deux visites médicales d'un même patient, en entrée de deux branches siamoises, puis réunissant les *features* représentant l'évolution de ce patient, afin de prédire si l'état du patient est ou non entrain de décliner. Les couches du réseau encadrées en rouge sur le schéma seront étudiées plus en détail dans la section 3.2.4.

3.2.3 Résultats et analyse

Etant donné que notre jeu de données est de petite taille, nous avons fait un *bootstrap* en répétant le test 10 fois, avec un échantillonnage aléatoire de nos données entre entraînement et validation à chaque fois. Après 600 *epochs*, la fonction de coût utilisée pour entraîner notre modèle (*cross-entropy*) atteint un plateau, avec une valeur moyenne de 0.624 pour l'entraînement et de 0.992 pour la validation (Figure 3.5).

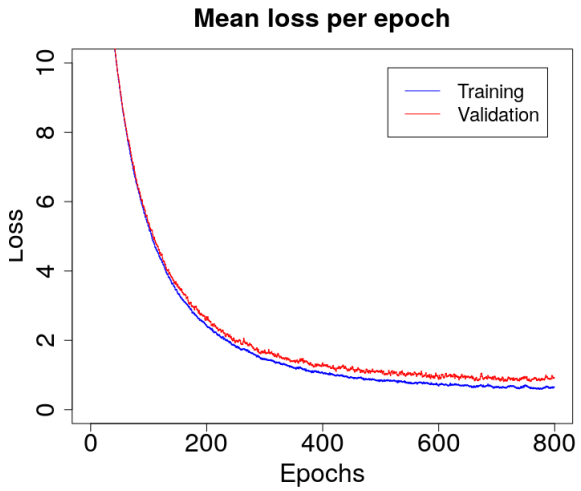


FIGURE 3.5 – Evolution de la fonction de coût pour le jeu de données d'entraînement et de test, pendant 800 *epochs*. Chaque point représente la moyenne des valeurs obtenues sur 10 échantillonnages aléatoires des données

Après l'entraînement, notre modèle atteint une *accuracy* moyenne de 90%. Pour comparer nos résultats obtenus avec 3DSiameseNet aux résultats de l'état de l'art, nous avons repris les éléments de comparaison donnés par Bhagwat *et. al.* Dans leur article, les auteurs ont entraîné leur modèle (LSN) avec un jeu de données de 1116 sujets, et comparé leurs résultats avec quatre modèles : une régression logistique (LR), une machine à vecteur de support (SVM), une forêt d'arbres aléatoires (RF), et un autre réseau de neurones (ANN).

Ces quatre modèles utilisent uniquement les données d'épaisseur corticale des régions d'intérêt du cerveau (données d'inclusion et de suivi), alors que le modèle LSN utilise à la fois les données d'imagerie (épaisseur corticale) et des données cliniques (score MMSE). D'après ces résultats (Table 3.2), notre modèle est également supérieur à l'état de l'art. Il est cependant difficile de comparer notre modèle au LSN, étant donné qu'à ce stade nous n'utilisons que des données d'imagerie.

	Données d'imagerie					Données d'imagerie + cliniques
	LR	**SVM**	**RF**	**ANN**	**3DSiameseNet**	**LSN**
Accuracy	77%	76%	76%	75%	90%	94%

TABLE 3.2 – Comparaison de l'*accuracy* de six modèles : régression logistique (LR), machine à vecteur de support (SVM), forêt d'arbres aléatoires (RF), réseau de neurones (ANN), Longitudinal Siamese Network (LSN), et notre 3DSiameseNet. Données de Bhagwat *et. al.* (Supplementary material). [85]

3.2.4 Capacités discriminantes de notre modèle

Afin de proposer une vue des différentes étapes d'extraction de caractéristiques à partir de paires d'IRM, nous avons utilisé l'algorithme t-SNE [52] pour visualiser les *feature maps* de 40 images de validation, à trois étapes différentes du réseau. Nous avons choisi les couches intermédiaires encadrées en rouge dans la Figure 3.4, page 70 :

— Dernière couche de chaque branche avant réunion des deux branches

— Couche de soustraction pour la réunion des branches

— Avant dernière couche *fully-connected*

Pour comparaison, nous avons aussi affiché les résultats obtenus à partir des images brutes (tableaux de pixels) aplaties et concaténées.

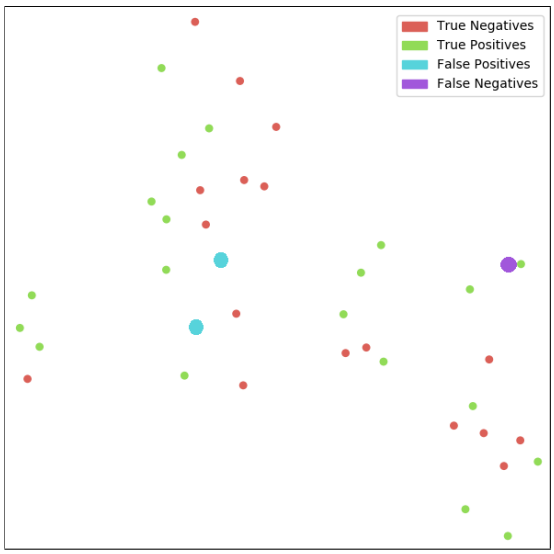


FIGURE 3.6 – Images d'entrée (inclusion et suivi) (concaténation des deux images)

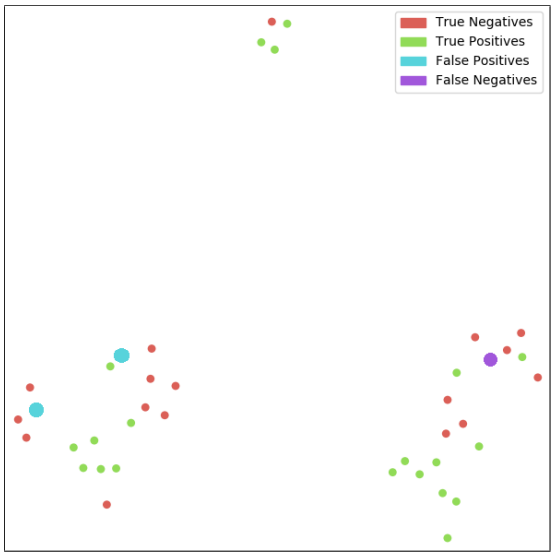


FIGURE 3.7 – Dernière couche de chaque branche avant fusion (concaténation des deux résultats)

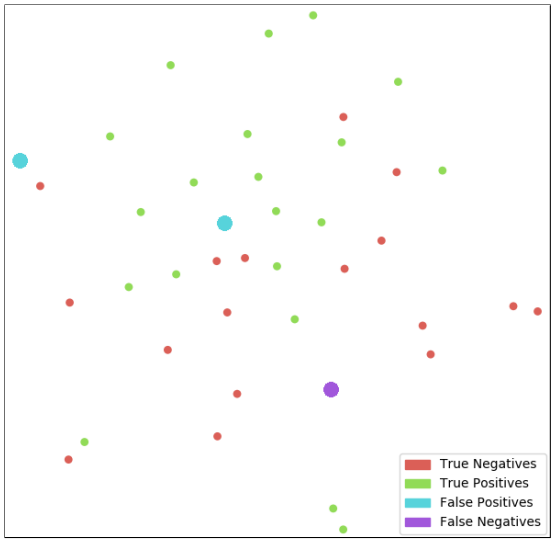


FIGURE 3.8 – Couche de fusion par soustraction

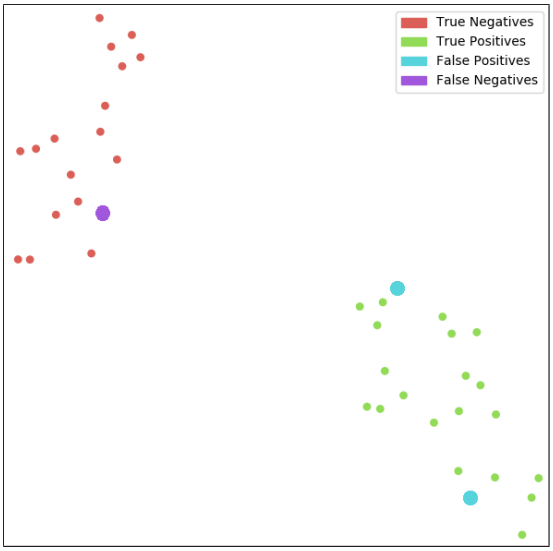


FIGURE 3.9 – Avant dernière couche *fully-connected*

Les Figures 3.6 à 3.9 représentent les résultats donnés par l'algorithme t-SNE. Dans cette projection en deux dimensions, chaque point représente les *feature maps* calculées à partir d'une image d'origine. Les données similaires sont proches, et les données dissimilaires sont éloignées. La classe positive est la classe Déclin, et la classe négative est la classe Stable. Bien sûr, l'étiquetage des vrais positifs, vrais négatifs, faux positifs et faux négatifs est obtenue en sortie du réseau, bien que visualisé sur les couches intermédiaires. Pour une meilleure visibilité, les faux positifs et faux négatifs (erreurs de prédiction) sont représentés par des points plus larges.

Lorsque l'on considère nos données d'entrée (Figure 3.7), nous voyons que leur distribution est assez hétérogène dans l'espace des *features*, donc que les données ne sont pas initialement bien séparées, ce qui est intuitivement évident. Une fois passé par les deux branches siamoises, nous pouvons observer la formation de trois *clusters* (Figure 3.7), qui ne sont pas liés à notre distinction entre Stable et Déclin. Il est possible que ces groupes soient liés à d'autres caractéristiques importantes présentes dans les IRM, par exemple la taille du cerveau. En revanche, après la fusion par soustraction (Figure 3.8), ces *clusters* ne sont plus identifiables. Nous avons donc émis l'hypothèse que l'opération de soustraction a eu pour effet de supprimer des *features* indésirables. Grâce à ce résultat, nous avons validé le choix de la fusion par soustraction plutôt que par simple concaténation comme dans le modèle LSN. Enfin, après l'avant dernière couche *dense* (Figure 3.9), les données sont bien groupées selon nos deux classes Stable et Déclin.

3.3 Utilisation de deux modalités : IRM et données cliniques

Une fois validés l'utilisation du modèle siamois pour l'étude de l'évolution de la maladie d'Alzheimer entre deux pas de temps, et l'utilisation des opérations de convolution et de *pooling* 3D pour prendre en compte la totalité des informations morphologiques du cerveau des sujets, nous avons proposé un réseau multimodal permettant d'utiliser les données cliniques en plus des données IRM. Pour l'analyse de l'évolution de l'état du patient au travers de ses données cliniques, nous allons également utiliser l'approche siamoise. Ces travaux ont été présentés à la conférence IPTA 2020 [97], et le code associé est disponible à cette adresse : *https ://github.com/CeciliaOstertag/MultiNet*).

3.3.1 Gestion des modalités en sous-modules du réseau

Cette nouvelle architecture, nommée Multimodal3DSiameseNet, est un réseau de neurones profonds composé de sous-modules spécifiques à chaque modalité.

Données IRM

Les IRM structurales de cerveau sont envoyées à un réseau similaire à notre 3DSiameseNet (voir Figure 3.10). Ce réseau a été adapté, d'une part en retirant les dernières couches, i.e. *dropout* et *softmax*, pour récupérer un vecteur de *features* décrivant l'évolution morphologique du cerveau du sujet. D'autre part, le nombre de couches ainsi que le nombre de filtres par couche a été réduit.

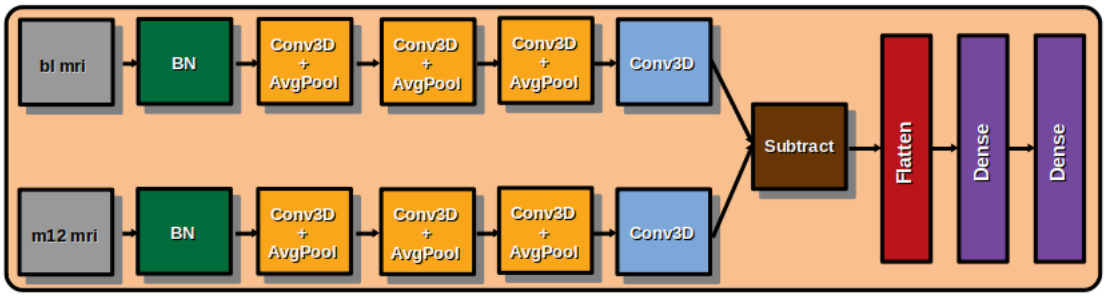


FIGURE 3.10 – Architecture du sous-module pour les données IRM, prenant en entrée deux visites, "bl" :*Baseline* (inclusion du patient) et "m12" : suivi à 12 mois

Données cliniques

Les données cliniques à disposition dans la base ADNI sont de deux types : d'une part celles qui ne varient pas dans le temps (sexe biologique ou génotype par exemple) ou qui varient indépendamment de la maladie (âge par exemple), et d'autre part celles qui vont

être modifiées par les effets de la maladie (scores cognitifs par exemple). Etant donné leurs spécificités, ces deux types de données ne seront pas traitées de la même façon dans notre modèle :

— Les données cliniques susceptibles de varier durant le temps de l'étude (mesures biologiques et scores neuro-cognitifs) sont analysées par un réseau siamois de type *feed forward* à une dimension (voir Figure 3.11). A la fin de ce réseau siamois, les deux branches sont rassemblées en prenant la valeur absolue de la soustraction des deux résultats, de la même façon qu'avec les IRM.

— Les autres données (informations démographiques et génétiques) sont directement envoyées dans un réseau composé de couches *fully connected* (voir Figure 3.12).

Ces deux *outputs*, qui concernent toujours uniquement les données cliniques, sont ensuite rassemblées en un seul vecteur par une opération de concaténation (voir Figure 3.13).

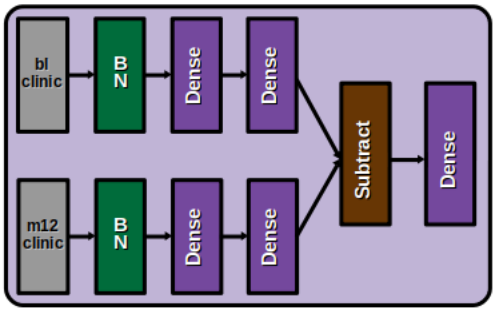


FIGURE 3.11 – Architecture du sous-module pour les données cliniques, prenant en entrée deux visites, "bl" :*Baseline* (inclusion du patient) et "m12" : suivi à 12 mois

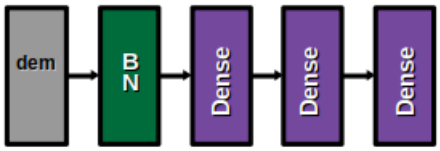


FIGURE 3.12 – Architecture du sous-module pour les facteurs de risque, prenant en entrée les données de la visite i

Fusion des modalités

Pour finir, les deux modalités seront fusionnées par concaténation des vecteurs de *features* suivie de plusieurs couches *fully connected*. Cette stratégie de fusion intermédiaire est assez similaire à celle utilisée dans l'approche proposée par Xu *et. al.* [83]. Les couches *fully connected* de cette dernière partie du réseau multimodal (nommées "*joint-features layers*" par Xu *et. al.*) ont pour but d'extraire des informations concernant de possibles corrélations entre les modalités. Une couche de *dropout* pourra être ajoutée avant la couche d'activation finale, utilisant une fonction d'activation adaptée aux problèmes de classification binaire (sigmoïde par exemple). La figure ci-dessous représente l'architecture de ce modèle (Figure 3.13).

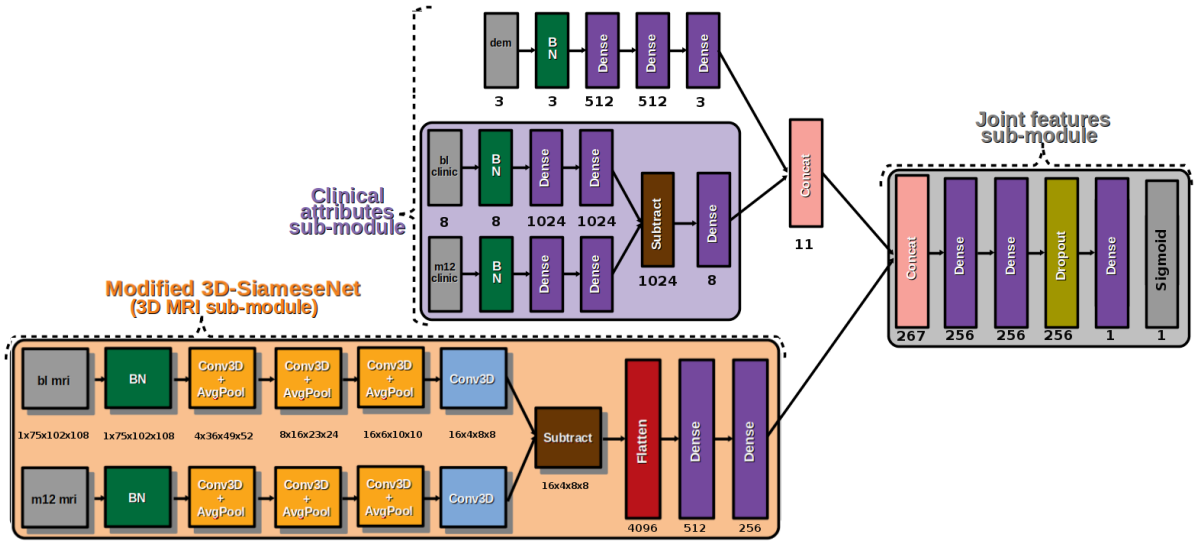


FIGURE 3.13 – Architecture de notre réseau multimodal 3DMultimodalSiameseNet, prenant en entrée deux visites, "bl" :*Baseline* (inclusion du patient) et "m12" : suivi à 12 mois

3.3.2 Apports de la multimodalité

Pour évaluer la valeur ajoutée de l'apprentissage multimodal, nous avons comparé deux architectures. La première est un réseau siamois prenant en entrée uniquement les variables cliniques sélectionnées (dénommé ClinicalSiameseNet), pour la visite d'inclusion et pour une visite de suivi à 12 mois, ou à 6 mois si la visite de 12 mois n'est pas disponible. En d'autres termes, ce réseau est un classifieur créé à partir du sous-module "données cliniques", avec l'addition d'une couche de *dropout* et d'une couche *fully-connected* avec une fonction d'activation de type sigmoïde. La deuxième architecture est notre réseau multimodal Multimodal3DSiameseNet.

Pour cette analyse, nous avons utilisé les sujets ADNI-1 possédant l'IRM d'inclusion, et au choix l'IRM de 6 mois ou de 12 mois. Nous avons ainsi obtenu un jeu de donné de 377 sujets (191 stables, et 186 en déclin cognitif). Les deux modèles ont été entraînés durant 75 *epochs*, avec une validation croisée à 4 échantillons, tout en ayant préalablement laissé de côté 57 sujets (29 "Stable" and 28 "Déclin") pour la phase de test. Nous avons utilisé les mêmes données et le même protocole, et fixé la valeur de *seed* du tirage aléatoire utilisé pour l'ensemble des procédures de l'implémentation, afin d'avoir des résultats reproductibles. Les variables cliniques que nous avons choisi sont les suivantes : âge, sexe biologique, génotype APOE4, ainsi que 8 scores neuro-cognitifs : LDELTOTAL, RAVLT learning, RAVLT immediate, CDRSB, FAQ, TRABSCOR, RAVLT forgetting, et DIGITSCOR (Table 6.1 en Annexe A pour plus de détails).

Après l'entraînement, les 57 sujets du jeu de test sont utilisés pour comparer les deux architectures : aire sous la courbe ROC (*Area Under ROC curve (AUC)*), précision, rappel (*recall*), score F1, et exactitude (*accuracy*). Nous avons ici considéré la classe Déclin comme classe positive, et la classe Stable comme classe négative.

Modèle	**Pas de temps**	**Acc**	**Pre**	**Rappel**	**AUC**	**F1**
ClinicalSiameseNet	bl + m06/12	0.899	0.822	0.92	0.968	0.899
Multimodal3DSiameseNet	bl + m06/12	0.925	**0.924**	**0.929**	**0.978**	**0.925**

TABLE 3.3 – *Accuracy* (Acc), Précision (Pre), Rappel , AUC, and F1-score, obtenus avec notre réseau clinique et notre réseau multimodal (320 sujets d'entraînement, 57 sujets de test)

Les résultats de la Table 3.3 montrent que le réseau multimodal est un meilleur classifieur que le réseau clinique, quelle que soit la métrique utilisée pour la comparaison. Il est donc profitable d'apporter des informations morphologiques, au travers des données d'imagerie médicale, pour la détection du déclin cognitif chez les sujets de notre jeu de données.

3.3.3 Robustesse aux données manquantes

Comme nous l'avons expliqué précédemment, l'existence de données manquantes est une caractéristique souvent rencontrée lors des analyses d'essais cliniques. Lors de l'analyse des données, la stratégie la plus simple est de ne garder que les sujets pour qui les données sont complètes. Une autre stratégie est de remplacer les valeurs manquantes. Cependant peu de travaux s'attachent à aborder le problème du point de vue de l'architecture des modèles de réseaux de neurones. Dans [100], la couche d'entrée du modèle est modifiée pour pouvoir modéliser les valeurs manquantes par une fonction de densité. Nous avons donc cherché à savoir si l'addition de la modalité IRM permet de compenser la présence aléatoire de données manquantes [1] dans les données cliniques.

Nous avons réutilisé nos 57 sujets du jeu de test pour tracer la courbe ROC de nos deux modèles, en faisant varier la proportion de données manquantes dans les données cliniques (excepté âge et sexe biologique) des sujets. Ces données manquantes ont été créées en supprimant aléatoirement des valeurs, avec une proportion allant de 12.5% (1 variable sur 8) à 37.5% (3 variables sur 8) des variables cliniques, dans les données d'inclusion et de visite de suivi. Ces résultats (Figure 3.3.3 ci-après) montrent que **la baisse de performance du modèle clinique est plus rapide que celle du modèle multimodal**.

1. Ici nous considérons uniquement les données manquantes de type "entièrement aléatoires", telles que définies par Little & Rubin [61].

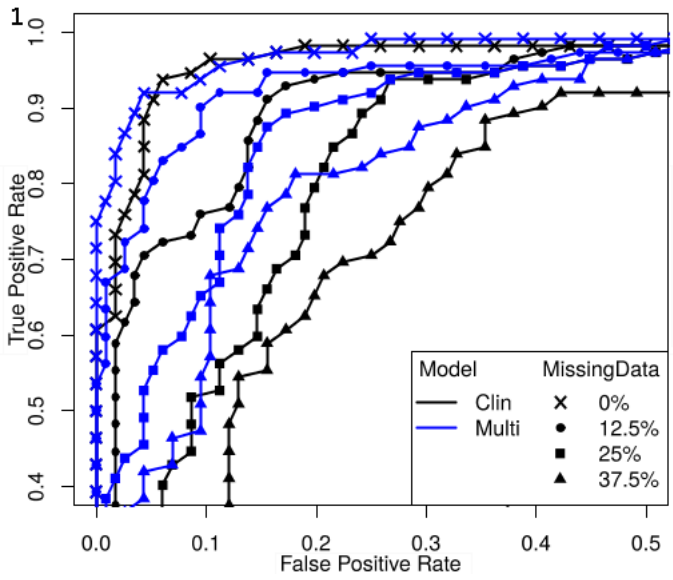


FIGURE 3.14 – Courbes ROC de notre modèle clinique et de notre modèle multimodal, pour une proportion de données manquantes allant de 0 à 37.5% du nombre de variables cliniques (excepté genre et sexe biologique)

Nous avons également comparé ces performances à celles obtenues par des approches de *machine learning* plus classiques : machine à vecteur de support (SVM) avec noyau linéaire ou noyau RBF, forêt d'arbres aléatoires (RF), *Multi Layer Perceptron* (MLP), et Adaboost [101]. Tous ces modèles de l'état de l'art ont été entraînés avec uniquement des données cliniques.

D'après les valeurs d'AUC (Table 3.4) que nous obtenons, nos deux modèles ont les meilleures performances. Nous pouvons également constater que le modèle multimodal a une AUC de 0.873 avec 37.5% de données manquantes, ce qui est supérieur à l'AUC des modèles « traditionnels » (SVM, RF, MLP, Adaboost), quel que soit le pourcentage de données manquantes entre 0% et 37.5%). Un test de Kruskal-Wallis sur les AUC a montré une différence statistiquement significative entre le modèle clinique et le modèle multimodal, en faveur du modèle multimodal, pour 37.5% de données manquantes (p-value = $0.043 < 0.05$).

% NA	Clin	Lin.SVM	Rbf.SVM	RF	MLP	Adaboost	Multi
0%	0.968	0.860	0.856	0.867	0.863	0.865	**0.978**
12.5%	0.931	0.830	0.803	0.840	0.828	0.843	**0.951**
25%	0.876	0.770	0.742	0.783	0.769	0.786	**0.916**
37.5%	0.807	0.728	0.704	0.738	0.727	0.732	**0.873**

TABLE 3.4 – Valeurs AUC de nos deux modèles ("Clin" et "Multi") et de 5 modèles de référence, pour une proportion de données manquantes allant de 0 à 37.5% du nombre de variables cliniques (excepté âge et sexe biologique)

3.3.4 Apprentissage du cas des données manquantes à travers l'entraînement du modèle

Etant donné que la capacité de généralisation d'un modèle d'apprentissage profond dépend de la diversité présente dans les données d'entraînement, nous avons ensuite cherché à savoir quel serait le comportement du modèle multimodal si on introduisait des données manquantes dans les données d'apprentissage.

Pour cela, nous avons entraîné de nouveau notre réseau multimodal, en réinitialisant tous les poids. Durant l'entraînement, certaines données sont supprimées de façon aléatoire et remplacées par 0, et ce pour les deux visites. A chaque *epoch* et pour chaque patient, nous avons donné une probabilité de 10% pour que les variables cliniques soient affectées par la destruction de données. Lorsque c'est le cas, deux variables choisies aléatoirement dans les données d'inclusion, et deux variables choisies aléatoirement dans les données de suivi sont supprimées pour simuler une valeur manquante.

Les résultats obtenus pour cette expérience ("MultiM") sont présentés dans la Figure 3.15 et dans la Table 3.5.

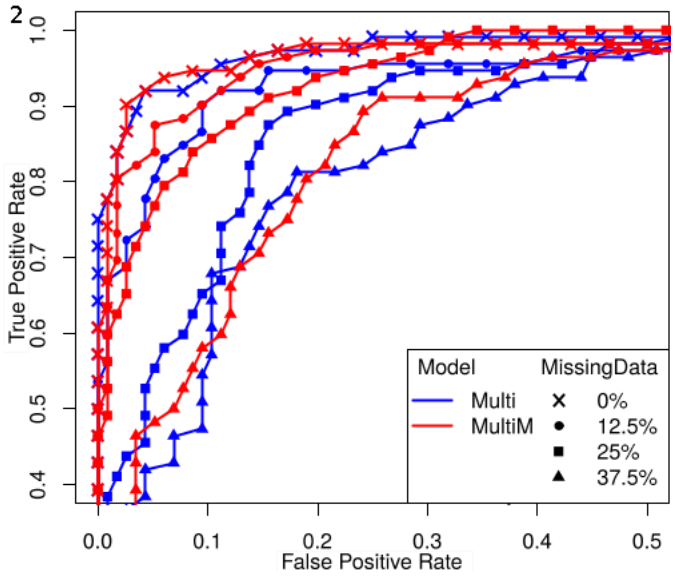


FIGURE 3.15 – Courbes ROC de notre modèle multimodal, entraîné avec données manquantes ("MultiM") ou sans ("Multi"), pour une proportion de données manquantes allant de 0 à 37.5% du nombre de variables cliniques (excepté genre et sexe biologique)

Ces résultats montrent que ce nouveau modèle entraîné est alors plus robuste aux données manquantes lors de l'inférence, que celui n'ayant pas été entraîné avec des données incomplètes. Nous pouvons en déduire que **ce modèle a "appris" à gérer la présence de valeurs manquantes dans les données**. Des tests de Kruskal-Wallis sur les résultats de la

Table 3.5 ont montré des différences significatives, sur l'ensemble des proportions de données manquantes, pour le modèle clinique (p-value = $0.0037 < 0.005$) et pour le modèle multimodal entraîné sur des données complètes (p-value = $0.0338 < 0.05$), mais pas pour le modèle multimodal entraîné avec des données manquantes (p-value = $0.0637 > 0.05$).

Cela signifie que la présence de données manquantes, dans la limite de 37.5%, n'a pas d'influence significative sur la performance du réseau "MultiM", donc que celui-ci est **robuste aux données manquantes**.

% NA	Multi	MultiM
0%	**0.978**	0.972
12.5%	0.951	**0.964**
25%	0.916	**0.958**
37.5%	0.873	**0.887**

TABLE 3.5 – Valeurs AUC de notre modèle multimodal Multimodal3DSiameseNet, entraîné avec données manquantes ("MultiM") ou sans ("Multi), pour une proportion de données manquantes allant de 0 à 37.5% du nombre de variables cliniques (excepté âge et sexe biologique)

3.4 Synthèse

Dans cette partie nous avons d'abord proposé un modèle appelé 3DSiameseNet, conçu pour prendre en entrée des images 3D entières d'IRM de cerveau. Une architecture siamoise est utilisée pour mesurer l'évolution morphologique entre une première visite médicale et une visite de suivi. Ce modèle peut donc être utilisé avec des IRM de cerveau, sans passer préalablement par une segmentation basée sur un atlas.

Nous avons ensuite ajouté une seconde modalité : les données cliniques, séparées en "facteurs de risque" et "scores cognitifs". Nous avons ainsi montré que l'utilisation des deux modalités donne de meilleurs résultats sur notre jeu de données. Nous avons aussi montré que la présence des données image rend le modèle plus robuste aux données manquantes qui peuvent survenir lors de l'acquisition ou de la saisie des données cliniques. De plus, nous avons structuré l'architecture de notre réseau en sous-modules, ce qui rend possible l'ajout d'une ou plusieurs modalités supplémentaires grâce à de nouveaux sous-modules, et ainsi facilite l'adaptation de notre modèle à de nouvelles applications.

Dans la suite de ce travail, nous allons étendre l'application de notre réseau à l'utilisation de paires de visites choisies arbitrairement, et avec un intervalle arbitraire entre ces deux visites, et nous allons analyser plus en détail les erreurs de classification commises par notre modèle.

Chapitre 4

Réseau de neurones profond multimodal pour la prédiction du déclin cognitif : Multimodal3DSiameseNet

Dans ce chapitre, nous décrirons l'architecture définitive de notre modèle, et nous analyserons les résultats de classification obtenus sur un large jeu de données de sujets suivis dans le cadre de l'étude de la maladie d'Alzheimer.

Dans ce jeu de données étendu, nous utiliserons comme entrée des visites choisies arbitrairement entre l'inclusion et le suivi à 24 mois, et avec un intervalle de temps variable entre les deux visites. Cela permet à la fois d'avoir une base d'entraînement plus large, et d'entraîner notre modèle à prédire le déclin à long terme (après 72 mois) à partir de deux visites prises aléatoirement dans les 24 premiers mois de l'étude. Particulièrement dans le cadre d'études cliniques longitudinales sur les maladies neuro-dégénératives où les patients, du fait de leur état, sont particulièrement susceptibles de manquer quelques visites, le fait de ne pas avoir à imposer de pas de temps fixé entre les deux visites considérées par le modèle est une avancée non négligeable par rapport au protocole expérimental présenté dans le Chapitre 3. Lors de l'analyse de ces résultats, nous nous attacherons en particulier à trouver des hypothèses permettant d'expliquer les erreurs de classification de notre modèle.

Enfin, nous présenterons dans ce chapitre une preuve de concept de la grande facilité avec laquelle ce modèle peut être adapté à la prédiction du déclin cognitif dans le cadre d'autres maladies neuro-dégénératives. Ici, nous utiliserons un jeu de données plus petit, de sujets suivis pour la maladie de Parkinson, et nous utiliserons un simple transfert d'apprentissage pour l'utilisation de notre modèle pour la prédiction du déclin cognitif dans le cas de cette autre pathologie. Ainsi, nous montrerons que l'adaptabilité de notre modèle Multimodal3DSiameseNet nous permet de le réutiliser dans le cadre d'autres maladies neuro-dégénératives, dont les jeux de données sont plus petits que notre jeu de données issu de la base ADNI. Cela est particulièrement important dans notre contexte d'études médicales, car nos potentiels jeux de données "cibles" (nouvelles pathologies) seront très probablement de tailles limitées.

4.1 Architecture définitive du réseau Multimodal3DSiameseNet

La conception de notre réseau de neurones final tient compte de plusieurs contraintes :

— Utilisation de plusieurs modalités : images médicales, facteurs de risque (n'évoluant pas avec le temps), autres données cliniques (évoluant avec le temps)

— Apprentissage d'éventuelles corrélations entre les modalités

— Comparaison de deux visites i et $i + \delta$ d'un même patient, pour chaque modalité (avec δ un intervalle quelconque entre les visites)

Le principe de notre réseau multimodal est présenté dans notre schéma ci-dessous (Figure 4.1). Pour rappel, l'architecture détaillée du réseau est présentée dans la Figure 3.13, en page 76.

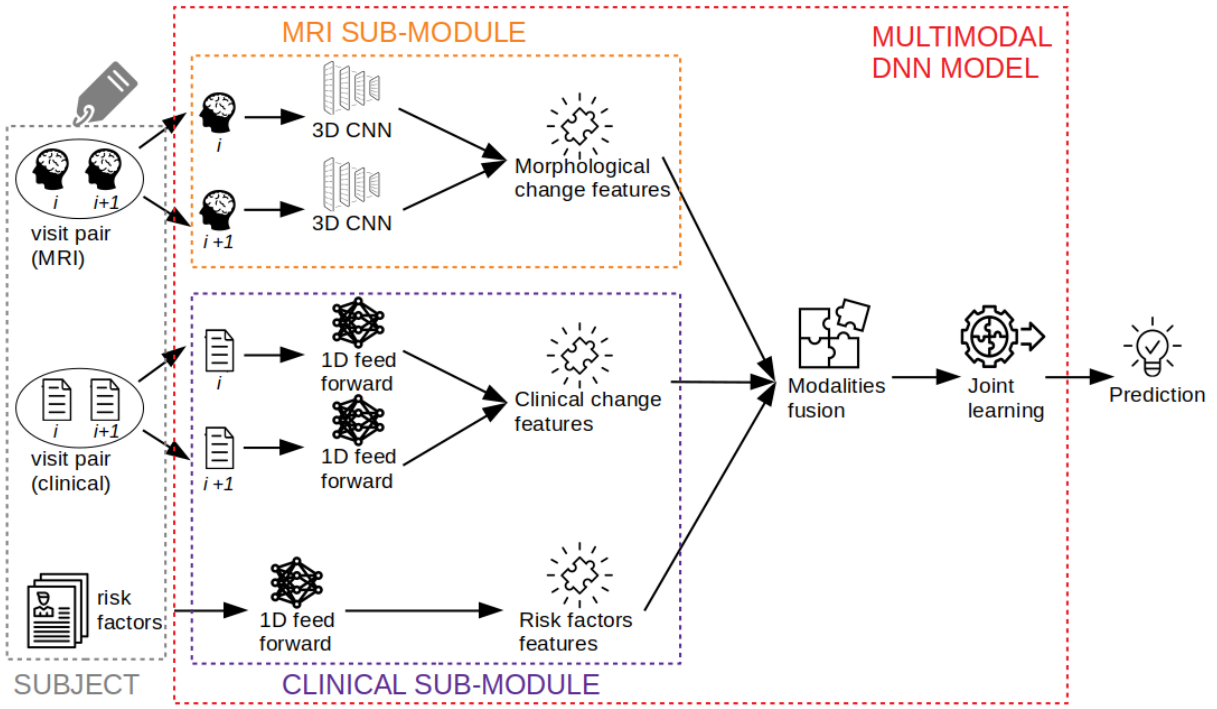


FIGURE 4.1 – Principe du réseau multimodal avec sous-modules adaptés à chaque modalité, puis fusion intermédiaire des modalités et apprentissage conjoint

Notre réseau multimodal se compose de sous-modules, adaptés aux caractéristiques de chaque modalité, comme nous l'avons décrit dans le chapitre 3 : CNN 3D pour les IRM, et succession de couches *fully-connected* pour les données cliniques, ici essentiellement des tests cliniques et pour les facteurs de risque. Ainsi nous pouvons obtenir différents points de vue sur l'évolution de la maladie : un point de vue morphologique avec les IRM, un point de vue cognitif avec les tests cliniques, et des informations complémentaires données par les facteurs de risque comme l'âge et le génotype.

Ces trois types de *features* sont ensuite fusionnés, pour une étape d'apprentissage conjoint, de sorte que cette dernière partie du réseau calcule des caractéristiques supplémentaires, notamment en tenant compte des corrélations entre les différentes modalités. Enfin, le réseau donne une prédiction d'appartenance à la classe Stable ou Déclin pour le sujet en question.

À noter que, tout comme dans le chapitre 3, la vérité-terrain est obtenue par *clustering* non-supervisé, en se basant sur les scores MMSE d'au moins trois visites médicales sur la durée totale de l'étude (72 mois pour la base ADNI pour la maladie d'Alzheimer, 36 mois pour la base PPMI sur la maladie de Parkinson).

4.2 Pipeline pour la création des jeux de données

La Figure 4.2 présente le *pipeline* pour la création d'un jeu de données multimodal labellisé. Nous présentons ici ce *pipeline* de manière générique, c'est-à-dire pouvant être appliqué pour n'importe quelle base de données d'étude longitudinale de patients pour une maladie neuro-dégénérative.

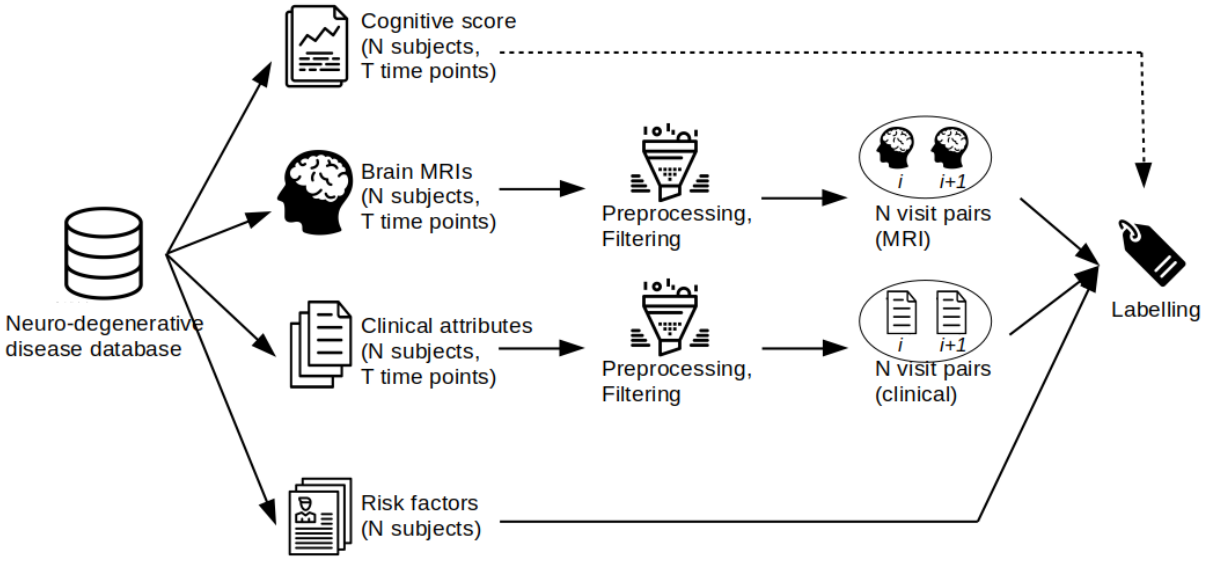


FIGURE 4.2 – Utilisation de diverses modalités pour constituer un jeu de données labellisé pour l'étude de l'évolution de maladies neuro-dégénératives

A partir de notre base de données concernant la maladie neuro-dégénérative d'intérêt, nous récupérons les IRM de cerveau, les données cliniques, et les facteurs de risque des sujets. Ces données sont récupérées pour chacune des visites que l'on souhaite prendre en compte dans l'étude. Ces données seront pré-traitées (*skull-stripping* pour les IRM, ou remplacement des valeurs manquantes pour les données cliniques, par exemple) et filtrées

(par exemple en supprimant les sujets ayant trop peu de visites médicales disponibles). Enfin, pour chaque sujet, nous groupons les visites médicales par paires. Nous récupérons également, pour chaque sujet et pour chaque visite, la valeur du score cognitif que nous utiliserons pour créer la vérité terrain en utilisant la méthode de Bhagwat *et. al.* décrite dans le Chapitre 3, Section 3.2.1, page 66.

Ce pipeline sera utilisé dans ce chapitre, pour la création de notre jeu de données final avec la base ADNI [5], ainsi que pour la création de notre jeu de données de patients Parkinson avec la base PPMI [6].

4.3 Jeu de données final pour ADNI

4.3.1 Création des paires de visites

Pour l'analyse présentée dans ce chapitre, nous avons inclus tous les sujets de la cohorte ADNI1 ayant réalisés une IRM structurale à l'inclusion, à 6 mois, à 12 mois, et à 24 mois, ce qui représente 382 sujets. La figure ci-dessous (Figure 4.3) montre la répartition de ces sujets. Cette figure permet de voir que parmi nos sujets MCI (en bleu), une partie évolue vers la maladie d'Alzheimer (en rouge) au cours du temps, ce qui correspond à une partie des sujets MCI identifiés comme étant en déclin cognitif.

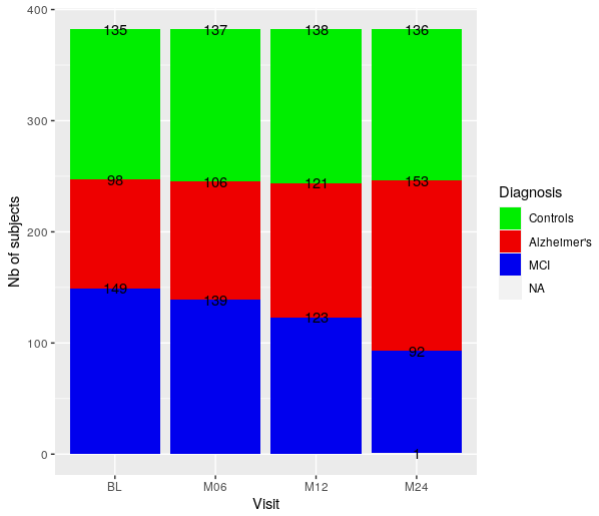


FIGURE 4.3 – Répartition des sujets en fonction de leur diagnostique, pour les quatre visites d'intérêt : inclusion, 6 mois, 12 mois, et 24 mois.

Pour chaque sujet nous formons 6 paires, correspondant aux combinaisons possibles de deux visites successives : Inclusion/6 mois, 6 mois/12 mois, 12 mois/24 mois, Inclusion/12 mois, Inclusion/24 mois, et 6 mois/24 mois. Pour ne pas créer de biais, l'ensemble des paires de visites d'un même sujet ne peut appartenir qu'à un seul des trois sous jeux de données, entraînement, validation, ou test.

Pour notre jeu de données de 381 sujets, nous obtenons après filtrage 2268 paires, dont 1197 sont négatives (Stable) et 1071 sont positives (Déclin).

4.3.2 Pré-traitement et augmentation des données d'imagerie

Les images de cerveau sont *skull-stripped* (suppression de la boîte crânienne), puis centrées sur le cerveau et rognées pour supprimer une grande partie de l'arrière plan noir. Enfin, comme les cerveaux des sujets ont tous des tailles différentes, nous avons modifié la taille des images à $204 \times 216 \times 150$ pixels en utilisant une interpolation bi-linéaire, puis nous avons réduit cette taille par deux pour obtenir des volumes de $102 \times 108 \times 75$ pixels, afin que les images puissent être chargées dans la mémoire CPU.

Un grand nombre de stratégies existent pour augmenter artificiellement la taille d'un jeu de données : transformations géométriques, transformations de l'espace colorimétrique, filtres, suppression aléatoire de parties de l'image, transfert de style, *etc.* [9]. Nous avons choisi les quatre opérations suivantes, de sorte à générer de nouvelles images réalistes (voir Figure 4.4) :

— flou gaussien d'écart-type choisi aléatoirement entre 0 et 0.8 ;

— rotation selon l'axe gauche-droite du sujet, avec un angle choisi aléatoirement entre 0°et 5° ;

— permutation (*flip*) selon l'axe de symétrie gauche-droite du sujet, en partant de l'*a priori* que la maladie d'Alzheimer atteint de manière indifférente les deux côtés du cerveau ;

— modification aléatoire du contraste, par réduction de l'amplitude de l'histogramme des images.

Les différents paramètres pour ces opérations ont été choisis de façon empirique, pour obtenir le moins d'*overfitting* possible durant l'entraînement. Cette augmentation de données est réalisée à chaque nouvelle *epoch* durant l'entraînement, de façon intégrée dans le code, comme le montre le pseudo-code suivant :

```
While current epoch < number of epochs:
    Get 1 batch of training data
    For each item in a batch:
        baseline_mri, month12_mri = item

        sigma = random {0, 0.8}
        \\Gaussian blur of baseline_mri and month12_mri,
        \\with standard deviation sigma
        gaussian_blur(baseline_mri,sigma)
        gaussian_blur(month12_mri,sigma)

        \\Random percentile of pixel values,
        \\between 1st and 3rd
        low = random {first_percentile, third_percentile}
        \\Random percentile of pixel values,
        \\between 98th and 100th
        high = random {ninetyeight_percentile, hundred_percentile}
        \\Histogram stretching of baseline_mri and
        \\month12_mri, between low and high
        hist_stretch(baseline_mri,low,high)
        hist_stretch(month12_mri,low,high)

        angle1 = random  {0,5}
        angle2 = random  {0,5}
        neg1 = random {0,1}
        neg2 = random {0,1}
        If neg1 = 1:
            angle1 = - angle1
        If neg2 = 1:
            angle2 = - angle2
        \\Rotate baseline_mri
        \\by angle1 degrees along left/right axis
        rotate(baseline_mri,angle1)
        \\Rotate month12_mri
        \\by angle2 degrees along left/right axis
        rotate(month12_mri,angle2)

        flip = random {0,1}
        If flip = 1:
            \\Flip baseline_mri and month12_mri
            \\along left/right axis
            flip(baseline_mri)
            flip(month12_mri)
```

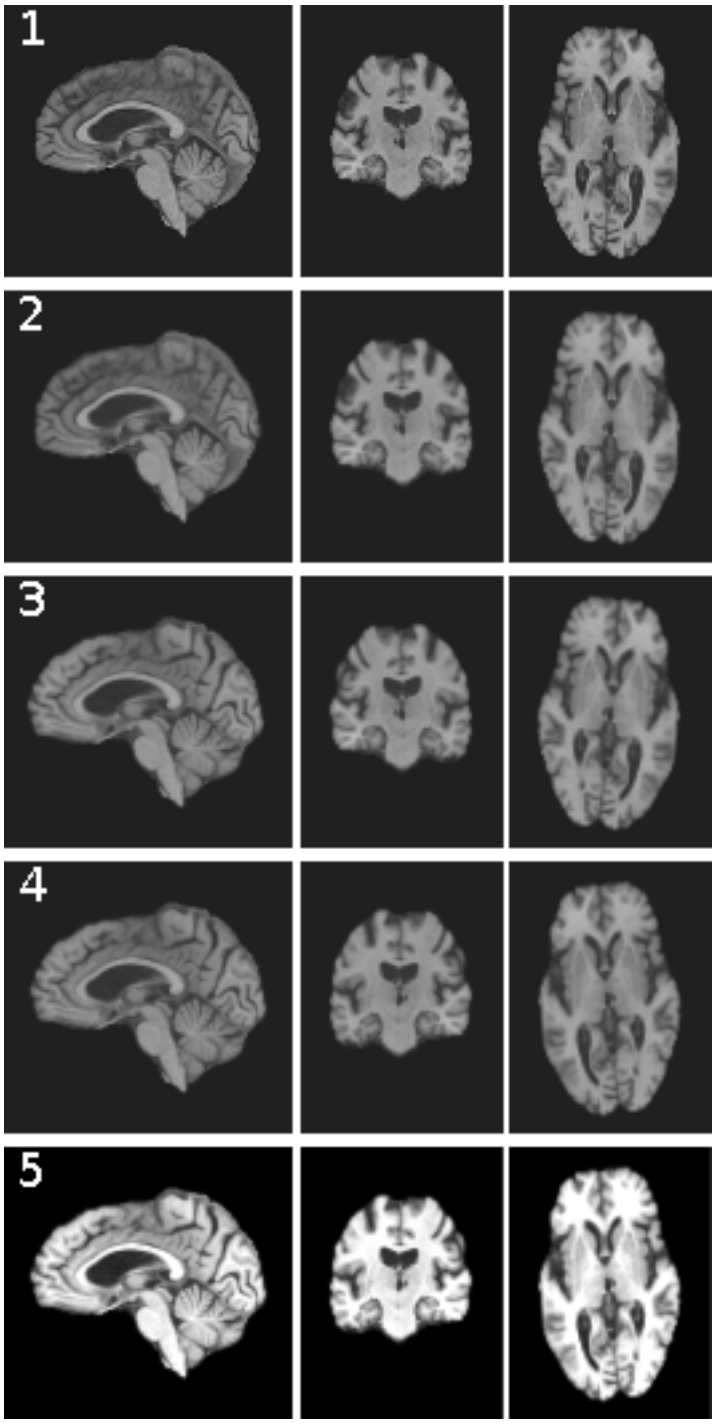


FIGURE 4.4 – Augmentation des IRM de cerveau (affichage de coupes médianes en plan sagittal à gauche, plan coronal au milieu, et plan axial à droite). 1 : image originale ; 2 : après flou gaussien (sigma = 0.5) ; 3 : après rotation (-3°) ; 4 : après symétrie gauche-droite ; 5 : après modification du contraste

4.3.3 Pré-traitement des données cliniques

Concernant les données cliniques, la base ADNI fournit des données démographiques (âge, sexe biologique), des informations génomiques comme le nombre d'allèles APOE4, des résultats de scores cognitifs, ainsi que des mesures sur des prélèvements biologiques (voir Table 4.1, et Table 6.1 en Annexe A, page 136.

Variable	**Catégorie**	**Nombre de valeurs manquantes**
AV45	Biomarqueur	1528 (totalité des observations)
LDELTOTAL	Score neuro	389
RAVLT learning	Score neuro	9
MMSE	Score neuro	2
RAVLT immediate	Score neuro	9
APOE4	Génotype	0
CDRSB	Score neuro	7
FAQ	Score neuro	7
ADASQ4	Score neuro	6
ADAS11	Score neuro	4
ADAS13	Score neuro	28
PIB	Biomarqueur	1436
ABETA	Biomarqueur	1068
TAU	Biomarqueur	1068
PTAU	Biomarqueur	1068
TRABSCOR	Score neuro	58
RAVLT forgetting	Score neuro	11
RAVLT % forgetting	Score neuro	34
DIGITSCOR	Score neuro	17
FDG	Biomarqueur	778

TABLE 4.1 – Nombre de valeurs manquantes par variable, pour les 382 sujets de la cohorte ADNI-1, en comptant les visites d'inclusion, 6 mois, 12 mois, et 24 mois

Pour pré-traiter ces données, nous commençons par supprimer les variables ayant un grand nombre de données manquantes. Ainsi, nous supprimons les bio-marqueurs PIB, ABETA, TAU, PTAU, et FDG. Ensuite, après avoir calculé les corrélations de Spearman deux à deux, nous sélectionnons une seule variable pour chaque groupe de variables fortement corrélées. Finalement, nous excluons de notre analyse la variable MMSE, étant donné que c'est cette variable qui est utilisée pour créer notre vérité terrain. Ainsi, au total nous sélectionnons les trois facteurs de risque AGE, GENDER, et allèles APOE4, ainsi que les huit scores cognitifs suivants : LDELTOTAL, RAVLT learning, RAVLT immediate, CDRSB, FAQ, TRABSCOR, RAVLT forgetting, et DIGITSCOR.

4.4 Entraînement et résultats

4.4.1 Protocole d'entraînement

Pour l'entraînement de notre Multimodal3DSiameseNet, nous avons séparé le jeu de données au niveau des sujets, pour que toutes les paires d'un même sujet soient dans un seul sous-ensemble de données (comme nous l'avons expliqué précédemment). Le jeu de données est distribué de la manière suivante : 60% pour l'entraînement (229 sujets), 20% pour la validation (76 sujets), et 20% pour le test (76 sujets). Nous avons entraîné notre modèle multimodal en suivant un protocole de validation croisée à 4 échantillons (4-*folds*). La table ci-dessous présente la répartition des paires entre nos deux classes :

	Echantillon 1		**Echantillon 2**		**Echantillon 3**		**Echantillon 4**	
	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs
Entraînement	744	624	696	666	708	660	750	615
Validation	222	228	270	186	258	192	216	237
Test	231	219	231	219	231	219	231	219

TABLE 4.2 – Répartition des données (paires de visites) en deux classes, pour les quatre échantillons (*folds*) de la validation croisée

A titre de comparaison, nous avons également entraîné une version "images uniquement" et une version "données cliniques uniquement" de notre réseau. Les trois versions ont été entraînées sur 15 *epochs* avec un *learning rate* de 0.005, en sauvegardant le modèle uniquement si la valeur de *loss* de validation avait diminué à chaque étape. En effet, après 15 *epochs*, nous n'avons plus constaté d'amélioration de la fonction de coût de validation, et nous avons donc arrêté l'entraînement, pour éviter le sur-apprentissage.

4.4.2 Résultats

La Table 4.3 ci-dessous, montre les principaux résultats expérimentaux. Le modèle **Image** obtient une *accuracy* stable à 0.50, et un score F1 très variable (écart-type important), qui sont expliqués par le fait que le modèle prédit toujours la même classe. Il n'y a donc en réalité aucun apprentissage de ce modèle.

Le modèle **Clinique** a de très bonnes performances deux fois sur quatre (*accuracy* de 0.91 et 0.93 ; F1 de 0.91 et 0.92), et très mauvaises les deux autres fois (*accuracy* de 0.51 et 0.18 ; F1 de 0.0 et 0.05). Cela montre que ce modèle est très dépendant du jeu de données d'entraînement. Enfin, le modèle **Multimodal** a des performances élevées et stables sur les 4 échantillons. La combinaison des deux modalités permet donc un apprentissage efficace et beaucoup moins dépendant des données utilisées pour l'entraînement, donc plus robuste.

	Test accuracy	Test F1
Image	0.50 (± 0.01)	0.33 (± 0.38)
Clinique	0.64 (± 0.38)	0.47 (± 0.53)
Multimodal	**0.91 (± 0.01)**	**0.90 (± 0.01)**

TABLE 4.3 – Comparaison des résultats de l'entraînement des deux versions unimodales et de la version multimodale de notre modèle. Les valeurs sont les moyennes sur les 4 échantillons de la validation croisée, et les écart-types entre parenthèses.

Nous avons également comparé ces résultats à ceux de notre modèle siamois de référence, le Longitudinal Siamese Network (LSN) de Bhagwat *et. al.*. (voir Table 4.4).

Notre modèle multimodal a des performances légèrement inférieures à celles du modèle LSN, mais comme nous l'avons fait remarquer dans le chapitre précédent, les bons résultats de Bhagwat *et. al.* peuvent être en partie expliqués par le fait que le score MMSE ayant servi à labelliser les données est également fourni à leur modèle.

Modèle	Taille du jeu de données	1ère visite	2nde visite	Données cliniques	Données IRM	Acc	F1	AUC
Régression logistique	1116 sujets (1116 paires)	inclusion	suivi à 6 ou 12 mois	- 2 facteurs de risque - 2 scores cognitifs (dont MMSE)	épaisseurs corticales de 78 ROIs	0.91	0.91	0.96
SVM						0.89	0.89	0.96
Forêt d'arbres décisionnels						0.88	0.89	0.96
LSN [85]						0.94	0.94	0.99
Multimodal 3DSiameseNet	382 sujets (2268 paires)	$i \in$ [inclusion, 6 mois, 12 mois]	$i + \delta \in$ [6 mois, 12 mois, 24 mois]	- 3 facteurs de risque - 9 scores cognitifs	IRM cerveau entier (3D)	0.91	0.90	0.96

TABLE 4.4 – Comparaison de notre modèle multimodal avec les résultats présentés par Bhagwat *et. al.* pour leur LSN et leur comparaison à l'état de l'art [85]

4.4.3 Importance des variations court-terme sur les performances de prédiction

Après avoir entraîné notre modèle, nous obtenons une *accuracy* moyenne de 91% avec un écart-type de 1%, et un score F1 moyen de 0.90 avec un écart-type de 0.01. Les histogrammes de la Figure 4.5 ci-après montrent le nombre de paires bien et mal classées (résultats de la matrice de confusion), ainsi que la répartition des diagnostiques AD, MCI, et contrôles (NC) dans chaque groupe.

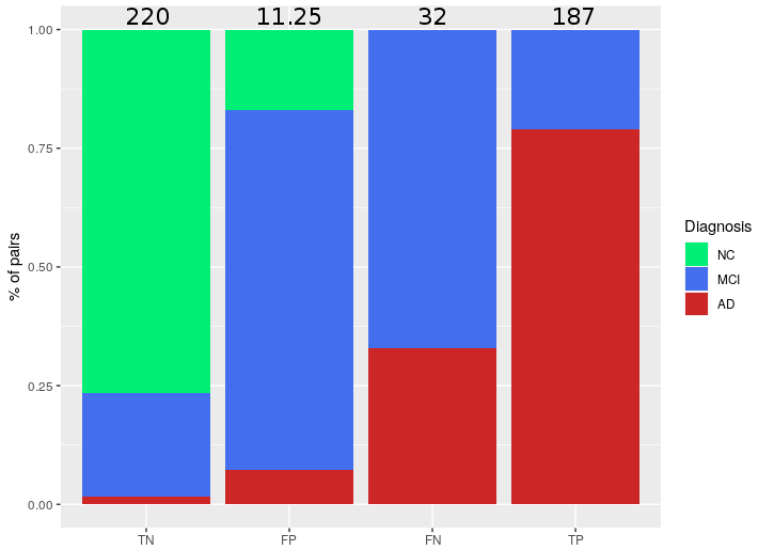


FIGURE 4.5 – Répartition des diagnostiques au niveau des sujets pour toutes les paires du jeu de test. TN : vrais négatifs, FP : faux positifs, FN : faux négatifs, TP : vrais positifs. Les valeurs au dessus des histogrammes sont les effectifs totaux moyens (nombres de paires), sur les quatre échantillons de la validation croisée.

Comme nous l'attendions, les sujets AD et contrôles sont majoritairement bien classés, respectivement dans nos classes Stable et Déclin. Cela confirme que les sujets qui ne montrent pas de signes pathologiques (maladie d'Alzheimer ou autre) sont facilement identifiables, de même que les sujets affectés sévèrement.

Cependant, un grand nombre de sujets MCI sont mal classés. Comme les sujets MCI forment un groupe hétérogène (ils peuvent être soit stables soit en déclin cognitif), nous devons nous intéresser plus précisément à la valeur de leurs scores MMSE, dans le but d'interpréter ces erreurs de classification.

Pour cela, nous avons comparé la valeur de MMSE correspondant à la seconde visite de chaque paire (seconde visite donnée à notre modèle pour l'inférence) à la dernière valeur de MMSE disponible pour chaque sujet dans la base de données entière (voir Figure 4.6). Ces résultats supplémentaires montrent que certains faux négatifs correspondent à des paires pour lesquelles la valeur de MMSE est faible à la fin de l'étude globale, mais était encore élevée à la seconde visite de la paire. Cela signifie que, bien que notre modèle ait une bonne performance pour la prédiction du déclin cognitif au long terme, il reste sensible aux variations locales de l'état cognitif du patient.

FIGURE 4.6 – En haut : Répartition des dernières valeurs de MMSE dans la base données entière (dans un maximum de 72 mois) pour les paires bien ou mal classées par notre modèle. En bas : Répartition des valeurs de MMSE à la seconde visite d'intérêt pour les paires bien ou mal classées par notre modèle. TN : true negatives, FP : false positives, FN : false negatives, TP : true positives. TN : vrais négatifs, FP : faux positifs, FN : faux négatifs, TP : vrais positifs. Sur l'échelle de couleurs le blanc correspond à une MMSE de 26, qui est le seuil pour le diagnostique de la maladie d'Alzheimer [102]. Les valeurs au dessus des histogrammes sont les effectifs totaux moyens, sur les quatre échantillons de la validation croisée.

Bien entendu cela était prévisible, étant donné que, même si notre vérité terrain est basée sur l'évolution au long terme du score MMSE, notre modèle ne peut inférer des informations qu'à partir de deux points de temps. Ainsi, dans le cas des sujets qui ont commencé à décliner après la seconde visite d'intérêt, notre modèle n'a pas accès à assez d'information pour faire une prédiction correcte.

Enfin, nous pouvons noter que ces résultats nous permettent d'expliquer les faux négatifs, mais pas les faux positifs, moins nombreux, que produit notre modèle.

4.4.4 Influence de la durée de l'intervalle entre les deux visites

Étant donné que les six différentes paires possibles par sujet correspondent à un intervalle de temps variable entre deux visites médicales (intervalles allant de 6 à 24 mois), nous avons cherché à savoir si la durée de l'intervalle avait une influence sur la performance du modèle. En effet, on pourrait s'attendre par exemple à ce que les paires Inclusion/24 mois soient mieux classées.

Pourtant, la Table 4.5 ci-dessous montre que contrairement à notre hypothèse, un intervalle plus grand entre deux visites ne permet pas nécessairement de mieux caractériser le déclin cognitif chez les patients avec notre modèle multimodal. Ainsi, notre modèle est capable de classer correctement les sujets entre Stable et Déclin à partir d'un intervalle aussi court que 6 mois.

Intervalle	**6 mois**	**12 mois**	**18 mois**	**24 mois**
Précision	0.97 (± 0.03)	0.95 (± 0.04)	0.93 (± 0.05)	0.94 (± 0.06)
Rappel	0.83 (± 0.05)	0.84 (± 0.02)	0.91 (± 0.02)	0.89 (± 0.02)
F1	0.89 (± 0.02)	0.90 (± 0.02)	0.92 (± 0.03)	0.91 (± 0.02)

TABLE 4.5 – Variation du score F1 en fonction de la durée de l'intervalle entre deux visites. Les valeurs sont les moyennes sur les 4 échantillons de la validation croisée, et les écart-types entre parenthèses

4.5 Transfert d'apprentissage pour l'étude d'autres maladies neuro-dégénératives

Notre modèle siamois multimodal ayant été validé sur la prédiction de l'évolution cognitive de patients atteints de la maladie d'Alzheimer, nous avons cherché à l'appliquer au suivi d'une autre maladie neuro-dégénérative. Pour cela, nous allons utiliser le principe de transfert d'apprentissage : le modèle pré-entraîné sur les patients Alzheimer (cf chapitre précédent) servira d'initialisation à un second entraînement avec des sujets issus d'un nouveau jeu de données. Pour cet apprentissage par transfert, la tâche à effectuer est toujours la même, c'est-à-dire la classification des sujets entre sujets stables et sujets en déclin cognitif.

4.5.1 Choix d'un second jeu de données

Nous avons choisi comme seconde maladie neuro-dégénérative la maladie de Parkinson. Cette maladie est caractérisée par des tremblements, une rigidité musculaire, et un ralentissement des mouvements (bradykinésie), ainsi qu'à des troubles cognitifs [103]. D'après DeMaagd *et. al.* [103], ces symptômes moteurs sont attribués à une dégénérescence progressive des neurones dopaminergiques de la substance noire (voir Figure 4.7).

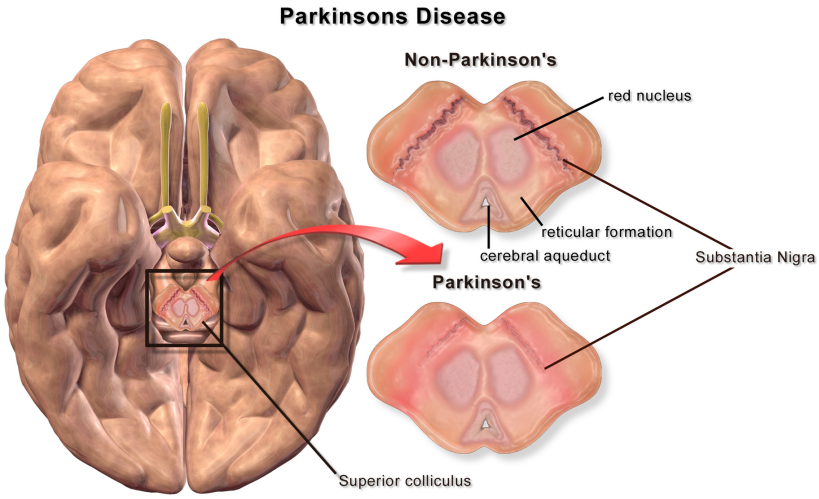


FIGURE 4.7 – Dégradation de la substance noire associée avec la maladie de Parkinson (figure issue de [104])

Nous avons utilisé la base de données PPMI (*Parkinson's Progression Markers Initiative*) [6], qui repose sur un suivi longitudinal de 200 patients contrôles et 400 patients atteints par la maladie. Lors de ce suivi, les patients se rendent à une visite médicale lors de l'inclusion, qui sera suivie de visites à 12 mois, 24 mois, et 36 mois. De nombreuses données sont ainsi récoltées, en particulier des IRM structurelles de cerveaux (3D volu-

mique), des scores cognitifs, et des informations sur les divers facteurs de risques. Cette base de données comporte donc des données provenant de modalités que nous avons utilisées pour construire notre modèle de prédiction de déclin cognitif, ce qui en fait une bonne candidate pour le *transfer learning*. De plus, cette base de données est de petite taille, comparativement à la base ADNI, ce qui en fait un bon cas d'application pour un transfert d'apprentissage entre deux études médicales.

Pour créer notre jeu de données, nous avons dans un premier temps créé une vérité terrain à long-terme, de façon similaire à l'approche utilisée avec la base ADNI. Le score que nous avons utilisé comme marqueur de la progression de la maladie de Parkinson chez les sujets est le score UPDRS (*Unified Parkinson Disease Rating Scale*), qui est utilisé par les médecins pour poser le diagnostique de maladie de Parkinson. Nous avons utilisé ce score pour grouper les sujets selon leur stabilité ou déclin cognitif, en fonction de l'évolution du score tout au long des 36 mois de suivi (voir Figure 4.8), en utilisant une classification ascendante hiérarchique avec distance Euclidienne et un saut de Ward sur au moins 3 visites par patient, de manière similaire à la méthode expliquée en Section 3.2.1, page 66.

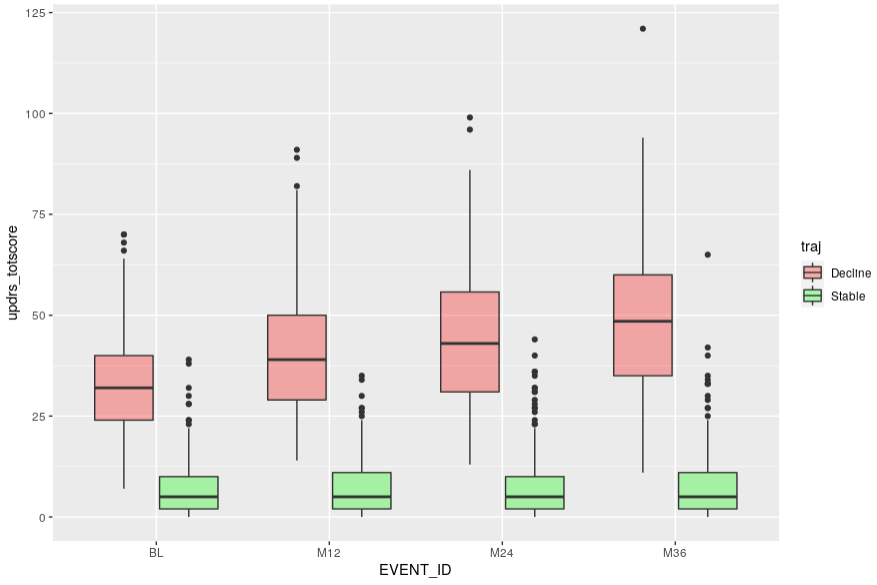


FIGURE 4.8 – Evolution du score UPDRS sur 4 visites, de l'inclusion à 36 mois, pour les deux groupes de sujets : Stable et Déclin

En ce qui concerne les données, nous avons utilisé les paires constituées de la visite d'inclusion et de la visite de suivi à 12 mois, ce qui nous donne 134 paires, dont 47 sont des sujets stables et 87 des sujets en déclin.

Nous avons utilisé les mêmes facteurs de risque que pour les données ADNI (i.e. âge, sexe biologique, et allèles APOE4), et nous avons utilisé les 13 scores cognitifs restants pour la partie données cliniques : hy, NHY, upsit, hvlt_immediaterecall, HVLTRDLY, HVLTREC, HVLTFPRL, hvlt_discrimination, hvlt_retention, lns, quip_any, scopa, stai (voir Table 6.2 en Annexe A, page 137, pour plus de détails sur les différents scores).

Les IRM de cerveau ont été pré-traitées de la même façon que pour les IRM de la base ADNI dans le chapitre précédent (*skull-stripping*, rognage autour du cerveau, et redimensionnement pour obtenir une image de taille $256 \times 256 \times 160$ pixels, dont le volume est ensuite divisé par deux pour obtenir des volumes de $102 \times 108 \times 75$ pixels.

4.5.2 Stratégies d'apprentissage par transfert

La base de l'apprentissage par transfert est la réutilisation des paramètres (poids des différents *kernels*) appris sur le domaine source pour l'apprentissage sur le domaine cible (voir état de l'art, Section 1.3). Pour rappel, nous utilisons ici le modèle Mutimodal3DSiameseNet défini dans le chapitre précédent, dont l'architecture est présentée de façon simplifiée dans la Figure 4.9.

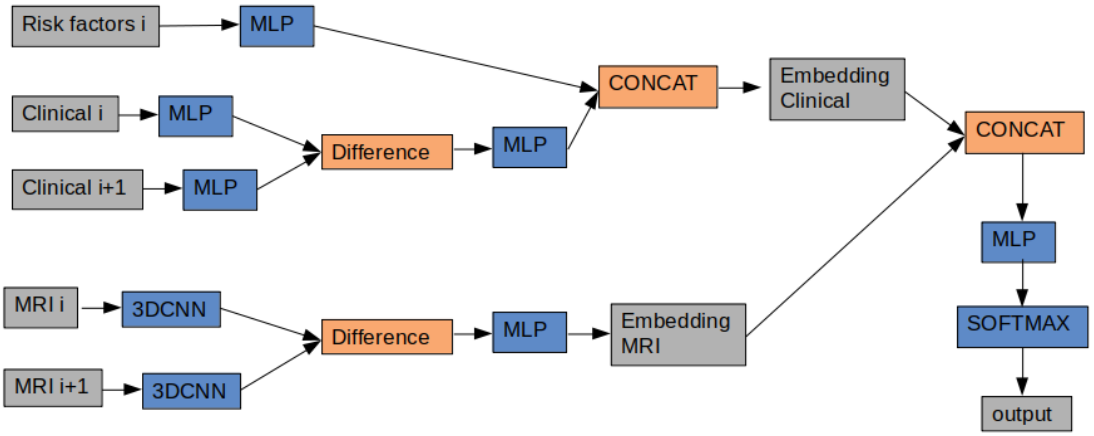


FIGURE 4.9 – Architecture simplifiée du modèle Mutimodal3DSiameseNet

Une fois les paramètres du modèle initialisés avec les valeurs pré-apprises, plusieurs choix sont possibles pour l'étape de *fine-tuning* (ré-entraînement) : il est possible de "geler" certaines couches, c'est-à-dire que leurs poids ne seront pas actualisés, et de choisir les couches que l'on veut ré-entraîner et celles que l'on veut entraîner à partir de zéro.

Ici nous avons comparé quatre approches :

— ① Entraînement du réseau entier à partir de zéro (initialisation aléatoires des paramètres de toutes les couches).

— ② *Fine-tuning* de toutes les couches

— ③ *Fine-tuning* des sous-modules IRM (couches de convolution 3D) et conjoint (couches entièrement connectées qui suivent la dernière opération de concaténation), et entraînement du sous-module clinique à partir d'une initialisation aléatoires des paramètres des couches.

— ④ Gel des couches de convolution, *fine-tuning* des couches *fully-connected* du sous-module IRM et du sous-module conjoint, et entraînement du sous-module clinique à partir de zéro

Les différents modèles ont été entraînés avec la même répartition des paires de visites en trois échantillons (3-*folds*) (voir Table 4.6 ci-après). La Figure 4.10 montre l'évolution de la fonction de coût pour les données de validation.

	Echantillon 1		**Echantillon 2**		**Echantillon 3**	
	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs
Entraînement	25	49	27	47	27	47
Validation	11	19	9	21	9	21
Test	11	19	11	19	11	19

TABLE 4.6 – Répartition des données (paires de visites) en deux classes, pour les trois échantillons (*folds*) de la validation croisée

Durant l'entraînement, nous avons sauvegardé les paramètres du modèle uniquement lorsque cette fonction de coût atteignait un nouveau minimum. Le modèle a été entraîné durant 30 *epochs* (voir Figure 4.10), et nous avons sauvegardé les paramètres des filtres après 5 *epochs* et après 30 *epochs*.

Enfin, nous avons utilisé le jeu de données de test sur ces deux configurations du modèle (entraîné durant 5 *epochs*, et entraîné durant 30 *epochs*), afin de voir d'une part quelle méthode donne les meilleurs résultats, et d'autre part quelle méthode converge le plus rapidement.

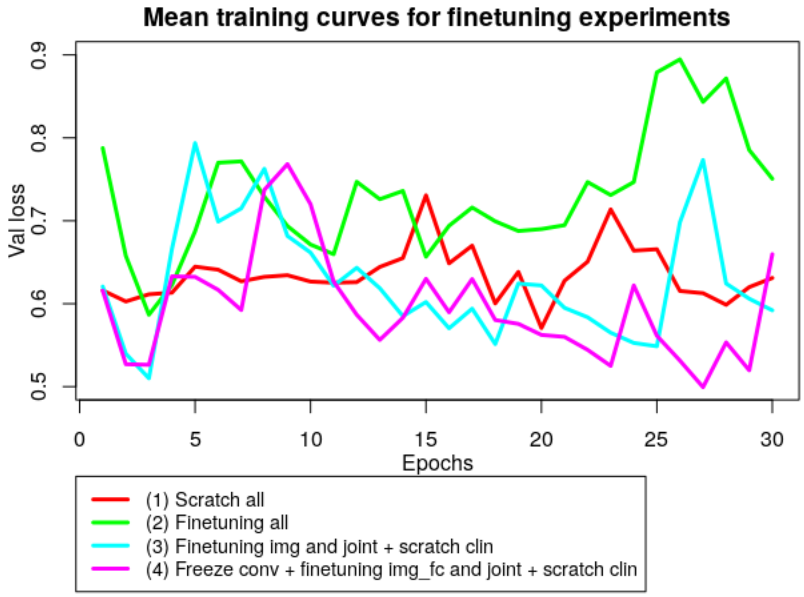


FIGURE 4.10 – Evolution de la fonction de coût de validation durant l'entraînement, pour nos quatre stratégies de *fine-tuning*

4.5.3 Résultats

En entraînant les modèles pendant 5 *epochs*, avec la stratégie ① le modèle n'a pas pu être optimisé lors de l'utilisation du troisième échantillon de validation croisée (la valeur de la fonction de coût pour les données de validation n'a jamais diminué). Nous présentons tout de même ces résultats dans la Table 4.7 et dans la Figure 4.11, mais cela montre déjà que ce modèle est le moins robuste, le plus sensible à son initialisation, et le plus susceptible d'*overfittting* en présence d'un petit jeu de données, parmi les 4 modèles étudiés.

Stratégie	**Test accuracy**	**Test F1**	**Test AUC**
①*	0.63 (± 0.0)	0.78 (± 0.0)	0.72 (± 0.13)
②	0.68 (± 0.08)	0.80 (± 0.04)	0.71 (± 0.13)
③	0.80 (± 0.06)	0.85 (± 0.05)	0.82 (± 0.10)
④	0.80 (± 0.07)	0.86 (± 0.04)	0.81 (± 0.10)

TABLE 4.7 – Résultats de l'apprentissage par transfert pour nos quatre différentes stratégies de *fine-tuning*, après entraînement pendant 5 *epochs*. Les valeurs sont les moyennes sur les échantillons de la validation croisée, et les écart-types entre parenthèses. *pour la méthode ①, l'entraînement n'a pas abouti avec le 3ème échantillon et il s'agit donc des moyennes sur les 2 échantillons restants.

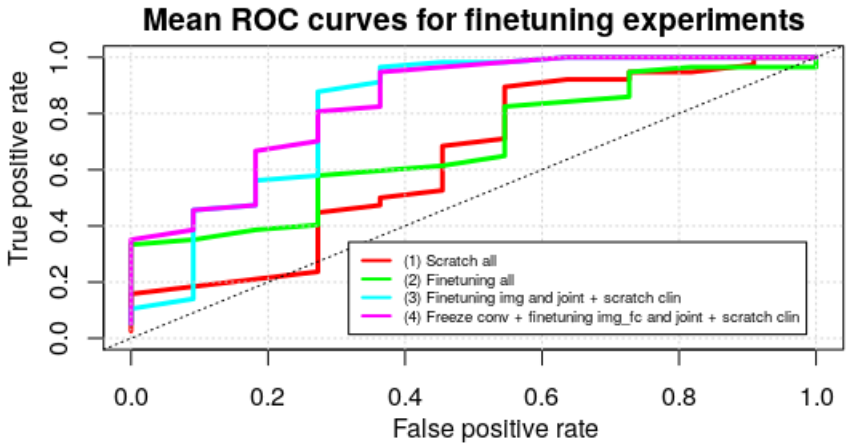


FIGURE 4.11 – Moyennes des courbes ROC pour nos quatre stratégies de *fine-tuning* sur les 3 échantillons de validation croisée, après entraînement pendant 5 *epochs*

Après un entraînement durant 30 *epochs*, nous constatons que les performances de tous les modèles ont augmenté (voir Table 4.8 et Figure 4.12). Cela est normal car plus de données augmentées ont pu être fournies au réseau durant ces *epochs* supplémentaires, étant donné que l'augmentation des données se fait de façon continue à chaque *epoch*.

Stratégie	**Test accuracy**	**Test F1**	**Test AUC**
①	0.77 (± 0.04)	0.84 (± 0.02)	0.82 (± 0.09)
②	0.71 (± 0.03)	0.76 (± 0.06)	0.82 (± 0.03)
③	0.78 (± 0.13)	0.85 (± 0.06)	0.90 (± 0.01)
④	0.83 (± 0.04)	0.88 (± 0.02)	0.87 (± 0.05)

TABLE 4.8 – Résultats de l'apprentissage par transfert pour nos quatre différentes stratégies de *fine-tuning*, après entraînement pendant 30 *epochs*. Les valeurs sont les moyennes sur les échantillons de la validation croisée, et les écart-types entre parenthèses.

Les trois autres modèles ont un faible ratio de vrais positifs par rapport au total des positifs, mais le modèle ② est celui qui obtient les moins bonnes performances. Les performances insatisfaisantes du modèle ② peuvent être expliquées par le fait que l'on ait utilisé les poids pré-entraînés pour la partie clinique, alors que les scores cliniques utilisés ne sont pas les mêmes dans la base ADNI et dans la base PPMI, ce qui a pu rallonger le temps d'apprentissage par rapport à une initialisation aléatoire.

Les méthodes ③ et ④ donnent de meilleures performances de classification que la méthode ① où le modèle est entraîné à partir d'une initialisation aléatoire, et avec eux nous n'avons pas rencontré le même problème de convergence de l'apprentissage, ce qui montre que l'apprentissage par transfert a permis d'obtenir un meilleur modèle pour la classification entre sujets stables et sujets en déclin, dans le cadre de la maladie de Parkinson, que l'apprentissage classique.

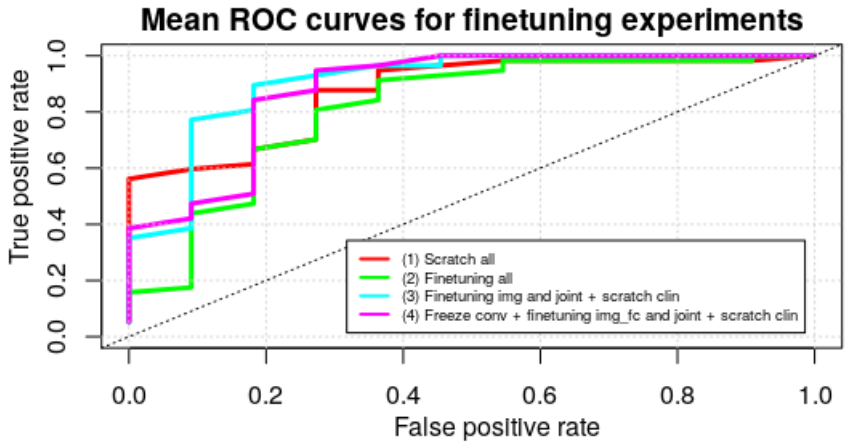


FIGURE 4.12 – Moyennes des courbes ROC pour nos quatre stratégies de *fine-tuning* sur les 3 échantillons de validation croisée, après entraînement pendant 30 *epochs*

Les stratégies ③ et ④ semblent donc être celles qui favorisent l'optimisation de l'architecture pour la classification sur les données PPMI. Ces deux stratégies donnent des résultats très proches, comme on peut le voir avec les écart-types qui se recouvrent. Cela peut être dû au fait que notre jeu de données de test est de très petite taille.

Nous noterons toutefois qu'aucune différence significative n'existe entre ces quatre résultats, comme l'a montré un test de Kruskall-Wallis (p-value $= 0.08798 > 0.05$). Ainsi, en se basant sur ces résultats, le choix d'une stratégie plutôt que l'autre se fera plutôt sur des considérations de temps de calcul (stratégie ④ plus rapide, car les couches de convolution 3D ne sont pas ré-entraînées).

En conclusion, nous avons proposé un modèle appelé Multimodal3DSiameseNet, qui est **adaptable**, au sens où il peut facilement être ré-entraîné sur une autre base de données concernant une maladie neuro-dégénérative différente : dans le cas de transfert d'Alzheimer vers Parkison, il suffit de faire un *fine-tuning* des couches *fully-connected* du sous-module IRM et du sous-module conjoint, et d'entraîner le sous-module clinique à partir de zéro. Il n'est pas besoin de ré-entraîner les couches de convolution.

4.6 Synthèse

Dans ce chapitre, nous avons présenté des résultats supplémentaires, obtenus en utilisant un jeu de données plus grand issu de la base ADNI (maladie d'Alzheimer). Dans ce nouveau jeu de données, constitué de 381 sujets, nous incluons quatre visites médicales possibles : inclusion, 6 mois, 12 mois, et 24 mois. Ces nouveaux résultats montrent que les performances de classification de notre modèle varient peu en fonction de l'intervalle de temps qui sépare les deux visites médicales d'entrée. Ce résultat est intéressant, car on pourrait s'attendre à ce que la qualité des prédictions augmente avec la durée entre deux visites (l'état de santé du sujet ayant eu plus de temps pour évoluer ou se stabiliser).

Cette caractéristique de notre modèle est importante, car il est fréquent dans les études de cohortes que des sujets manquent des visites de suivi.

Nous avons également montré que notre modèle, Multimodal3DSiameseNet, est capable de prédire l'évolution d'un patient sur le long terme (vérité terrain décrivant l'évolution des sujets sur 72 mois), à partir de visites médicales acquises tôt dans la période de développement de la maladie (données d'entrées acquises entre l'inclusion et 24 mois).

Cependant, nous avons aussi démontré que certaines erreurs de classification commises par notre modèle sont causées par des variations ponctuelles de l'état du patient. Ces variations se ressentent sur le score cognitif utilisé pour la création de la vérité terrain, et donc conduisent à une vérité terrain qui peut être inadéquate. Ainsi, il serait profitable d'intégrer l'avis d'un médecin ou d'un expert de la maladie lors de la phase d'étiquetage des données (construction de la vérité terrain).

Enfin, nos résultats de transfert d'apprentissage sur un petit jeu de données issu de la base de données PPMI (maladie de Parkinson) ont montré que le modèle pré-entraîné est plus rapidement optimisé et donne de meilleurs résultats de classification que le même modèle entraîné à partir de zéro sur le même jeu de données. La meilleure stratégie pour le second entraînement a été de ré-entraîner uniquement les couches qui reçoivent des données très similaires à celles du pré-entraînement (ici les images IRM de cerveau et les *features* multimodales), et de ré-initialiser aléatoirement les poids des autres couches (ici le module données cliniques).

En particulier, nous avons montré que les filtres appris par notre modèle dans les couches de convolution n'ont pas besoin d'être ré-apprises dans le cadre du transfert d'Alzheimer vers Parkinson, ce qui suggère que ces *features* images sont caractéristiques du déclin cognitif, et peuvent être réutilisées pour une maladie similaire. En effet, dans notre cas d'étude, seules les couches entièrement connectées (pour chacun des sous-modules et pour l'apprentissage commun après fusion intermédiaire) doivent être ré-entraînées.

Ce transfert d'apprentissage peut donc se faire, à moindre coût calculatoire, même pour des bases de taille restreinte, puisque la plupart des hyperparamètres à ajuster se trouvent dans les couches de convolution et que bien sûr, réduire le nombre d'hyperparamètres à ajuster réduit mécaniquement le risque d'*overfitting* (pour une taille de base d'apprentissage donnée).

Notre modèle pourrait donc être utilisé pour la prédiction de l'évolution au long terme de maladies neuro-dégénératives rares ou peu étudiées, pour lesquelles de grandes bases de données ne seraient pas disponibles.

Chapitre 5

Comparaison avec l'approche récurrente et retour sur les choix effectués pour la vérité terrain

Dans ce chapitre, nous comparerons notre modèle siamois avec l'approche des réseaux récurrents (RNN). Cette approche, popularisée par différents travaux sur le traitement du langage [41] ou l'analyse de flux vidéo [105] est basée sur le concept de séquences.

Dans un second temps, nous reviendrons sur certains choix qui ont été faits durant la thèse, avec notamment le protocole d'étiquetage des données lors de la création de la vérité terrain, et nous proposerons une façon alternative de construire cette vérité terrain.

5.1 Comparaison avec l'approche RNN

5.1.1 Modèle MildInt pour la maladie d'Alzheimer

Dans ces travaux de thèse, nous avons très tôt choisi de nous orienter vers une architecture siamoise pour notre tâche de prédiction d'évolution des maladies neuro-dégénératives. L'avantage de l'utilisation de ces réseaux est qu'ils permettent d'évaluer une évolution en ne prenant en compte que deux points de temps, et peuvent donc être utilisés même si l'on n'a pas à disposition beaucoup de visites médicales par patient. Par ailleurs, Bhagwat *et. al.* [85] ont publié au début de la période de ces travaux de thèse un modèle utilisant ce type de réseaux.

Dans d'autres domaines, comme le traitement du langage écrit ou parlé, les architectures de type réseaux récurrents ont été proposées, pour traiter des données séquentielles. Récemment, dans le domaine bio-médical, une approche utilisant les RNN a été publiée par Lee *et. al.* [86], pour la prédiction de l'évolution de patients atteints de la maladie d'Alzheimer (voir Etat de l'art, Section 2.4). Pour rappel, il s'agit d'un réseau multimodal, utilisant dans l'article d'origine quatre modalités : scores cognitifs, mesures de molécules

issues du liquide cérébro-spinal, mesures calculées sur des IRM, et facteurs de risque. Nous avons tenu à comparer leur modèle, nommé MildInt, à notre Multimodal3DSiameseNet, afin d'évaluer la différence de performances entre l'approche siamoise et l'approche récurrente pour la même tâche de classification.

Le code fourni par les auteurs de cet article présentant des incohérences avec l'architecture décrite dans l'article lui-même, nous avons ré-implémenté le modèle MildInt (code disponible à *https ://github.com/CeciliaOstertag/MildInt_implementation*).

De plus, afin de comparer équitablement les performances du modèle MildInt et celles de notre Multimodal3DSiameseNet, nous avons choisi d'utiliser exactement les mêmes 381 sujets de la base ADNI qu'au chapitre précédent, et des modalités identiques (scores cognitifs et facteurs de risques) ou provenant de la même source (mesures IRM pour MidlInt *v.s.* images IRM 3D pour Multimodal3DSiameseNet). Nous avons donc ré-implémenté MildInt avec 3 modalités au lieu de 4, ignorant les mesures de molécules issues du liquide cérébro-spinal auxquelles nous n'avons pas accès dans nos études.

La Figure 5.1 ci-dessous présente de façon simplifiée notre implémentation de cette architecture.

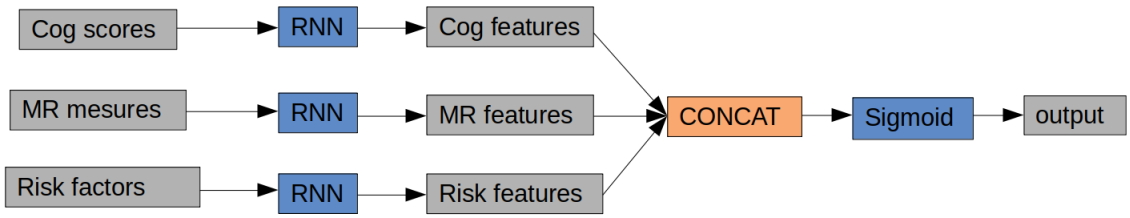


FIGURE 5.1 – Représentation simplifiée de l'architecture MildInt adaptée à nos trois modalités

Plus précisément, nous avons fourni comme entrées au modèle MildInt des séquences de quatre visites successives (Inclusion, 6 mois, 12 mois, et 24 mois), pour chaque modalité. Les modalités que nous avons utilisées sont les suivantes :

— **Scores cognitifs** : tous les scores cognitifs que nous avions sélectionnés préalablement (LDELTOTAL, RAVLT learning, RAVLT immediate, CDRSB, FAQ, TRABSCOR, RAVLT forgetting, et DIGITSCOR) (voir Annexe A).

— **Mesures IRM** : les 7 mesures calculées à partir des IRM, et disponibles dans la base ADNI (volume des ventricules, de l'hippocampe, du cerveau entier, du cortex entorhinal, du gyrus fusiforme, du lobe temporal médian, et volume intra-crânial total).

— **Facteurs de risques** : l'âge, le sexe biologique, et le génotype APOE4.

5.1.2 Résultats de l'entraînement du RNN avec notre jeu de données

Chaque sous-module RNN a été entraîné durant 30 *epochs*, en sauvegardant le modèle uniquement si la valeur de *loss* de validation avait diminué à chaque étape. Puis les poids finaux ont été chargés dans le modèle multimodal, lui-même ré-entraîné avec une stratégie de validation croisée à quatre échantillons (voir Table 5.1). Nous avons choisi ce nombre d'*epochs* car c'est le même que celui utilisé dans le chapitre précédent, pour entraîner notre modèle Multimodal3DSiameseNet. En pratique, nous avons observé qu'il est suffisant pour la convergence de l'apprentissage de MildInt.

	Echantillon 1		**Echantillon 2**		**Echantillon 3**		**Echantillon 4**	
	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs
Entraînement	123	106	117	112	118	111	125	104
Validation	38	38	44	32	43	33	36	40
Test	39	37	39	37	39	37	39	37

TABLE 5.1 – Répartition des sujets en deux classes, pour les quatre échantillons (*folds)* de la validation croisée

Les résultats obtenus sur le jeu de test, avec les trois sous-modules de MildInt pris séparément, puis avec le modèle MildInt multimodal, sont présentés dans la Table 5.2 ci-dessous.

	Test accuracy	**Test F1**
Mesures IRM	0.60 ($\pm$ 0.05)	0.56 ($\pm$ 0.02)
Scores cognitifs	0.96 ($\pm$ 0.01)	0.96 ($\pm$ 0.02)
Facteurs de risque	0.58 ($\pm$ 0.05)	0.39 ($\pm$ 0.28)
Multimodal	**0.97 ($\pm$ 0.02)**	**0.97 ($\pm$ 0.02)**

TABLE 5.2 – Résultats de l'entraînement des trois sous-modules de MildInt, et du modèle multimodal MildInt [86]. Les valeurs sont les moyennes sur les 4 échantillons de la validation croisée, et les écart-types entre parenthèses.

Les résultats obtenus par le module utilisant uniquement les facteurs de risque sont les plus mauvais, ce qui n'est pas étonnant car ces données sont peu informatives à elles seules.

Le module utilisant uniquement les mesures issues des IRM est légèrement meilleur, mais a quand même des performances très faibles. Il est possible que l'utilisation des images d'origine (IRM) en 3D au lieu de mesures pré-calculées conduise à de meilleurs résultats, mais cela signifierait remplacer la couche RNN 1D en une couche RNN convolutif 3D, et donc augmenterait l'espace mémoire et le temps de calcul nécessaire pour l'entraînement de cette partie, en plus de modifier complètement l'architecture de MildInt telle que présentée originellement par ses auteurs. Nous ne l'avons donc pas fait.

Le module utilisant uniquement les scores cognitifs donne des résultats très satisfaisants. Cela est attendu, car les scores cognitifs sont très corrélés avec le score MMSE

utilisé pour créer notre vérité terrain. Enfin, le modèle multimodal complet a les meilleures performances, ce qui montre encore une fois l'intérêt de combiner les modalités.

5.1.3 Comparaison avec notre modèle

La comparaison du modèle MildInt avec notre modèle Multimodal3DSiameseNet (Table 5.3) montre que le modèle MildInt a des performances de classification légèrement meilleures que celles de notre modèle Multimodal3DSiameseNet. Un test statistique de Kruskall-Wallis sur les scores F1 a montré que la différence entre les résultats de ces deux modèles est significative (p-value = 0.01942 $<$ 0.05). Notre approche reste cependant compétitive. En effet, la différence entre les valeurs d'*accuracy* des deux modèles est de 0.06 en moyenne, la différence entre les deux scores F1 est de 0.07, et la différence entre les deux valeurs d'AUC est de 0.04.

	Test accuracy	**Test F1**	**Test AUC**
Multimodal RNN (MildInt) [86]	0.97 ($\pm$ 0.02)	0.97 ($\pm$ 0.02)	1.0 ($\pm$ 0.0)
Multimodal3DSiameseNet	0.91 ($\pm$ 0.01)	0.90 ($\pm$ 0.01)	0.96 ($\pm$ 0.01)

TABLE 5.3 – Comparaison des résultats de l'entraînement du modèle MildInt [86], et de notre modèle multimodal. Les valeurs sont les moyennes sur les échantillons de la validation croisée, et les écart-types entre parenthèses.

Cependant, l'utilisation d'une architecture récurrente nécessite d'importantes ressources mémoire, ainsi qu'un temps de calcul important, dû au grand nombre de paramètres à optimiser. De plus, pour pouvoir s'en servir au moment de l'inférence, c'est-à-dire avec un nouveau sujet, il est nécessaire de posséder pour ce sujet une séquence relativement complète de visites médicales. En effet, il est techniquement possible d'utiliser les RNN avec séquences de tailles plus petites que celles utilisées pour entraîner le modèle, ou bien d'utiliser des séquences où certaines visites sont manquantes, mais le *padding* utilisé pour pallier à ces problèmes peut introduire un biais et être source d'erreurs de classification.

De plus, en pratique, l'acquisition des données médicales est coûteuse et difficile d'un point de vue logistique, et souvent les sujets n'ont que des séquences incomplètes de visites, en particulier dans le cas de maladies neuro-dégénératives. Ainsi, par rapport à MildInt, notre modèle présente l'avantage de n'utiliser que deux visites médicales, sans contraintes de temps entre les deux.

5.2 Retour sur les choix effectués pour la vérité terrain

Après avoir comparé notre approche siamoise à une approche récurrente de l'état de l'art, il nous est apparu important de revenir sur un autre choix que nous avons fait très tôt dans la thèse : l'élaboration de la vérité terrain.

5.2.1 Utilisation d'une vérité terrain alternative

Considérant notre choix d'utiliser un réseau siamois avec deux visites, i et $i + \delta$, nous pouvons choisir de créer notre vérité terrain de deux façons (voir Figure 5.2). La première est de considérer la trajectoire (évolution) du score cognitif de référence sur une durée de plusieurs années. Cette approche a été celle que nous avons présentée depuis le début de ce manuscrit, en accord avec la stratégie proposée par Bhagwat *et. al.* [85]. Nous l'appelons ici ***long-term trajectory ground truth***. Elle décrit l'évolution des sujets au long-terme, mais présuppose d'avoir accès à une étude longitudinale existant depuis plusieurs années, ce qui n'est pas forcément le cas en pratique.

Nous présentons ici une autre façon de considérer l'évolution de la maladie, correspondant au cas où l'on souhaite simplement savoir si l'état d'un patient s'est détérioré ou non par rapport à une visite précédente. Cette approche a été présentée en 2020 par Li *et. al.*, avec comme cas d'application la rétinopathie du prématuré et l'arthrite du genou [106]. Dans cet article, les auteurs disposent d'une échelle de stades d'évolution, pour chacune des deux maladies, et cherchent à distinguer les sujets en deux classes : ceux qui restent au même stade entre deux visites médicales, et ceux qui passent à un stade plus avancé. Ici, nous nous inspirons de ces travaux, en considérant l'évolution du score MMSE uniquement entre les deux visites médicales servant d'*input* au modèle, pour caractériser l'évolution de l'état de santé du patient. Nous appelons cette autre approche ***pairwise ground truth***.

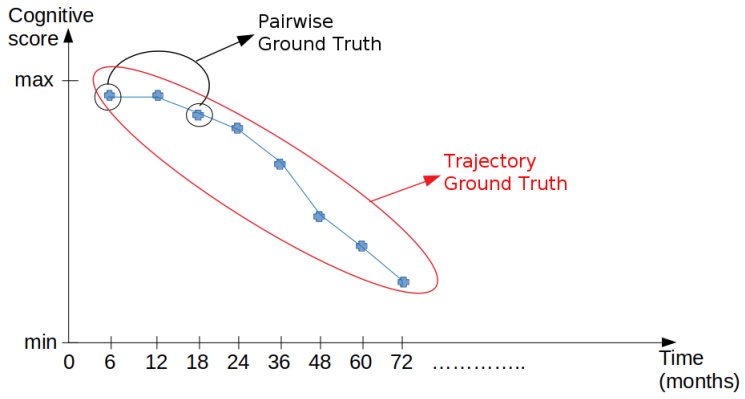


FIGURE 5.2 – Visites prises en compte pour la construction de la vérité terrain d'un patient

5.2.2 Etiquetage des données avec la nouvelle vérité terrain

Nous avons réutilisé les 381 sujets de la base ADNI (maladie d'Alzheimer) que nous avons présenté dans la Section 4.3. Notre *pairwise ground truth* est basée sur le score obtenu au test MMSE, comme dans les chapitres précédents. Ce score peut varier entre 0 et 30, et a été utilisé dans la littérature pour définir les stades suivants d'évolution de la maladie d'Alzheimer [102] (Table 5.4) :

MMSE	30	29-26	25-21	20-11	10-0
Stade	Sain (0)	Hypothétique (1)	Léger (2)	Modéré (3)	Sévère (4)

TABLE 5.4 – Stades de la maladie d'Alzheimer en fonction du score MMSE [102]

Ainsi, pour nos 381 sujets, pris à la visite d'inclusion, les effectifs correspondant à chacun des stades de la maladie sont les suivants (Table 5.5) :

Stade	Sain (0)	Hypothétique (1)	Léger (2)	Modéré (3)	Sévère (4)
Nb de sujets	75	192	107	8	0

TABLE 5.5 – Répartition des sujets dans les différents stades de la maladie d'Alzheimer, pour la visite d'inclusion

Cette répartition en stades d'évolution de la maladie nous permet d'identifier les sujets chez qui la maladie d'Alzheimer se développe entre deux visites (sujets en déclin), et ceux chez qui elle ne se développe pas (sujets stables) (voir Figure 5.3).

On voit aussi que cette nouvelle vérité-terrain est basée sur une évolution ponctuelle de l'état du patient, et qu'il existe notamment des patients qui oscillent entre les stades Hypothétique et "Sain, certainement en fonction de leur état les jours de visites, des heures de ces visites, etc.

Nous voyons en revanche que dans l'ensemble, parmi les patients dans les stades Léger, Modéré, et Sévère, la maladie évolue malheureusement de manière irrémédiable vers la sévérité, avec des vitesses différentes d'un patient à l'autre.

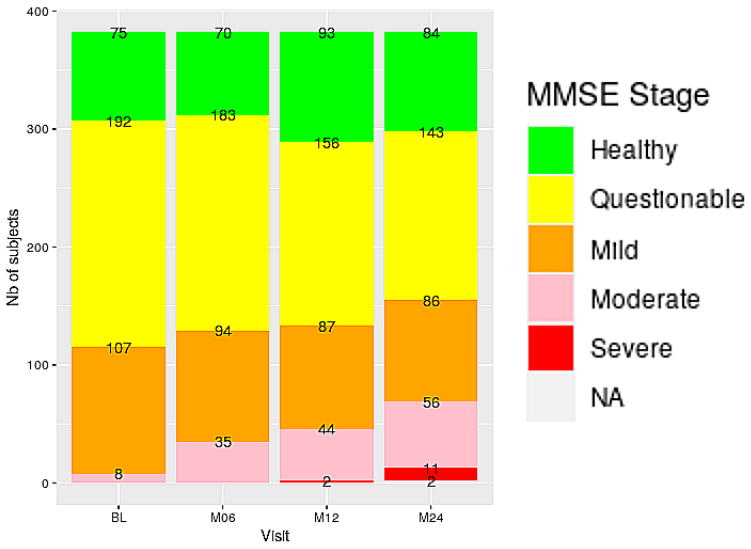


FIGURE 5.3 – Répartition des sujets en fonction de leur stade d'avancée de la maladie d'Alzheimer, donné par les groupes de score MMSE, pour les quatre visites d'intérêt

Nous nous intéressons donc aux changements de stades de la maladie entre deux visites d'intérêt, nommées i et $i + \delta$. Dans un premier temps, étant donné que la maladie d'Alzheimer est irréversible, nous avons filtré les paires pour lesquelles le stade à la visite $i + \delta$ était moins sévère que le stade à la visite i, et nous les avons supprimées de notre jeu de données, en considérant que les variations dans le score MMSE qui définit l'état de ces patients étaient dues à des phénomènes ponctuels, comme expliqué plus haut. Après avoir testé plusieurs protocoles pour la sélection des paires entrant dans notre jeu de données, nous avons également écarté de l'expérience les paires pour lesquelles les sujets sont déjà malades (MMSE inférieure à 26) mais dont le stade n'a pas évolué entre les deux visites d'intérêt. En effet, c'est ce protocole qui nous a permis d'avoir les meilleurs résultats après l'entraînement du modèle. Il est possible que cela soit dû au fait que nous supprimons ainsi un cas particulier de sujets, qui sont les sujets malades mais dont l'état reste stable, donc pour lesquels l'appellation "en déclin" ne convient pas vraiment.

La Table 5.6 résume la répartition des sujets en fonction du stade à la visite i et du stade à la visite $i + \delta$. On y voit en vert la répartition des paires incluses dans la classe "Stable", et en rouge la répartition des paires incluses dans la classe "Déclin".

Les paires incluses dans la classe Stable sont celles pour lesquelles le score MMSE de la visite i et celui de la visite $i + \delta$ est entre 30 et 26. Strictement parlant, on pourrait considérer que les sujets passant du stade Sain au stade Hypothétique sont en fait en train de décliner, cependant étant donné que le stade Sain ne contient qu'une seule valeur de MMSE, nous avons choisi de considérer pour notre expérience les stades Sains et Hypothétique comme un seul stade.

Les paires incluses dans la classe Déclin sont, d'une part celles pour qui le score MMSE de la visite i est entre 30 et 26 et le score MMSE de la visite $i + \delta$ est inférieure à 26, mais aussi celles pour lesquelles le stade évolue de léger à modéré, de modéré à sévère, ou de léger à sévère. Nous obtenons ainsi 1065 paires négatives (classe Stable) et 443 paires positives (classe Déclin).

Visite i / **Visite $i + \delta$**	**Sain (0)**	**Hypothétique (1)**	**Léger (2)**	**Modéré (3)**	**Sévère (4)**
Sain (0)	249	251	2	0	0
Hypothétique (1)	197	619	102	0	0
Léger (2)	8	191	289	29	0
Modéré (3)	0	26	177	85	0
Sévère (4)	0	1	18	22	2

TABLE 5.6 – Répartition des paires de visites, dans la classe Stable (en vert) ou la classe Déclin (en rouge), en fonction de la transition de stade entre une visite i et une visite $i + \delta$. Les cellules en gris correspondent aux transitions de stades que nous avons considéré comme aberrants (étant donné que la progression de la maladie d'Alzheimer ies irréversible), et que nous avons donc supprimé du jeu de données. Les cellules en blanc correspondent à des paires de visites où le sujet est déjà malade, mais reste stable, et que nous avons également écartées pour cette expérience.

5.2.3 Entraînement et résultats

Pour ces nouvelles expériences avec Multimodal3DSiameseNet (pour rappel, architecture décrite dans la Figure 4.9), nous avons séparé le jeu de données au niveau des sujets, pour que toutes les paires d'un même sujet soient dans un seul sous-ensemble de données. Ainsi, nous supprimons le risque d'un biais qui pourrait être introduit si le cerveau d'un sujet du jeu de validation avait déjà été fourni au modèle durant l'entraînement.

Le jeu de données est découpé en 60% pour l'entraînement (229 sujets), 20% pour la validation (76 sujets), et 20% pour le test (76 sujets). Nous avons entraîné notre modèle multimodal en suivant un protocole de validation croisée à 3 échantillons (*folds*). La Table 5.7 ci-dessous présente la répartition des paires entre nos deux classes :

	Echantillon 1		**Echantillon 2**		**Echantillon 3**	
	Négatifs	Positifs	Négatifs	Positifs	Négatifs	Positifs
Entraînement (228 sujets)	646	256	644	275	608	274
Validation (76 sujets)	203	100	205	81	241	82
Test (76 sujets)	216	87	216	87	216	87

TABLE 5.7 – Répartition des paires de visites en deux classes, pour les trois échantillons *folds*) de la validation croisée

Le modèle a été entraîné durant 30 *epochs*. Notre validation croisée comporte trois *run*. Le modèle a convergé pour deux *folds*, mais l'apprentissage n'a pas abouti pour l'entraînement avec le troisième *fold* (la valeur de la fonction de coût sur les données de validation n'a jamais diminué). Cela est très certainement dû à la taille de notre jeu de données d'apprentissage, et au tirage aléatoire des données dans chaque échantillon. Par honnêteté, nous n'avons pas changé le tirage aléatoire des échantillons. Il est donc probable que nous ayons eu un phénomène de sur-apprentissage pour le troisième *fold*.

Nous obtenons sur le jeu de test une *accuracy* moyenne de 88%, un score F1 moyen de 0.75, et une AUC moyenne de 0.93. Ces résultats, moins bons que ceux obtenus avec la *long-term trajectory ground truth* (voir Chapitre 4), peuvent être dus au fait que nous avons dû réduire notre jeu de données en filtrant les paires (suppression de 760 paires), ce qui a conduit à un fort déséquilibre de nos deux classes. Nous n'avons pas fait d'expériences additionnelles en filtrant autrement les paires de visites, ce qui nous aurait permis d'avoir un jeu de données plus grand. Il pourrait être interessant de faire ces expériences dans le futur.

Ces résultats soulèvent aussi une autre question : il est également possible que les seuils de scores MMSE utilisés pour déterminer les différents stades de la maladie ne soient pas optimaux.

En effet, d'après les résultats d'inférence obtenus avec la validation croisée (voir Figure 5.4), la majorité des faux positifs sont les sujets qui sont au stade 1 de la maladie (présence hypothétique de la maladie d'Alzheimer) à la première visite, et qui restent dans le même stade à la deuxième visite de la paire. La majorité des faux négatifs sont les sujets qui sont au stade 1 à la première visite, et passent au stade suivant (maladie légèrement développée) à la deuxième visite de la paire. Dans la littérature, la valeur MMSE seuil pour diagnostiquer la maladie d'Alzheimer est très controversée, entre 26 et 23 [107, 108]. Cela peut avoir conduit à une vérité terrain ne représentant pas correctement le déclin des patients appartenant au stade 1 de la maladie d'Alzheimer.

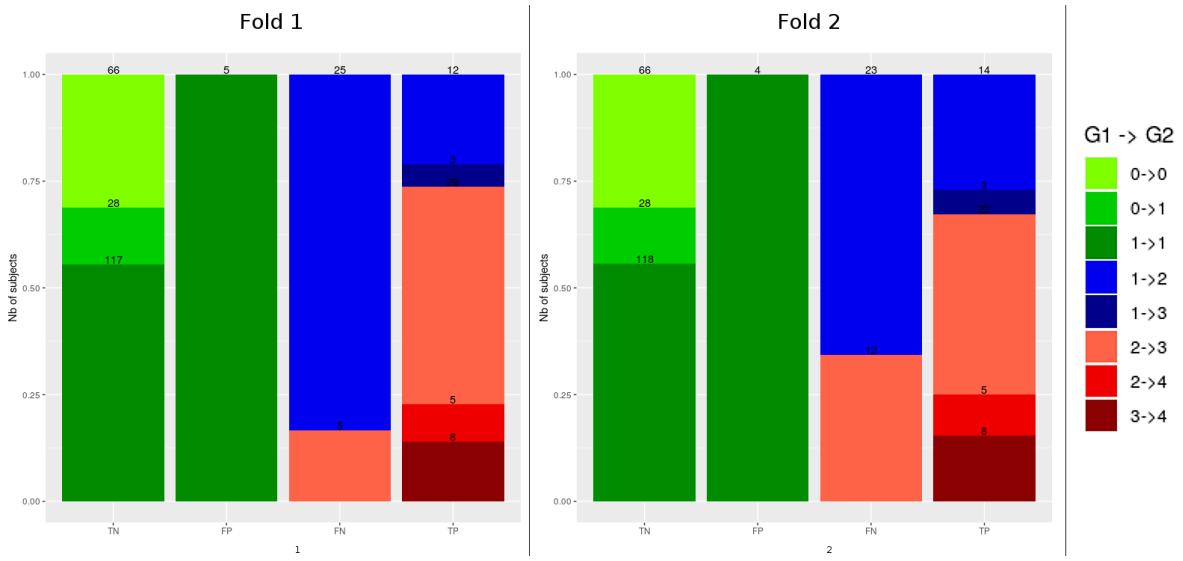


FIGURE 5.4 – Répartition des changements de stades de la maladie, chez les sujets bien ou mal classés par le modèle, pour les 2 premiers échantillons de la validation croisée. G1 : groupe MMSE à la visite i. G2 : groupe MMSE à la visite $i + \delta$. TN : Vrais négatifs, FP : Faux positifs, FN : Faux négatifs, TP : Vrais positifs.

Nous avons également étudié l'influence de la durée de l'intervalle entre les deux visites d'une paire sur la performance du modèle. La comparaison des scores F1 dans la Table 5.8 ci-dessous montre qu'il y a peu de différences entre 6, 12, et 18 mois d'écarts entre les deux visites, hormis le fait que les résultats obtenus pour des intervalles de 6 mois sont plus variables (écart-type de 0.08 contre 0.02 pour 12 et 18 mois).

En revanche, un intervalle de 24 mois permet d'obtenir des résultats sensiblement meilleurs. Dans le cas de la maladie d'Alzheimer, les écarts de déficience cognitive entre sujets stables et sujets en déclin se creusent au cours du temps. Il semble donc normal que les différences morphologiques et cliniques des sujets soient plus informatives avec un intervalle de 24 mois.

Intervalle	**6 mois**	**12 mois**	**18 mois**	**24 mois**
Nb de paires	477	502	256	273
F1	0.79 ($\pm$ 0.08)	0.84 ($\pm$ 0.02)	0.77 ($\pm$ 0.02)	0.90 ($\pm$ 0.02)

TABLE 5.8 – Variation de la précision, du rappel, et du score F1 en fonction de l'intervalle entre deux visites. Les valeurs sont les moyennes sur les 2 échantillons de la validation croisée, et les écart-types entre parenthèses

Si l'on compare les résultats obtenus par notre modèle Multimodal3DSiameseNet dans le chapitre précédent avec la *long-term trajectory ground truth*, et ceux obtenus dans ce chapitre avec la *pairwise ground truth*, on peut conclure que le choix de vérité terrain

le plus adapté pour caractériser l'évolution de l'état de santé des sujets est la stratégie *long-term trajectory ground truth*.

Les résultats expérimentaux avec les deux types de vérité-terrain sont recensés dans la Table 5.9. Avec la *long-term trajectory ground truth*, notre modèle obtient une AUC moyenne de 0.96, contre 0.93 avec la *pairwise ground-truth*. Un test statistique de Kruskall-Wallis sur les scores F1 a montré que la différence entre ces deux résultats est significative (p-value = $0.003231 < 0.05$)

	Test accuracy	**Test F1**	**Test AUC**
Trajectory ground truth	0.91 ($\pm$ 0.01)	0.90 ($\pm$ 0.01)	0.96 ($\pm$ 0.01)
Pairwise ground truth	0.88 ($\pm$ 0.01)	0.75 ($\pm$ 0.02)	0.93 ($\pm$ 0.01)

TABLE 5.9 – Résultats de l'entraînement de notre modèle multimodal, avec nos deux protocoles de création de vérité terrain. Les valeurs sont les moyennes sur les échantillons de la validation croisée, et les écart-types entre parenthèses.

La comparaison des résultats du chapitre précédent, aux résultats obtenus avec cette nouvelle approche *pairwise ground truth*, a montré que l'approche *long-term trajectory ground truth* permet mieux de modéliser l'évolution des maladies neuro-dégénératives. Toutefois, l'utilisation de la *long-term trajectory ground truth* n'est pas sans inconvénient : d'une part elle requiert l'accès à une étude longitudinale déjà conséquente en terme de taille de cohorte et de durée de suivi des sujets, et d'autre part elle requiert de passer par une étape de *clustering* (décrite par Bhagwat *et. al.*) pour l'étiquetage de nos données, ce qui peut également être source d'erreurs. On notera que, au vu de ces résultats, la stratégie *pairwise ground truth* est donc à considérer dans le cas d'un jeu de données de taille réduite.

5.3 Synthèse

Le travail de ce dernier chapitre nous a permis de mettre en évidence le compromis qui doit être fait entre obtenir la meilleure modélisation possible de l'évolution de la maladie, et utiliser un petit nombre de visites médicales.

En effet, il faut d'une part construire une vérité terrain pour la classification, en prenant en compte soit une évolution ponctuelle (deux-à-deux) de la maladie, soit une véritable trajectoire au long terme de la maladie. Nous avons montré ici que l'approche qui modélise le mieux l'évolution de la maladie d'Alzheimer chez les patients est celle qui est basée sur une trajectoire.

D'autre part, il faut choisir entre un modèle qui nécessite une séquence de visites médicales, ou un modèle légèrement moins performant mais qui n'utilise que des paires de visites en entrée. Si l'on ne dispose que de peu de visites par patients, notre modèle Multimodal3DSiameseNet est alors adapté à la tâche de prédiction de l'évolution cognitive des patients, et reste compétitive par rapport à l'approche RNN. Si l'on dispose de nombreuses visites pour tous les patients, on pourra utiliser l'approche RNN, comme proposée par Lee *et. al.*. Nous notons toutefois que l'architecture de Lee *et. al.* possède ses propres limites, en particulier le recours à des mesures calculées sur les IRM de cerveau, au lieu des images originelles.

Le chapitre suivant reprendra la discussion sur les intérêts et les limites de notre proposition de modélisation incluant des modèles de *deep learning*.

Chapitre 6

Discussion et Perspectives

Au cours de cette thèse, nous avons conçu et implémenté un modèle basé sur les réseaux de neurones profonds, pour la prédiction du déclin cognitif chez des patients potentiellement concernés par une maladie neuro-dégénérative. Cette prédiction est réalisée à partir de données multimodales (IRM de cerveau en 3D volumique, scores issus de tests cognitifs, et facteurs de risques) issues de deux visites médicales effectuées à des intervalles de temps quelconques. Ces travaux ont donné lieu à deux publications dans des conférences internationales, que nous résumerons dans la section qui suit. Après avoir rappelé les contributions de ces travaux de thèse, nous discuterons de leurs limites, et nous présenterons plusieurs perspectives d'amélioration à notre modèle.

6.1 Contributions et diffusion des travaux de thèse

Nos premiers résultats, présentés dans le **Chapitre 3, Section 3.2**, nous ont permis de montrer l'amélioration des résultats de classification de patients stables ou en déclin cognitif en utilisant des IRM de cerveaux entiers, en 3D volumique, au lieu de mesures issues de régions d'intérêt sélectionnées à partir d'un atlas. Ces résultats ont été publiés dans l'article "3D Siamese Net to Analyze Brain MRI" [96], présenté à la conférence internationale *10^{th} International Conference on Pattern Recognition Systems (ICPRS) 2019*. Cet article a gagné le prix *ex aequo* du meilleur article de la conférence.

Les résultats présentés dans le **Chapitre 3, Section 3.3**, nous ont permis de montrer d'une part la valeur ajoutée de la multimodalité par rapport à l'utilisation de données uniquement images ou uniquement cliniques, et d'autre part la plus grande robustesse du modèle multimodal en présence de données manquantes lors de l'inférence. Ces résultats ont été présentés à la conférence internationale *10^{th} International Conference on Image Processing Theory, Tools and Applications (IPTA) 2020*, dans notre article intitulé "Predicting Brain Degeneration with a Multimodal Siamese Network" [97].

Nous avons ensuite présenté dans le **Chapitre 4** de nouveaux résultats, en étendant l'utilisation de notre modèle à la prédiction du déclin à partir de deux visites i et $i + \delta$ quelconques (avec δ un intervalle de temps variable), montrant que notre modèle est capable de prédire l'évolution à long terme d'une maladie neuro-dégénérative chez un sujet, même à partir de visites médicales précoces. Enfin, nous avons montré que notre modèle, entraîné pour la maladie d'Alzheimer, est adaptable à d'autres pathologies neuro-dégénérative après une étape de *fine-tuning*. Cette preuve de concept a été réalisée en étudiant le déclin cognitif dans le cas de la maladie de Parkinson. Ces travaux ont fait l'objet d'un article long, en cours de soumission au journal *Computers in Biology and Medicine*.

Enfin, dans le **Chapitre 5**, nous avons montré la compétitivité des réseaux siamois par rapport aux RNN pour évaluer l'évolution d'une maladie sur un patient. En effet, comme notre modèle utilise l'architecture siamoise et non récurrente, il ne requiert que deux visites médicales comme données d'entrées. Cette caractéristique de notre modèle est importante car le fait de n'avoir besoin que de deux visites au lieu d'une séquence de visites est un avantage dans le milieu médical, étant donné le coût important des études de cohortes longitudinales.

6.2 Limites des travaux

Les modèles d'apprentissage profond sont souvent considérés comme des "boîtes noires", et l'explicabilité de l'apprentissage et des prédictions de ces modèles est un champ de recherche à part entière. Ainsi, une des limites de notre Multimodal3DSiameseNet est l'interprétation des prédictions du modèle, en particulier pour l'utilisation des données d'imagerie médicale. En effet, nous obtenons une prédiction sur l'évolution cognitive du patient, mais nous ne pouvons pas, en l'état actuel de nos travaux, l'associer à des zones spécifiques du cerveau qui seraient plus affectées que d'autres.

A titre d'exemple, pour l'utilisation de notre modèle avec la base de données sur la maladie de Parkinson, nous aimerions être certains qu'une des zones du cerveau ayant eu une grande importance pour la prédiction soit la substance noire, dont la dégradation est associée aux symptômes moteurs de cette maladie.

Plus généralement, dans ce contexte médical, il est crucial de pouvoir proposer aux médecins une explication concernant la prédiction du modèle. Pour résoudre ce problème, nous avons essayé d'utiliser des modèles tels que Grad-CAM [109], pour la visualisation des *features* d'intérêt conduisant aux prédictions à partir des images, mais nous n'avons pas obtenu de résultats satisfaisants.

Nous avons également utilisé une stratégie basée sur les auto-encodeurs pour la récupération d'une "carte de changements" entre les deux visites prises en compte (similaire aux stratégies utilisées dans [110, 111]), mais une fois de plus ces expériences n'ont pas été concluantes.

En ce qui concerne la taille des jeux de données utilisés, l'entraînement des réseaux de neurones profonds repose sur l'utilisation de très grands jeux de données. Or, dans notre contexte d'étude de cohortes de patients, il est difficile d'obtenir les données patients, pour des raisons de coût, des raisons logistiques, et des problèmes liés à l'acquisition de données privées. Cette difficulté à obtenir des données a restreint nos possibilités d'expérimentation. Ainsi, nous n'avons pas pu utiliser de données issues d'encéphalogrammes (EEG) dans notre modèle car ces données n'ont pas été récoltées durant le protocole des bases de données que nous avons utilisées. Pourtant, il aurait été intéressant d'ajouter cette modalité à notre modèle, car l'EEG est un examen souvent réalisé pour étudier les maladies neurologiques.

De même, nous avons tenté d'entraîner un module pour les données IRM fonctionnelle, en utilisant les réseaux récurrents convolutifs (Conv-RNN, voir Section 1.2.6 de l'état de l'art, et Annexe B pour une proposition d'architecture) pour traiter ces données, mais nous n'avons pas pu obtenir de résultats concluants à cause de la faible quantité de données de ce type dans les bases utilisées.

Enfin, on pourrait se poser la question de l'obtention d'un pronostique sur l'état de santé d'un patient à partir d'une seule visite. Dans ce cas, il serait indispensable d'avoir recours à un "cerveau modèle", en entrée de la seconde branche de notre modèle siamois. Cette question du diagnostique ou pronostique pour un patient à partir d'un seul point de temps, et utilisant une ou plusieurs images de référence (issues d'un sujet sain) a été discutée dans l'article de Li *et. al.* [106] présenté dans le chapitre précédent (vérité terrain alternative).

Dans cet article, en plus de présenter une stratégie *pairwise* de vérité terrain, les auteurs utilisent un réseau siamois convolutif 2D pour l'identification du stade d'évolution de deux maladies (arthrite du genoux, et rétinopathie du prématuré) à partir d'un score de distance calculé par le modèle. Ainsi, ce modèle est entraîné pour la classification des sujets dont la maladie évolue entre deux visites médicales, contre ceux qui restent au même stade. Par la suite, les auteurs montrent que le score de distance obtenu à la sortie du modèle est corrélé avec la sévérité de la maladie. A l'étape d'inférence, seule une image d'un patient inconnu est utilisée, et comparée à une ou plusieurs images de référence (images de sujets sains) à l'aide du réseau siamois, et la valeur du score de distance est utilisée pour inférer le stade de la maladie.

Dans notre contexte d'analyse d'images de cerveau, nous ne pouvons pas appliquer cette stratégie. En effet, elle repose essentiellement sur la comparaison à des images de sujets sains, faisant office de référence. Cela est possible pour les deux pathologies étudiées dans

cet article, cependant ce n'est pas le cas pour le cerveau car la variation morphologique inter individus est trop grande pour pouvoir se fier à un "cerveau modèle" [64].

6.3 Perspectives

A l'issue de ces travaux de thèse, nous proposons un modèle entraîné, permettant de prédire l'évolution du déclin cognitif d'un patient à partir de deux visites médicales. Ce modèle a été entraîné en prenant comme cas d'application la maladie d'Alzheimer, puis nous avons montré son adaptabilité à d'autres maladies neuro-dégénératives, par le biais du transfert d'apprentissage, en utilisant un petit jeu de données, constitué de sujets suivis dans le cadre d'une étude portant sur la maladie de Parkinson.

L'architecture de notre modèle en sous-modules permettrait également de l'adapter à des modalités supplémentaires, telles que l'IRM fonctionnelle ou les électro-encéphalogrames (EEG), à condition que les bases de données existantes se développent pour inclure plus de sujets et des modalités de données plus variées. Ainsi, il serait également possible d'utiliser une partie des paramètres de notre modèle entraîné, tout en modifiant son architecture pour inclure de modalités que nous n'avons pas inclues dans ces travaux.

De plus, notre modèle, qui a été conçu pour la prédiction du déclin cognitif pourrait également être utilisé pour la prédiction de rechute chez des patients suivis en addictologie. En effet, des changements neurologiques structuraux et fonctionnels ont été identifiés chez les patients dépendants, en sevrage ou non [112, 113]. L'évolution de ces changements pourrait être caractérisée par notre modèle de la même manière que l'évolution des changements dus aux pathologies neuro-dégénératives. Ainsi, une fois un jeu de données suffisant obtenu dans le cadre du projet ADDICTO/ICE, notre modèle pourra être testé pour la prédiction de rechute dans le cadre de pathologies addictives, après une étape de transfert d'apprentissage.

Enfin, une autre perspective d'amélioration de notre modèle, serait de combiner les contributions du réseau récurrent MildInt et celles de notre modèle Multimodal3DSiameseNet, par exemple en remplaçant les parties siamoises de notre modèle par des couches récurrentes. Pour cela, il faudrait utiliser l'architecture Conv-RNN, qui est une variation des RNN, introduits par Xingjian *et. al.* [43], utilisant des opérations de convolution pour traiter des séquences de données images. Afin de pallier au manque de données que nous avons rencontré lors de nos premières tentatives d'utiliser des CNN 3D pour ce problème, nous devrons certainement utiliser du transfert d'apprentissage, voir de l'adaptation de domaine (voir Etat de l'art, Section 1.3).

Par ailleurs, parallèlement aux travaux de cette thèse, j'ai eu l'occasion de m'intéresser à l'utilisation des réseaux siamois pour d'autres tâches, et dans d'autres domaines d'application. Ainsi, l'Annexe C présente une application des réseaux siamois en dehors du contexte médical de cette thèse. Dans ces travaux supplémentaires, cette architecture

a été utilisée dans le but de reconstituer des documents anciens. Pour cette application, nous utilisons les propriétés des réseaux siamois pour la comparaison deux-à-deux de fragments, dans le but de reconstituer des documents anciens. Ces travaux annexes ont donné lieu à une publication dans un volume spécial du journal *Pattern Recognition Letters* : *Pattern Recognition and Artificial Intelligence Techniques for Cultural Heritage* en 2020 [114], ainsi qu'à une publication dans le *workshop Pattern Recognition for Cultural Heritage* de la conférence ICPR 2020 [115], et sont donc détaillés dans ce manuscrit pour mettre en valeur ces contributions.

Conclusion

Les travaux réalisés dans le cadre de cette thèse ont porté sur la conception et l'implémentation d'un modèle d'apprentissage profond pour la prédiction du déclin cognitif de patients, à partir de données multimodales, en particulier les IRM de cerveau et les données cliniques. L'analyse de cette problématique a fait ressortir plusieurs sous-problèmes : premièrement l'analyse de l'évolution de l'état de santé d'un sujet au cours du temps, ensuite l'intégration et la fusion des différentes modalités de données, puis le cas particulier du traitement des IRM en tant qu'images 3D volumiques, et enfin l'adaptabilité de notre modèle à plusieurs pathologies neuro-dégénératives.

Un modèle, nommé Multimodal3DSiameseNet, de type **réseau de neurones profond** a été développé pour répondre à toutes ces problématiques :

- ✔ Ce modèle utilise une **architecture siamoise**, c'est-à-dire avec des paires de branches parallèles, pour la **comparaison de deux visites,** i **et** $i+\delta$ **d'un même sujet**. Ces deux visites peuvent être choisies arbitrairement, avec un intervalle de temps δ quelconque entre les deux. Ainsi, à partir de deux visites précoces, prises entre l'inclusion du patient et le suivi à 24 mois, notre modèle est capable d'**identifier les sujets en déclin cognitif au long-terme**. Nous avons ensuite comparé cette approche siamoise à un modèle de l'état de l'art basé sur les réseaux récurrents (RNN) et prenant en entrée une séquence de visites médicales. Nous avons montré d'une part que notre approche est compétitive en terme de performances de classification, et d'autre part qu'elle est avantageuse dans le contexte spécifique de l'étude de données médicales, où les données sont difficiles à obtenir, car elle ne requiert que deux visites par patient.
- ✔ Nous avons organisé notre modèle en **sous-modules, adaptés aux spécificités de chaque modalité**, pour extraire les *features* les plus informatives possibles. Le modèle permet ensuite une fusion intermédiaire des vecteurs de *features* provenant des différentes modalités, puis un apprentissage conjoint au travers de la dernière partie du réseau. Grâce à cet apprentissage conjoint, de nouvelles *features* sont créées, qui prennent également en compte les **corrélations entre les modalités**.
- ✔ Dans le sous-module dédié à la modalité IRM, nous avons utilisé des opérations de **convolutions 3D**, afin de mieux caractériser la morphologie du cerveau des patients grâce aux informations tridimensionnelles contenues dans ces images. Notre modèle calcule donc des *features* plus riches et moins biaisées que si nous avions utilisé une approche plus courante dans la littérature, basée sur la division du cerveau en régions d'intérêt, et nécessitant le recours à un atlas, ou "cerveau modèle".

- ✔ Après l'obtention de résultats de classification très satisfaisants pour une application à la maladie d'Alzheimer, nous avons montré que notre modèle est adaptable à d'autres pathologies du même type. Pour cela, nous avons réalisé un **transfert d'apprentissage**, à l'aide d'une base de données sur l'étude de la maladie de Parkinson.

Nous avons cependant pu identifier une limite principale à notre modèle. Comme tous les modèles basés sur les réseaux de neurones profonds, il est pour l'instant très difficile d'interpréter les prédictions fournies par notre modèle. Etant donné que nous travaillons ici dans un contexte médical, il est pourtant très souhaitable de pouvoir fournir une explication accompagnant ces prédictions.

Enfin, nous pouvons proposer quelques perspectives pour l'amélioration de notre modèle. Tout d'abord, l'implémentation et l'ajout de sous-modules supplémentaires, afin d'ajouter de nouvelles modalités, telles que les IRM fonctionnelles ou les EEG. Il serait également possible de remplacer les parties siamoises du modèle par des couches récurrentes, afin d'obtenir de meilleures prédictions pour les sujets ayant déjà plusieurs visites de suivi médical.

Finalement, à l'issue de ces travaux de thèse, nous proposons un modèle entraîné, permettant de prédire l'évolution du déclin cognitif d'un patient à partir de deux visites médicales. Ce modèle, ainsi que son implémentation, sont disponibles en *open source*. Il peut donc être facilement ré-utilisé, que ce soit pour les deux pathologies, Alzheimer et Parkinson, étudiées ici, ou pour des pathologies similaires par le biais d'un transfert d'apprentissage.

Bibliographie

[1] Rubinsztein, D. C. The roles of intracellular protein-degradation pathways in neurodegeneration. *Nature* **443**, 780–786 (2006).

[2] Maladies neurodégénératives. URL `https://www.santepubliquefrance.fr/maladies-et-traumatismes/maladies-neurodegeneratives`.

[3] Gillies, R. J., Kinahan, P. E. & Hricak, H. Radiomics : images are more than pictures, they are data. *Radiology* **278**, 563–577 (2016).

[4] O'Mahony, N. *et al.* Deep learning vs. traditional computer vision. In *Science and Information Conference*, 128–144 (Springer, 2019).

[5] ADNI | Alzheimer's Disease Neuroimaging Initiative. URL `http://adni.loni.usc.edu/`.

[6] Initiative, P. P. P. M. The parkinson progression marker initiative (ppmi) – experience with data and biospecimen access (p06.083). *Neurology* **78**, P06.083–P06.083 (2012). URL `https://n.neurology.org/content/78/1_Supplement/P06.083`. `https://n.neurology.org/content`.

[7] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).

[8] Maas, A. L., Hannun, A. Y., Ng, A. Y. *et al.* Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, vol. 30, 3 (Citeseer, 2013).

[9] Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data* **6**, 60 (2019).

[10] Kazeminia, S. *et al.* Gans for medical image analysis. *Artificial Intelligence in Medicine* 101938 (2020).

[11] Kumar, A. Overfitting & Underfitting Concepts & Interview Questions (2021). URL `https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/`.

[12] Introduction aux réseaux de neurones convolutifs. URL `https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks`.

[13] Szegedy, C. *et al.* Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9 (2015).

[14] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556* (2014).

[15] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).

[16] Paszke, A., Chaurasia, A., Kim, S. & Culurciello, E. Enet : A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv :1606.02147* (2016).

[17] Gundogdu, B. *Keyword Search for Low Resource Languages*. Ph.D. thesis (2017).

[18] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. & Shah, R. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, 737–744 (1994).

[19] Zagoruyko, S. & Komodakis, N. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4353–4361 (2015).

[20] Lin, S., Zhao, Z. & Su, F. Homemade ts-net for automatic face recognition. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, 135–142 (ACM, 2016).

[21] Koch, G., Zemel, R. & Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, vol. 2 (2015).

[22] Hadsell, R., Chopra, S. & LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 1735–1742 (IEEE, 2006).

[23] Bekuzarov, M. Losses explained : Contrastive Loss (2020). URL `https://medium.com/@maksym.bekuzarov/losses-explained-contrastive-loss-f8f57fe32246`.

[24] Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv :1706.05098* (2017).

[25] David, O. E. & Netanyahu, N. S. Deeppainter : Painter classification using deep convolutional autoencoders. In *International conference on artificial neural networks*, 20–28 (Springer, 2016).

[26] Ronneberger, O., Fischer, P. & Brox, T. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).

[27] Goodfellow, I. J. *et al.* Generative adversarial networks. *arXiv preprint arXiv :1406.2661* (2014).

[28] Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv :1511.06434* (2015).

[29] Hergott, M. Generative adversarial networks – key milestones and state of the art (2019). URL `https://www.kdnuggets.com/2019/04/future-generative-adversarial-networks.html`.

[30] Mirza, M. & Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv :1411.1784* (2014).

[31] Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232 (2017).

[32] Vaswani, A. *et al.* Attention is all you need. *arXiv preprint arXiv :1706.03762* (2017).

[33] Alammar, J. The illustrated transformer. URL `http://jalammar.github.io/illustrated-transformer/`.

[34] Parmar, N. *et al.* Image transformer. In *International Conference on Machine Learning*, 4055–4064 (PMLR, 2018).

[35] Carion, N. *et al.* End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229 (Springer, 2020).

[36] Dosovitskiy, A. *et al.* An image is worth 16x16 words : Transformers for image recognition at scale. *arXiv preprint arXiv :2010.11929* (2020).

[37] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *nature* **323**, 533–536 (1986).

[38] Yu, Y., Si, X., Hu, C. & Zhang, J. A review of recurrent neural networks : Lstm cells and network architectures. *Neural computation* **31**, 1235–1270 (2019).

[39] Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* **5**, 157–166 (1994).

[40] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).

[41] Cho, K. *et al.* Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078* (2014).

[42] Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* **45**, 2673–2681 (1997).

[43] Xingjian, S. *et al.* Convolutional lstm network : A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, 802–810 (2015).

[44] Donahue, J. *et al.* Decaf : A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, 647–655 (2014).

[45] Deng, J. *et al.* Imagenet : A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255 (Ieee, 2009).

[46] Choudhary, A., Tong, L., Zhu, Y. & Wang, M. D. Advancing medical imaging informatics by deep learning-based domain adaptation. *Yearbook of medical informatics* **29**, 129 (2020).

[47] Doersch, C., Gupta, A. & Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, 1422–1430 (2015).

[48] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T. & Efros, A. A. Context encoders : Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544 (2016).

[49] Fawcett, T. An introduction to roc analysis. *Pattern recognition letters* **27**, 861–874 (2006).

[50] Glen, S. URL `https://www.datasciencecentral.com/profiles/blogs/roc-curve-explained-in-one-picture`.

[51] Hosseini-Asl, E., Gimel'farb, G. & El-Baz, A. Alzheimer's disease diagnostics by a deeply supervised adaptable 3d convolutional network. *arXiv preprint arXiv :1607.00556* (2016).

[52] Maaten, L. v. d. & Hinton, G. Visualizing data using t-sne. *Journal of machine learning research* **9**, 2579–2605 (2008).

[53] Fowler, J. S., Volkow, N. D., Kassed, C. A. & Chang, L. Imaging the addicted human brain. *Science & practice perspectives* **3**, 4 (2007).

[54] MRI Basics. URL `https://case.edu/med/neurology/NR/MRI%20Basics.htm`.

[55] Sarraf, S. & Tofighi, G. Classification of alzheimer's disease using fmri data and deep learning convolutional neural networks. *arXiv preprint arXiv :1603.08631* (2016).

[56] Khvostikov, A., Aderghal, K., Benois-Pineau, J., Krylov, A. & Catheline, G. 3d cnn-based classification using smri and md-dti images for alzheimer disease studies. *arXiv preprint arXiv :1801.05968* (2018).

[57] Korolev, S., Safiullin, A., Belyaev, M. & Dodonova, Y. Residual and plain convolutional neural networks for 3d brain mri classification. In *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*, 835–838 (IEEE, 2017).

[58] Godenschweger, F. *et al.* Motion correction in mri of the brain. *Physics in Medicine & Biology* **61**, R32 (2016).

[59] Payan, A. & Montana, G. Predicting alzheimer's disease : a neuroimaging study with 3d convolutional neural networks. *arXiv preprint arXiv :1502.02506* (2015).

[60] Litjens, G. *et al.* A survey on deep learning in medical image analysis. *Medical image analysis* **42**, 60–88 (2017).

[61] Little, R. J. & Rubin, D. B. *Statistical analysis with missing data*, vol. 793 (John Wiley & Sons, 2019).

[62] Dziura, J. D., Post, L. A., Zhao, Q., Fu, Z. & Peduzzi, P. Strategies for dealing with missing data in clinical trials : from design to analysis. *The Yale journal of biology and medicine* **86**, 343 (2013).

[63] Campos, S. *et al.* Evaluating imputation techniques for missing data in adni : a patient classification study. In *Iberoamerican Congress on Pattern Recognition*, 3–10 (Springer, 2015).

[64] Giedd, J. N. *et al.* Child psychiatry branch of the national institute of mental health longitudinal structural magnetic resonance imaging study of human brain development. *Neuropsychopharmacology* **40**, 43–49 (2015).

[65] Liu, J. *et al.* Applications of deep learning to mri images : A survey. *Big Data Mining and Analytics* **1**, 1–18 (2018).

[66] Vieira, S., Pinaya, W. H. & Mechelli, A. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders : Methods and applications. *Neuroscience & Biobehavioral Reviews* **74**, 58–75 (2017).

[67] Hwang, S. & Kim, H.-E. Self-transfer learning for weakly supervised lesion localization. In *International conference on medical image computing and computer-assisted intervention*, 239–246 (Springer, 2016).

[68] Dou, Q. *et al.* Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks. *IEEE transactions on medical imaging* **35**, 1182–1195 (2016).

[69] Yang, X., Kwitt, R. & Niethammer, M. Fast predictive image registration. In *Deep Learning and Data Labeling for Medical Applications*, 48–57 (Springer, 2016).

[70] Milletari, F., Navab, N. & Ahmadi, S.-A. V-net : Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, 565–571 (IEEE, 2016).

[71] Zhou, T., Canu, S. & Ruan, S. Fusion based on attention mechanism and context constraint for multi-modal brain tumor segmentation. *Computerized Medical Imaging and Graphics* **86**, 101811 (2020).

[72] Zhou, T., Canu, S., Vera, P. & Ruan, S. Latent correlation representation learning for brain tumor segmentation with missing mri modalities. *IEEE Transactions on Image Processing* **30**, 4263–4274 (2021).

[73] Billones, C. D., Demetria, O. J. L. D., Hostallero, D. E. D. & Naval, P. C. Demnet : A convolutional neural network for the detection of alzheimer's disease and mild cognitive impairment. In *Region 10 Conference (TENCON), 2016 IEEE*, 3724–3727 (IEEE, 2016).

[74] Aderghal, K., Benois-Pineau, J. & Afdel, K. Classification of smri for alzheimer's disease diagnosis with cnn : Single siamese networks with 2d+ ? approach and fusion on adni. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 494–498 (ACM, 2017).

[75] Khosla, M., Jamison, K., Ngo, G. H., Kuceyeski, A. & Sabuncu, M. R. Machine learning in resting-state fmri analysis. *Magnetic resonance imaging* **64**, 101–121 (2019).

[76] Wen, D. *et al.* Deep learning methods to process fmri data and their application in the diagnosis of cognitive impairment : a brief overview and our opinion. *Frontiers in neuroinformatics* **12**, 23 (2018).

[77] Wang, L., Li, K., Chen, X. & Hu, X. P. Application of convolutional recurrent neural network for individual recognition based on resting state fmri data. *Frontiers in Neuroscience* **13**, 434 (2019).

[78] URL `http://fcon_1000.projects.nitrc.org/indi/abide/`.

[79] van Wingen, G. & Thomas, R. M. A hybrid 3dcnn and 3dc-lstm based model for 4d spatio-temporal fmri data : An abide autism classification study. In *OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging : Second International Workshop, OR 2.0 2019, and Second International Workshop, MLCN 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13 and 17, 2019, Proceedings*, vol. 11796, 95 (Springer Nature, 2019).

[80] Hinrichs, C. *et al.* Predictive markers for ad in a multi-modality framework : an analysis of mci progression in the adni population. *Neuroimage* **55**, 574–589 (2011).

[81] Eshaghi, A. *et al.* Classification algorithms with multi-modal data fusion could accurately distinguish neuromyelitis optica from multiple sclerosis. *NeuroImage : Clinical* **7**, 306–314 (2015).

[82] Galveia, J. N., Travassos, A. & da Silva Cruz, L. A. An ophthalmology clinical decision support system based on clinical annotations, ontologies and images. In *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)*, 94–99 (IEEE, 2018).

[83] Xu, T., Zhang, H., Huang, X., Zhang, S. & Metaxas, D. N. Multimodal deep learning for cervical dysplasia diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 115–123 (Springer, 2016).

[84] van der Burgh, H. K. *et al.* Deep learning predictions of survival based on mri in amyotrophic lateral sclerosis. *NeuroImage : Clinical* **13**, 361–369 (2017).

[85] Bhagwat, N. *et al.* Modeling and prediction of clinical symptom trajectories in alzheimer's disease using longitudinal data. *PLoS computational biology* **14**, e1006376 (2018).

[86] Lee, G., Nho, K., Kang, B., Sohn, K.-A. & Kim, D. Predicting alzheimer's disease progression using multi-modal deep learning approach. *Scientific reports* **9**, 1–12 (2019).

[87] Ramachandram, D. & Taylor, G. W. Deep multimodal learning : A survey on recent advances and trends. *IEEE Signal Processing Magazine* **34**, 96–108 (2017).

[88] Gupta, A., Ayhan, M. & Maida, A. Natural image bases to represent neuroimaging data. In *International conference on machine learning*, 987–994 (2013).

[89] Bron, E. E. *et al.* Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural mri : The caddementia challenge. *NeuroImage* **111**, 562–579 (2015). URL `https://www.sciencedirect.com/science/article/pii/S1053811915000737`.

[90] Ghafoorian, M. *et al.* Transfer learning for domain adaptation in mri : Application in brain lesion segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 516–524 (Springer, 2017).

[91] van Norden, A. G. *et al.* Causes and consequences of cerebral small vessel disease. the run dmc study : a prospective cohort study. study rationale and protocol. *BMC neurology* **11**, 1–8 (2011).

[92] Vu, T. D., Yang, H.-J., Nguyen, V. Q., Oh, A.-R. & Kim, M.-S. Multimodal learning using convolution neural network and sparse autoencoder. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, 309–312 (IEEE, 2017).

[93] Chen, L. *et al.* Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis* **58**, 101539 (2019).

[94] Nguyen, X.-B., Lee, G. S., Kim, S. H. & Yang, H. J. Self-supervised learning based on spatial awareness for medical image analysis. *IEEE Access* **8**, 162973–162981 (2020).

[95] Niu, S., Liu, M., Liu, Y., Wang, J. & Song, H. Distant domain transfer learning for medical imaging. *IEEE Journal of Biomedical and Health Informatics* (2021).

[96] Ostertag, C., Beurton-Aimar, M. & Urruty, T. 3d-siamesenet to analyze brain mri. In *10th International Conference on Pattern Recognition Systems (ICPRS-2019)* (IET, 2019).

[97] Ostertag, C., Beurton-Aimar, M., Visani, M., Urruty, T. & Bertet, K. Predicting brain degeneration with a multimodal siamese neural network. In *2020 Tenth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, 1–6 (IEEE, 2020).

[98] Wenk, G. L. *et al.* Neuropathologic changes in alzheimer's disease. *Journal of Clinical Psychiatry* **64**, 7–10 (2003).

[99] Burns, A. & Iliffe, S. Alzheimer's disease. bmj 338, b158 (2009).

[100] Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B. & Spurek, P. Processing of missing data by neural networks. In *Advances in Neural Information Processing Systems*, 2719–2729 (2018).

[101] Vapnik, V., Golowich, S. E. & Smola, A. J. Support vector method for function approximation, regression estimation and signal processing. In *Advances in neural information processing systems*, 281–287 (1997).

[102] Perneczky, R. *et al.* Mapping scores onto stages : mini-mental state examination and clinical dementia rating. *The American journal of geriatric psychiatry* **14**, 139–144 (2006).

[103] DeMaagd, G. & Philip, A. Parkinson's disease and its management : part 1 : disease entity, risk factors, pathophysiology, clinical presentation, and diagnosis. *Pharmacy and therapeutics* **40**, 504 (2015).

[104] Medical, B. Medical gallery of blausen medical 2014. *WikiJournal of Medicine* **1**, 1–79 (2014).

[105] Donahue, J. *et al.* Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634 (2015).

[106] Li, M. D. *et al.* Siamese neural networks for continuous disease severity evaluation and change detection in medical imaging. *NPJ digital medicine* **3**, 1–9 (2020).

[107] Kim, S. Y. & Caine, E. D. Utility and limits of the mini mental state examination in evaluating consent capacity in alzheimer's disease. *Psychiatric Services* **53**, 1322–1324 (2002).

[108] Arevalo-Rodriguez, I. *et al.* Mini-mental state examination (mmse) for the detection of alzheimer's disease and other dementias in people with mild cognitive impairment (mci). *The Cochrane database of systematic reviews* CD010783 (2015). URL `https://europepmc.org/articles/PMC6464748`.

[109] Selvaraju, R. R. *et al.* Grad-cam : Visual explanations from deep networks via gradient-based localization (2016). `1610.02391`.

[110] Daudt, R., Le Saux, B., Boulch, A. & Gousseau, Y. Détection dense de changements par réseaux de neurones siamois (2018).

[111] Dunnhofer, M. *et al.* Siam-u-net : encoder-decoder siamese network for knee cartilage tracking in ultrasound images. *Medical Image Analysis* **60**, 101631 (2020).

[112] Suckling, J. & Nestor, L. J. The neurobiology of addiction : the perspective from magnetic resonance imaging present and future. *Addiction* **112**, 360–369 (2017).

[113] Moeller, S. J. & Paulus, M. P. Toward biomarkers of the addicted human brain : Using neuroimaging to predict relapse and sustained abstinence in substance use disorder. *Progress in Neuro-Psychopharmacology and Biological Psychiatry* **80**, 143–154 (2018).

[114] Ostertag, C. & Beurton-Aimar, M. Matching ostraca fragments using a siamese neural network. *Pattern Recognition Letters* **131**, 336–340 (2020).

[115] Ostertag, C. & Beurton-Aimar, M. Using graph neural networks to reconstruct ancient documents. In Del Bimbo, A. *et al.* (eds.) *Pattern Recognition. ICPR International Workshops and Challenges*, 39–53 (Springer International Publishing, Cham, 2021).

[116] Wadhwani, M., Kundu, D., Chakraborty, D. & Chanda, B. Text extraction and restoration of old handwritten documents. *arXiv preprint arXiv :2001.08742* (2020).

[117] Rasheed, N. A. & Nordin, M. J. A survey of classification and reconstruction methods for the 2d archaeological objects. In *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, 142–147 (IEEE, 2015).

[118] Kleber, F. & Sablatnig, R. A survey of techniques for document and archaeology artefact reconstruction. In *2009 10th International Conference on Document Analysis and Recognition*, 1061–1065 (IEEE, 2009).

[119] da Gama Leitão, H. C. & Stolfi, J. A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 1239–1251 (2002).

[120] Cho, T. S., Avidan, S. & Freeman, W. T. A probabilistic image jigsaw puzzle solver. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 183–190 (IEEE, 2010).

[121] Mondal, D., Wang, Y. & Durocher, S. Robust solvers for square jigsaw puzzles. In *2013 International Conference on Computer and Robot Vision*, 249–256 (IEEE, 2013).

[122] Jin, S.-Y., Lee, S., Azis, N. A. & Choi, H.-J. Jigsaw puzzle image retrieval via pairwise compatibility measurement. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, 123–127 (IEEE, 2014).

[123] Paikin, G. & Tal, A. Solving multiple square jigsaw puzzles with missing pieces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4832–4839 (2015).

[124] Le, C. & Li, X. Jigsawnet : Shredded image reassembly using convolutional neural network and loop-based composition. *IEEE Transactions on Image Processing* (2019).

[125] Paumard, M.-M., Picard, D. & Tabia, H. Jigsaw puzzle solving using local feature co-occurrences in deep neural networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 1018–1022 (IEEE, 2018).

[126] Lin, M., Chen, Q. & Yan, S. Network in network. *arXiv preprint arXiv :1312.4400* (2013).

[127] Wu, Z. *et al.* A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[128] Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).

[129] Papyrology collection. URL `https://www.lib.umich.edu/collections/collecting-areas/special-collections-and-archives/papyrology-collection`.

[130] Brodersen, K. H., Ong, C. S., Stephan, K. E. & Buhmann, J. M. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, 3121–3124 (IEEE, 2010).

[131] Kelleher, J. D., Mac Namee, B. & D'arcy, A. *Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies* (MIT press, 2020).

[132] Franz, M. *et al.* Cytoscape. js : a graph theory library for visualisation and analysis. *Bioinformatics* **32**, 309–311 (2016).

Annexes

Annexe A : Détails des données cliniques utilisées dans les bases de données ADNI et PPMI

Variable	Nom complet	Catégorie	Description
APOE4	allèles APOE4	Génotype	Nombre d'allèles 4 du gène APOE
AV45	Florbetapir	Biomarqueur	Niveau de AV45 dans le cervellet
PIB	Pittsburgh Compound B marqué au carbone 11	Biomarqueur	Niveau de PIB dans le cortex frontal, le gyrus cingulaire antérieur, le précunéus, et le cortex pariétal
ABETA	peptide beta-amyloïde	Biomarqueur	Niveau de peptide beta-amyloïde dans le liquide cérébro-spinal
TAU	protéine Tau	Biomarqueur	Niveau de protéine Tau dans le liquide cérébro-spinal
PTAU	protéine Tau phosphorylée	Biomarqueur	Niveau de protéine Tau phosphorylée dans le liquide cérébro-spinal
FDG	fluorodesoxyglucose marqué au fluor 18	Biomarqueur	Niveau de FDG dans les lobes angulaire, temporal, et le gyrus cingulaire postérieur
LDELTOTAL	Logical Memory - Delayed Recall	Score neuro-cognitif	Test d'apprentissage verbal mesurant la mémoire différée
RAVLT	Rey Auditory Verbal Learning Test	Score neuro-cognitif	Test mesurant la capacité à encoder, combiner, stocker et récupérer des informations verbales dans la mémoire immédiate
MMSE	Mini-Mental State Examination	Score neuro-cognitif	Test d'orientation, attention, mémoire, langage, et compétences visuo-spatiales
CDRSB	Clinical Dementia Rating Scale	Score neuro-cognitif	Test de mémoire, orientation, jugement, résolution de problèmes, et autres capacités affectées par la maladie d'Alzheimer
FAQ	Functional Activities Questionnaire	Score neuro-cognitif	Test mesurant le fonctionnement pour des activitées quotidiennes
ADAS	Alzheimer's Disease Assessment Scale	Score neuro-cognitif	Mesure du dysfonctionnement cognitif
TRABSCOR	Trails B	Score neuro-cognitif	Mesure de la vitesse psychomoteure, de la recherche visuelle, et de l'attention
DIGITSCOR	Digit Symbol Substitution	Score neuro-cognitif	Mesure de la vitesse moteure, de l'attention, et des fonctions de perception visuelle

TABLE 6.1 – Détail des données cliniques utilisées avec notre modèle pour les sujets de la base ADNI

Variable	Nom complet	Catégorie	Description
APOE4	allèle APOE4	Génotype	Nombre d'allèles 4 du gène APOE
HY	Catégories de Hoehn & Yahr	Score neuro-cognitif	Evalue la progression de la maladie de Parkinson
NHY	Echelle de Hoehn & Yahr	Score neuro-cognitif	Evalue la progression de la maladie de Parkinson
UPSIT	University of Pennsylvania Smell Identification Test	Score neuro-cognitif	Détection de problèmes d'odorat
HVLT	Hopkins Verbal Learning Test	Score neuro-cognitif	Mesure l'apprentissage verbal et la mémoire
LNS	Letter-Number Sequencing	Score neuro-cognitif	Mesure la capacité à traîter et ré-organiser l'information à court-terme
QUIP	Questionnaire for Impulsive-Compulsive Disorders in Parkinson's Disease	Score neuro-cognitif	Mesure les comportements impulsifs et compulsifs
SCOPA	Scales for Outcomes in Parkinson's Disease	Score neuro-cognitif	Evalue les symptômes moteurs de la maladie de Parkinson
STAI	State-Trait Anxiety Inventory	Score neuro-cognitif	Mesure l'anxiété
UPDRS	Unified Parkinson Disease Rating Scale	Score neuro-cognitif	Evalue la progression de la maladie de Parkinson

TABLE 6.2 – Détail des données cliniques utilisées avec notre modèle pour les sujets de la base PPMI

Annexe B : Proposition d'architecture de sous-module pour l'analyse d'IRM fonctionnelles

Durant les travaux de thèse, nous avons réfléchi à une architecture de modèle siamois prenant en entrée les IRM fonctionnelles de cerveaux des patients. Les IRM fonctionnelles peuvent être considérées comme des séquences d'images, dont chaque image est une IRM de cerveau, donc est en 3D volumique. Ainsi, leur utilisation pose à la fois le problème de l'analyse d'images en trois dimensions, et le problème de l'analyse de séquences temporelles.

Nous avons donc conçu un réseau de neurones profond (Figure 6.1), très similaire dans son architecture globale à notre modèle 3DSiameseNet, dans lequel les opérations de convolutions sont remplacées par des couches appelées *Conv3D-LSTM*. Ces couches, basées sur les travaux de Xingjian *et. al.* [43] (voir Section 1.2.6 de l'Etat de l'art, page 34), sont des couches RNN de type LSTM, dans lesquelles les opérations de produits matriciels sont remplacées par des opérations de convolutions 3D, de sorte à pouvoir traiter des entrées en 3D.

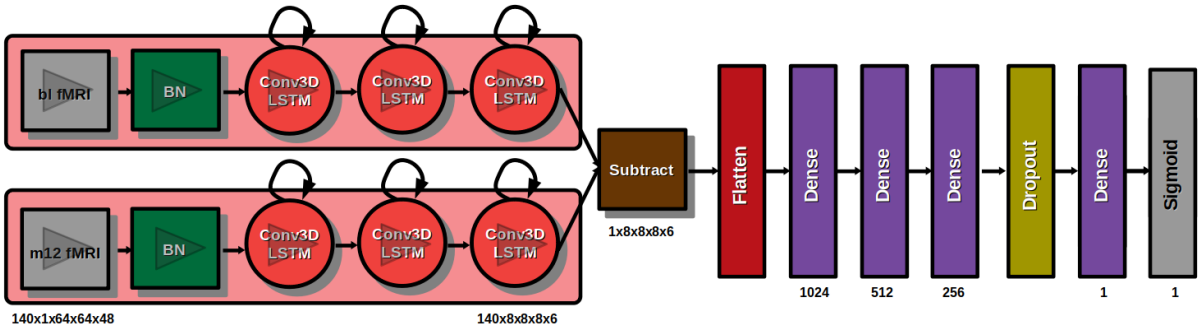


FIGURE 6.1 – Proposition d'architecture de sous-module pour l'analyse d'IRM fonctionnelles, utilisant les Conv-RNN pour traiter chaque IRM fonctionnelle en tant que séquence d'images 3D

Ce réseau a été implémenté (code correspondant disponible à cette adresse : *https ://-github.com/CeciliaOstertag/4DSiameseNet*), et nous avons cherché à le tester sur un jeu de données d'IRM fonctionnelles issues de la base ADNI. Cendant, nous avons été limités par le fait que peu de sujets de cette base ont passé une IRM fonctionnelle, ce qui ne nous a permi d'obtenir que 182 sujets (118 stables, et 65 en déclin cognitif). Avec un jeu de données si petit, comparativement à la taille et au nombre de paramètres du modèle que nous avons proposé, nous n'avons pas réussi à entraîner ce modèle pour notre tâche de classification.

Nous pouvons tout de même supposer que ce modèle est approprié à l'analyse d'IRM fonctionnelles de cerveaux, et pourrait être entraîné avec succès à condition de disposer d'un jeu de données plus grand, ou de pouvoir mettre en oeuvre une adaptation de domaine efficace à partir d'une base de données source de grande taille.

Annexe C : Utilisation de l'architecture siamoise pour la reconstruction de documents anciens

Au cours de mes travaux de thèse, j'ai eu l'occasion de participer à des discussion autour de l'utilisation des méthodes d'apprentissage profond dans le domaine de l'héritage culturel, et en particulier pour la reconstruction de documents anciens. J'ai ainsi pu proposer l'approche des réseaux siamois pour cette seconde problématique, dans le cadre d'un projet parallèle à mes travaux de thèse.

En effet, les réseaux siamois ayant été conçus à l'origine pour évaluer des similarités entre échantillons appariés, ils peuvent être utilisés ici pour comparer deux-à-deux des fragments issus de documents anciens.

Ces travaux ayant fait l'objet de deux publications (dans le journal *Pattern Recognition Letters*, et dans un *workshop* de la conférence ICPR 2020), nous y consacrons ici une annexe bien plus fournie que les deux annexes précédentes.

C.1. Contexte

L'étude de documents et artefacts anciens est une source de connaissance sur les civilisations qui nous ont précédées. En effet, ils contiennent des informations concernant l'organisation économique, religieuse, et politique à l'époque de leur écriture. Malheureusement, les conditions de préservations de ces artefacts ont souvent été mauvaises, et de nos jours les archéologues ne peuvent exhumer que des fragments de documents, provenant de différents supports : papyri ou tessons de poteries (utilisés comme support d'écriture) par exemple. La restauration des documents à partir de ces fragments est une tâche titanesque, mais nécessaire pour pouvoir déchiffrer leur contenu.

Les techniques récentes de traitement d'image peuvent être appliquées à ce problème, par exemple pour la reconnaissance de texte [116] ou la reconstruction d'objets en 3D [117]. En particulier, les réseaux de neurones convolutifs peuvent être utilisés pour obtenir de meilleurs résultats que les méthodes traditionnelles de traitement d'image. Dans ces travaux, nous considérerons la tâche de reconstruction de documents anciens comme analogue à une résolution de puzzles [118], avec la difficulté additionnelle de ne pas pou-

voir utiliser la forme des fragments comme aide à l'assemblage, étant donné que les bords sont souvent dégradés par le temps. Plus de détails sur les spécificités de la reconstruction d'objets dans le cas d'artefacts anciens sont détaillés par da Gama Leitão *et. al.* [119], qui listent ainsi comme problèmes spécifiques l'érosion le long des lignes de fracture, les fragments manquants, et les fragments qui ne peuvent être assemblés que partiellement (voir Figure 6.2).

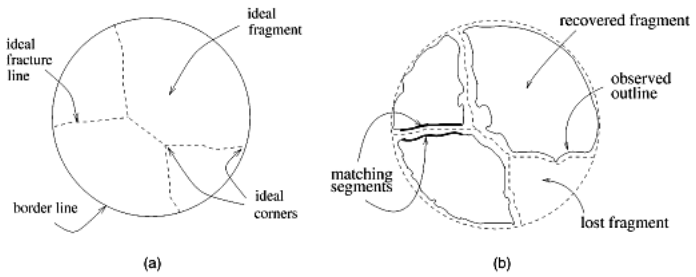


FIGURE 6.2 – Exemples de difficultés rencontrées pour la reconstruction d'anciens artefacts [119]

En dépit de ces limites, la texture et les écrits présents au niveau de ces lignes de fractures sont des informations qui peuvent être utilisées pour évaluer la probabilité d'assemblage entre fragments. Nous proposons donc une stratégie de reconstruction de documents anciens, basée sur les alignements locaux deux-à-deux entre fragments, et utilisant le *deep learning* pour ré-assembler ces documents.

C.2. Etat de l'art

En 2010, Cho *et. al.* [120] ont présenté un modèle basé sur les graphes pour résoudre le problème de résolution de puzzle. Leur approche utilise la somme du carré de la différence de couleur autour des zones de jonctions possibles comme métrique pour évaluer la compatibilité de deux fragments pour l'assemblage.

Pour réaliser un assemblage global, c'est-à-dire à partir de plusieurs fragments, ils utilisent une chaîne de Markov afin de spécifier des contraintes pour la reconstruction globale. Par la suite, d'autres métriques ont été proposées pour des applications similaires : le gradient de Mahalanobis [121], ou l'utilisation d'une métrique prenant en compte la similarité des contours ainsi que la similarité cosinus entre les histogrammes des deux fragments [122]. En 2015, Paikin *et. al.* [123] ont présenté un algorithme glouton (*greedy algorithm*) pour la résolution de puzzle à partir de morceaux dont on ne connaît pas l'orientation, et où certains morceaux sont manquants. Le point le plus important de leur travail est la recherche du morceau le plus adapté pour le début de l'algorithme. Enfin, en 2019 Le *et. al.* [124] ont proposé le modèle JigsawNet, basé sur l'utilisation d'un réseau de neurones convolutif pour la détection de compatibilité deux-à-deux entre morceaux, suivi par un algorithme de composition global utilisant les graphes.

Tous les modèles présentés ci-dessus ont été conçus pour des puzzles représentant des images naturelles, qui contiennent donc des informations sémantiques importantes (avant et arrière-plan, objets, personnages, ...), ainsi qu'une grande diversité de formes et de couleurs. Dans notre contexte de documents anciens, les fragments auxquels nous avons accès contiennent peu d'informations de ce type : en majorité, il s'agit de faibles gradients de couleur, d'inscriptions (écrits ou dessins), et de la texture du document lui-même. En effet, dans les exemples de résolution de puzzle donnés dans tous ces travaux de l'état de l'art, les zones les moins bien reconstruites dont les zones des images contenant peu d'informations sémantiques, comme le ciel, la mer, ou des étendues d'herbe. Un article de Paumard *et. al.* [125] nous a semblé plus proche de notre problème de reconstruction de documents anciens, car les auteurs cherchent à reconstruire des anciens tableaux ou des images d'objets anciens, à partir de morceaux d'images. Cependant, leur jeu de données est également composé majoritairement d'images contenant beaucoup d'informations sémantiques.

C.3. Premier modèle : Utilisation d'un réseau siamois pour l'assemblage de fragments deux-à-deux

Comme nous l'avons expliqué précédemment, il n'est pas suffisant de se baser sur la complémentarité de contours des fragments pour reconstruire des artefacts anciens. Dans certains cas, l'utilisation seule de la complémentarité de contours peut même introduire un biais lors de la reconstruction, étant donné que ces contours ont été érodés et ne correspondent donc plus aux cassures originelles. Pour notre premier cas d'application, des poteries d'Egypte antique servant de support d'écriture, appelées ostraca, nous nous sommes concentrés sur un assemblage deux-à-deux basé sur l'information de texture des fragments. La Figure 6.3 ci-dessous montre quelques exemples d'ostraca exhumés par des archéologues.

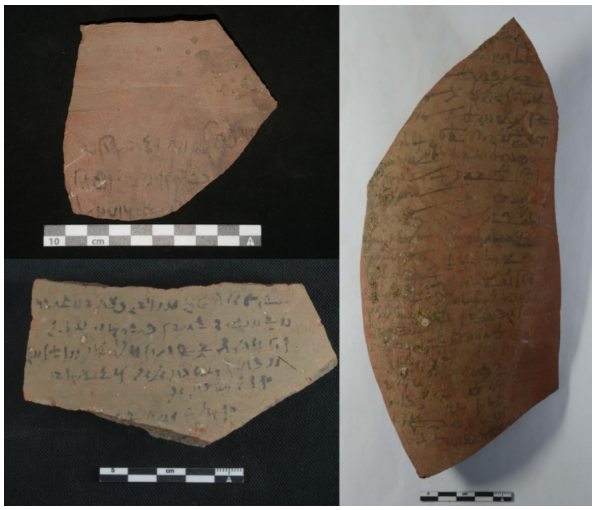


FIGURE 6.3 – Exemples d'ostraca, avec barres d'échelle en dessous

Notre première approche est donc basée sur l'utilisation d'un réseau siamois convolutif, entraîné pour classer des paires de fragments selon leur direction hypothétique d'assemblage (par exemple "Fragment 1 aligné sur la droite de Fragment 2"). Ici, les entrées du réseau de neurones qui représentent les fragments sont des images carrées, de taille 400 × 400 pixels, que nous appellerons par la suite "*patchs*". Les paires de *patchs* sont classées en cinq classes. D'une part, un label allant de 1 à 4 représente la direction d'alignement respectivement au premier *patch* de la paire. Ainsi nous avons les labels suivants : 1 = Droite (D), 2 = Gauche (G), 3 = Haut (H), 4 = Bas (B). D'autre part, pour les paires qui ne peuvent pas être assemblées, le label 0 sera utilisé (voir Figure 6.4).

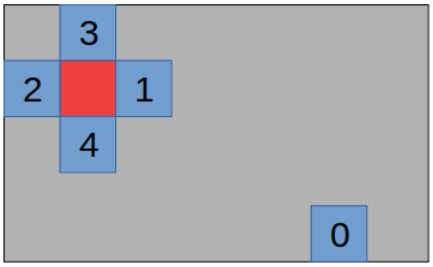


FIGURE 6.4 – Exemple de *patch* (en rouge) et de patchs qui seront testés contre lui, avec leur label associé

Ce modèle a été publié dans un volume spécial du journal Pattern Recognition Letters : *Pattern Recognition and Artificial Intelligence Techniques for Cultural Heritage* [114]. Le code correspondant est disponible à cette adresse : *https ://github.com/CeciliaOstertag/OstraNet*.

Architecture de notre modèle

L'architecture du réseau OstraNet, que nous avons proposé (voir Figure 6.5), est un réseau siamois convolutif 2D, dont chaque branche est constituée d'une succession de "blocs de convolution". Chaque bloc est lui-même composé de :

— Une couche de convolution 2D

— Une normalisation par *batch*

— Une activation de type *leaky ReLU*

— Une couche de *max pooling*

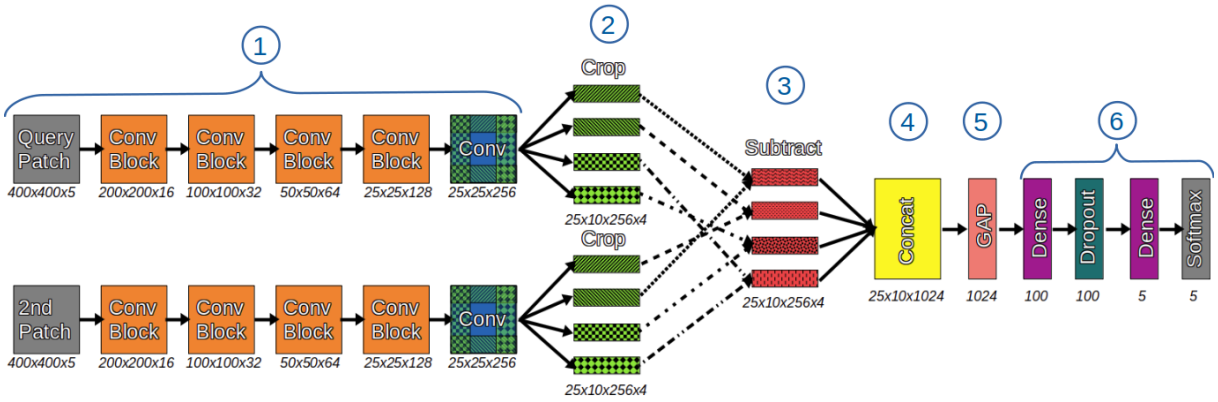


FIGURE 6.5 – Architecture de notre réseau siamois OstraNet, pour la comparaison deux-à-deux de fragments

Après la dernière couche de convolution, une bande de 10 pixels de large est extraite à partir du haut, du bas, de la gauche, et de la droite de l'image de sortie de chaque branche siamoise (Opération "Crop" (②) sur la Figure 6.5). Cette opération permet de restreindre les calculs aux quatre zones de jonctions possibles entre les deux *patchs*, afin de réduire le temps de calcul nécessaire durant l'entraînement du modèle. Nous obtenons ainsi 4 sous *feature maps* en sortie de chaque branche du réseau.

Ensuite, des "cartes de distance" sont créées, en soustrayant chaque sous *feature map* issue de la première branche à la *feature map* correspondante issue de la seconde branche (Opération "Subtract" (③) sur la Figure 6.5). Nous prenons ensuite la valeur absolue de ces soustractions. Ainsi notre modèle pourra prendre en compte les variations de texture et couleur entre les paires d'images au niveau des zones de jonctions possibles.

Ces quatre résultats sont ensuite concaténés, puis passent par une couche de *pooling* global par moyennage (*global average pooling*) [126] (Opération "GAP" (⑤) sur la Figure 6.5).

Enfin, le reste du réseau (couches ⑥ sur la Figure 6.5) est un classifieur classique, avec une activation de type *softmax* qui renvoie la probabilité d'appartenance à chacune des cinq classes.

Entraînement et résultats

Les données que nous utilisons pour l'entraînement de notre modèle sont des images RGB d'ostraca provenant d'Athribis en Egypte. Etant donné que ce jeu de données ne contient qu'un seul exemple de reconstruction, nous n'avions pas de vérité terrain données par des égyptologues pour notre tâche de reconstruction. Ainsi, pour créer nos propres vérités terrain nous avons retenu 30 images d'ostraca, que nous avons découpé en *patchs* de 400×400 (sans recouvrement). Pour chaque image originelle, nos obtenons donc un ensemble de *patchs* pour lesquels nous pouvons définir avec certitude qu'ils appartiennent au même ostraca. Nous avons ensuite créé notre jeu de données de paires, dont nous connaissons les différentes directions d'assemblage (ou l'absence d'assemblage), comme nous l'avons expliqué précédemment.

Nous avons entraîné le modèle en utilisant un premier jeu de données de 900 *patchs*, dont 100 ont été utilisés pour la validation. A chaque *epoch* les données d'entraînement ont été augmentées en réalisant des rotations de 90°et des permutations horizontales et verticales des images. Notre modèle a convergé après 150 *epochs*, avec une *accuracy* de 81% sur le jeu de données de validation.

Nous avons également entraîné notre modèle OstraNet avec un jeu de données plus grand, contenant 7000 *patchs* extraits de 160 images d'ostraca. Cependant, à cause de ressources mémoire limitées, nous avons dû réduire la taille des *patchs* à 200×200 pixels, et supprimer le premier bloc de convolution de notre modèle. Nous avons utilisé 6000 *patchs* pour l'entraînement, et 1000 pour la validation. A la fin de l'entraînement, nous avons obtenu une *accuracy* de 96% sur les données de validation.

Nous avons ainsi proposé un réseau de neurones siamois convolutif prenant en entrée une paire de *patchs*, et capable d'identifier l'existence d'un assemblage et la direction de l'assemblage entre les deux. Après nos bons résultats sur ces alignements deux-à-deux, nous avons proposé un *pipeline* pour une reconstruction globale à partir de plusieurs *patchs*.

Méthode de reconstruction globale

Notre méthode de reconstruction globale est basée sur un graphe dirigé construit dynamiquement à partir d'alignements deux-à-deux successifs. Dans ce graphe, les noeuds sont les *patchs*, et chaque noeud peut avoir au maximum quatre voisins, correspondant à chaque direction d'assemblage (voir Figure 6.6 pour un exemple de reconstruction). Chaque arête dirigée entre un noeud et son voisin est étiquetée avec la direction d'assemblage correspondant, ainsi qu'avec la probabilité associée à cet assemblage, telle que calculée par le modèle OstraNet.

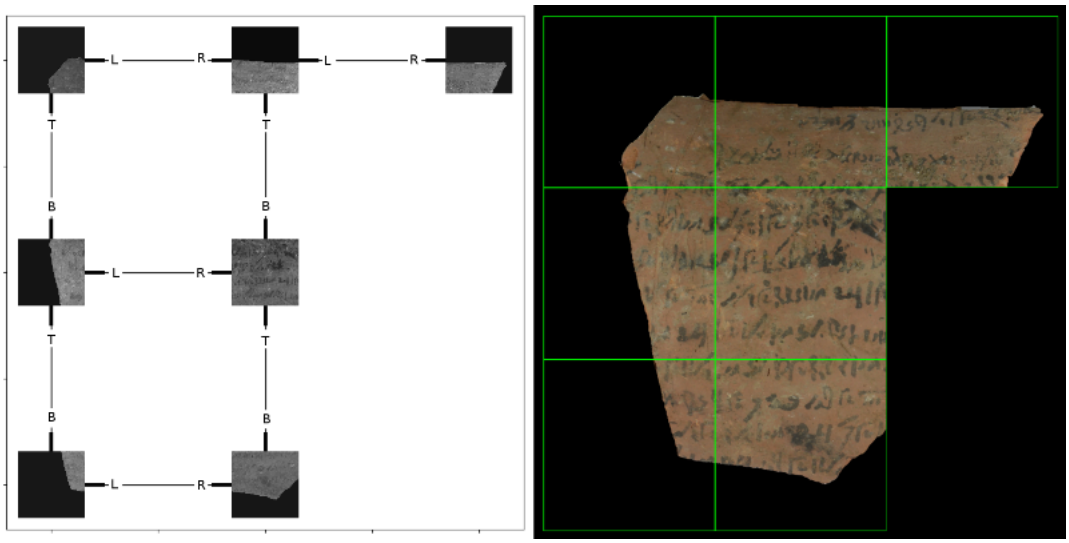


FIGURE 6.6 – Exemple de graphe associé à une proposition d'assemblage. Toutes les arêtes sont dirigées, avec l'étiquette correspondant à la direction d'assemblage (T : top, D : down, L : left, R : right)

A titre d'exemple, si l'on souhaite réassembler une image à partir de N *patchs*, le scénario est le suivant :

1. Un *patch* de la liste devient le *patch* "requête", qui ira dans la première branche du modèle OstraNet.
2. Chacun des *patchs* restant (*patch* "candidat") passera dans la seconde branche du réseau, de façon séquentielle.
3. Pour chaque paire, l'assemblage est considéré comme bon si la classe prédite n'est pas 0, et si la probabilité associée à la classe est supérieure à un seuil de 0.7.
4. Pour chaque bon assemblage trouvé, le graphe courant est dupliqué, et une nouvelle arête est ajoutée entre les deux noeuds correspondants. Cela permet de prendre en compte le fait qu'un même *patch* puisse avoir plusieurs *patchs* pour lesquels le modèle OstraNet prédit un assemblage correct, et ce pour la même direction. Ainsi, un nouveau graphe est créé pour chaque proposition d'assemblage.
5. Le même procédé est répété pour chaque *patch* dans la liste de *patchs* disponibles, jusqu'à ce que tous les *patchs* aient été utilisés en tant que "requête".

Enfin, pour éviter des calculs inutiles, si un noeud est déjà présent dans le graphe, seuls les voisins dans les directions qui n'ont pas déjà été identifiées vont être acceptés pour l'assemblage.

Lorsqu'un *patch* "candidat" ne peut être assemblé avec aucun des *patchs* déjà ajoutés au graphe courant, il est ajouté au graphe, mais en n'étant relié à aucun noeud déjà présent. Ce nouveau noeud est donc le point de départ d'une nouvelle composante connexe du graphe. Ainsi, les composantes fortement connexes du graphe correspondent à des propositions d'assemblages partiels.

Limites

Notre modèle siamois OstraNet est efficace pour la détection d'assemblage deux-à-deux, car il teste à la fois si l'assemblage est possible (*i. e.* si le label est différent de 0), et, le cas échéant, la direction de l'alignement (labels 1 à 4). Cependant, lorsque l'on cherche à reconstituer une image entière à partir d'une collection de *patchs*, nous obtenons des résultats moins concluants. Cela est dû au fait que l'assemblage correct entre deux *patchs* dans l'image d'origine n'est pas forcément celui qui sera identifié par notre modèle avec la plus haute probabilité.

A cause de ce problème, les erreurs s'accumulent à chaque itération lors de la reconstruction globale. En théorie, la solution de la reconstruction est incluse dans l'ensemble des propositions données par notre *pipeline* de reconstruction globale (autrement dit, un des graphes produits devrait correspondre à la reconstruction correcte), mais le nombre de graphes possibles qui sont générés par notre *pipeline* est trop important pour être analysé manuellement par un utilisateur.

Les limites de cette approche nous ont conduit à développer une approche légèrement différente, que nous avons appliquée à un autre jeu de données : les papyrus de la collection de l'Université du Michigan.

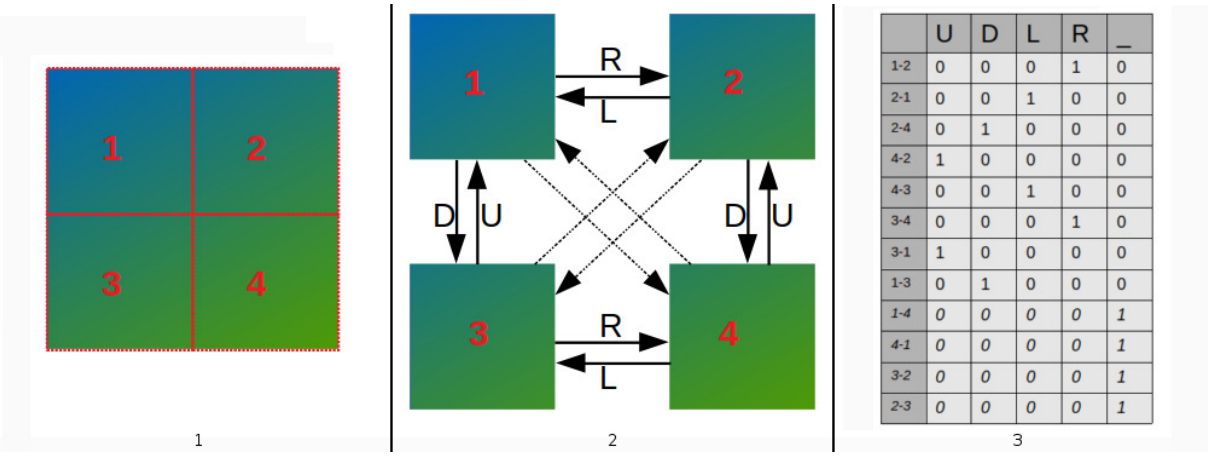
C.4. Deuxième modèle : Approche basée sur les graphes pour la comparaison simultanée de tous les fragments

Après avoir proposé une approche en deux temps pour la reconstitution d'images globales, nous proposons ici un modèle basé sur les réseaux de neurones pour graphes (*Graph Neural Networks (GraphNN)*) [127], qui permet en une seule passe d'évaluer tous les assemblages deux-à-deux possibles entre *patchs* et de produire un unique graphe contenant tous les assemblages possibles.

Les GraphNN sont utilisés pour travailler sur des données non euclidiennes, comme des graphes représentant des réseaux sociaux, des réseaux de citations, ou des graphes métaboliques par exemple [127]. Contrairement aux réseaux de neurones classiques, dans lesquels les filtres sont appliqués sur des suites de valeurs contiguës en 1D, 2D ou 3D, dans les GraphNN les filtres sont appliqués sur une séquence de noeuds adjacents, c'est-à-dire liés entre eux par au moins une arête. Dans le contexte de classification, ces architectures peuvent être utilisées pour différentes applications : classification des noeuds, des arêtes, voire des graphes entiers. Tous ces exemples servent de tutoriels dans la bibliothèque python torch-geometric [128], que nous avons utilisée pour implémenter notre propre modèle.

Dans notre cas, nos données (documents anciens sous la forme d'images) sont des données euclidiennes à l'origine, mais nous transformons chaque collection de *patchs* représentant une image en un graphe complet dirigé, dans lequel les noeuds sont les *patchs* (voir Figure 6.7). Les arêtes du graphe représentent ainsi les assemblages potentiels entre deux

patchs. Nous utilisons un graphe dirigé de façon à évaluer dans les deux sens la possibilité d'assemblage pour chaque paire de *patchs*. Ainsi, pour chaque paire de *patchs*, les deux patchs sont reliés par deux arêtes dirigées réciproques. Nous définissons donc notre problème de reconstitution de document comme une tâche de classification d'arêtes.



	U	D	L	R	_
1-2	0	0	0	1	0
2-1	0	0	1	0	0
2-4	0	1	0	0	0
4-2	1	0	0	0	0
4-3	0	0	1	0	0
3-4	0	0	0	1	0
3-1	1	0	0	0	0
1-3	0	1	0	0	0
1-4	0	0	0	0	1
4-1	0	0	0	0	1
3-2	0	0	0	0	1
2-3	0	0	0	0	1

FIGURE 6.7 – Exemple de transformation d'une image composée de quatre *patchs* en un graphe complet dirigé. 1 : Image d'origine, 2 : Graphe complet dirigé (les noeuds sont les *patchs* et les arêtes représentent les relations spatiales entre *patchs*), 3 : Labels (sous forme *one-hot* pour chacune des arêtes : Haut (H), Bas (B), Gauche (G), Droite (D), et aucun (_).

Ce modèle a été présenté au *workshop Pattern Recognition for Cultural Heritage* de la conférence ICPR 2021 [115]. Le code correspondant est disponible à cette adresse : *https ://github.com/CeciliaOstertag/AssemblyGraphNet*.

Architecture du modèle

Notre modèle, nommé AssemblyGraphNet, est un GraphNN convolutif, qui utilise les attributs des noeuds (tableaux de pixels représentant les *patchs* correspondants) pour prédire la classe des arêtes (assemblages potentiels de deux *patchs*). Pour cela, son architecture est composée deux parties : un modèle **Global**, et un modèle de **Comparaison Deux-à-deux**. Le modèle de **Comparaison Deux-à-deux** (voir Figure 6.8) est un réseau siamois avec une partie convolutive, qui prend en entrée les attributs de deux noeuds connectés, que nous appellerons classiquement "source" et "cible", et prédit un label pour l'arête connectant le noeud source au noeud cible. Comme dans le travail précédent, nous définissons cinq classes, qui correspondent à la direction d'assemblage du *patch* cible par rapport au *patch* source. Ces classes sont : 1 = Haut (H), 2 = Bas (B), 3 = Gauche (G), 4 = Droite (D), et 0 = aucun (_).

Etant donné que nous voulons tester les assemblages dans quatre directions, la première étape du réseau est de découper une bande de 40 pixels de large dans chaque direction, pour chaque *patch*.

Ensuite, ces bandes sont concaténées selon les zones d'assemblage possible (par exemple haut du *patch* source avec bas du *patch* cible).

Ces quatre résultats sont ensuite concaténés, puis le résultat de cette concaténation sert d'entrée à un CNN (voir Table 6.3 pour un détail de l'architecture de ce CNN), suivi par quatre couches entièrement connectées de tailles de plus en plus petites.

Enfin, une activation de type *softmax* renvoie une probabilité d'appartenance pour chacune de nos cinq classes. La classe qui sera finalement prédite pour l'arête allant du *patch* source au *patch* cible sera la classe pour laquelle la probabilité est la plus grande, parmi les cinq classes possibles.

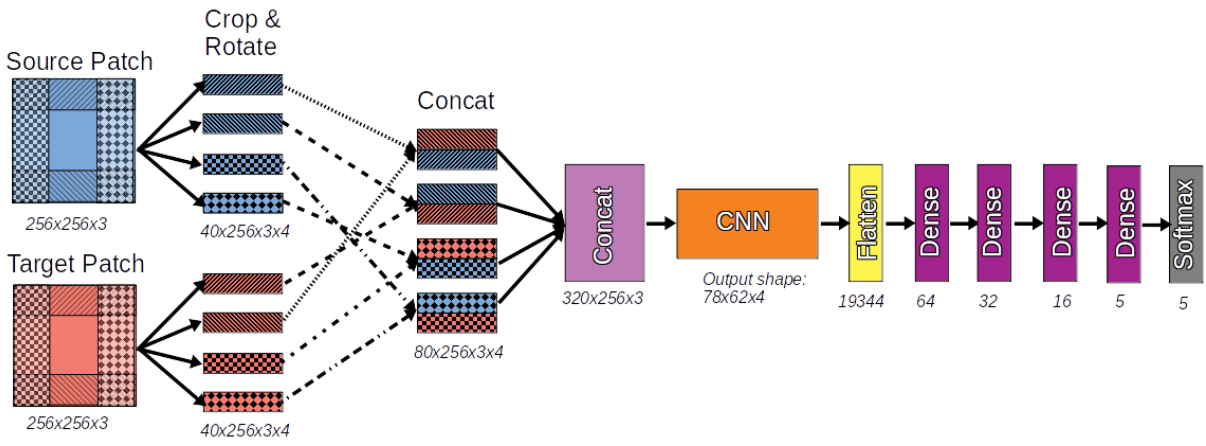


FIGURE 6.8 – Architecture de la partie **Comparaison Deux-à-deux** de notre modèle AssemblyGraphNet

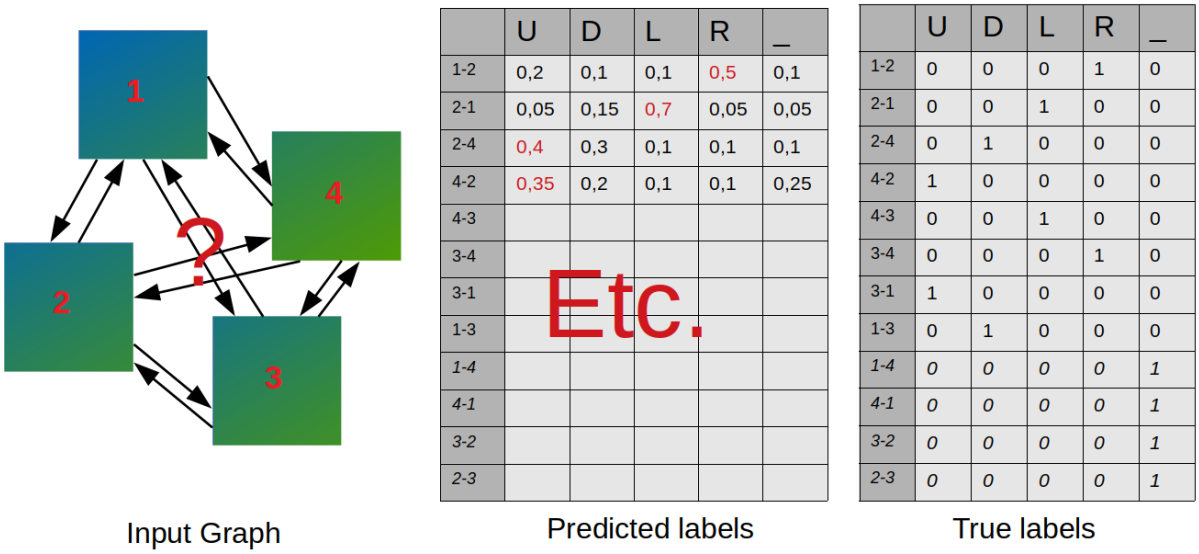
Il est important de noter que nous choisissons ici d'effectuer des concaténations plutôt que des soustractions ou des moyennages, au niveau des zones d'assemblage, de façon à bénéficier de l'asymétrie que crée la concaténation. En effet, notre but est d'évaluer l'assemblage entre les deux patchs dans les deux sens pour chaque direction d'assemblage, afin d'avoir plus d'informations pour la reconstruction globale : deux noeuds reliés par deux arêtes réciproques seront plus susceptibles d'être correctement ré-assemblés, que deux noeuds reliés par une seule arête dirigée.

Layer	Output Shape	Kernel Size	Stride	Output Nodes
BatchNorm	$320 \times 256 \times 3$			
Convolution	$318 \times 254 \times 4$	3	1	4
ReLU	$318 \times 254 \times 4$			
MaxPooling	$159 \times 127 \times 4$	2	2	
BatchNorm	$159 \times 127 \times 4$			
Convolution	$157 \times 125 \times 4$	3	1	4
ReLU	$157 \times 125 \times 4$			
MaxPooling	$78 \times 62 \times 4$	2	2	
BatchNorm	$78 \times 62 \times 4$			

TABLE 6.3 – Architecture et paramètres du réseau de neurones convolutif utilisé dans le modèle **Comparaison Deux-à-deux**

Le modèle **Global** est la partie de notre AssemblyGraphNet qui utilise dans son entièreté la représentation en graphe de l'image complète. Les relations entre les noeuds du graphe sont représentées par une matrice de taille [2 × nombre d'arêtes], dans laquelle la première ligne contient les identifiants de tous les noeuds source, et la deuxième ligne contient les identifiants de tous les noeuds cibles. Nous l'appellerons matrice de connectivité. L'ensemble des tableaux de pixels (valeurs RGB) de chaque *patch* est stocké dans une matrice, que nous appellerons *node features matrix*, et l'ensemble des classes correspondant à chaque arête (représentant la direction d'assemblage entre noeud source et noeud cible) est stocké dans une seconde matrice, que nous appellerons *edge features matrix*. Dans cette matrice, les classes sont stockées sous la forme *one-hot*, c'est-à-dire dans des vecteurs dont la taille est égale au nombre de classes (ici, cinq), et dont l'indice de l'unique valeur égale à 1 représente la classe (voir Figure 6.7).

A partir de la matrice de connectivité, deux matrices de *features* sont calculées : une pour les noeuds sources, et une pour les noeuds cibles. Ainsi, en utilisant ces deux matrices comme entrées, le modèle de **Comparaison Deux-à-deux** est ensuite utilisé simultanément sur l'ensemble des paires noeud source / noeud cible du graphe, en considérant la suite de noeuds comme un *batch* de données. La sortie du modèle **Global** est un nouveau graphe, dans lequel ni le nombre d'arêtes, et ni les attributs des noeuds, ne sont modifiés, mais dans lequel les attributs des arêtes sont remplacés par la classe prédite par notre modèle de **Comparaison Deux-à-deux** (voir Figure 6.9). Les probabilités associées à ces classes sont également stockées, pour un éventuel seuillage *a posteriori*, au moment de la visualisation des résultats (voir Section 5 de cette annexe). Ce graphe est donc toujours un graphe complet, dans lequel de nombreuses arêtes seront identifiées comme appartenant à la classe 0 (pas d'assemblage). Après suppression de ces arêtes, nous obtenons un graphe final de taille considérablement réduite.



	U	D	L	R	_
1-2	0,2	0,1	0,1	0,5	0,1
2-1	0,05	0,15	0,7	0,05	0,05
2-4	0,4	0,3	0,1	0,1	0,1
4-2	0,35	0,2	0,1	0,1	0,25
4-3					
3-4					
3-1					
1-3					
1-4					
4-1					
3-2					
2-3					

	U	D	L	R	_
1-2	0	0	0	1	0
2-1	0	0	1	0	0
2-4	0	1	0	0	0
4-2	1	0	0	0	0
4-3	0	0	1	0	0
3-4	0	0	0	1	0
3-1	1	0	0	0	0
1-3	0	1	0	0	0
1-4	0	0	0	0	1
4-1	0	0	0	0	1
3-2	0	0	0	0	1
2-3	0	0	0	0	1

FIGURE 6.9 – Exemple de prédiction de labels (sous la forme *one-hot*) pour les arêtes du graphe complet représentant une image à reconstituer

Création du jeu de données

Pour cette seconde expérience, nous avons utilisé un autre jeu de données. Celui-ci consiste en un ensemble de 5959 images de papyri, provenant de la collection de la bibliothèque de l'Université du Michigan, qui contient des documents datant de 1000 avant JC, à 1000 après JC [129]. Nous avons supprimé de ce jeu de données 1865 images correspondant à des papyri ne contenant aucun texte, dont l'utilisation pourrait conduire à un mauvais entraînement de notre modèle étant donné leur manque d'information.

Bien que le jeu de données de l'Université du Michigan possède une vérité terrain, nous avons procédé comme dans la section précédente, pour recréer une vérité terrain image par image. Nous avons ainsi créé notre jeu de données avec vérité terrain en divisant les images de papyri en *patchs*. Ainsi, chaque image a d'abord été redimensionnée à 768×1280 pixels, puis divisée en 15 *patchs* sans recouvrement. Nous avons ensuite construit notre graphe dirigé complet en liant tous les *patchs* entre eux en tant que noeuds du graphe.

Étant donné la nature des graphes complets, nos classes sont très déséquilibrées en faveur de la classe 0 (absence d'assemblage), comme le montre la Table 6.4. En effet, une grande majorité des arêtes correspondent en réalité à une absence d'assemblage. De plus, on peut noter que ce déséquilibre ne fait que se creuser en augmentant le nombre de *patchs* par image. Nous avons donc pris en compte ce déséquilibre lors de l'entraînement, en pondérant la fonction de coût utilisée.

Classe	H	B	G	D	_
Effectifs par image	12	12	10	10	166
Effectifs totaux	49128	49128	40940	40940	679604

TABLE 6.4 – Effectif de chaque classe, par image et dans le jeu de données complet

Entraînement et résultats

Notre modèle AssemblyGraphNet a été implémenté en utilisant les bibliothèques pytorch et torch-geometric [128]. Les calculs ont été réalisés en demi-précision (*float16*) de sorte à réduire les ressources de calcul et de mémoire nécessaires. Nous avons utilisé la *cross-entropy* catégorielle comme fonction de coût. Notre jeu de données a été divisé en 3394 images d'entraînement, 500 images de validation, et 200 images de test. Durant l'entraînement, nous avons fait varier aléatoirement l'ordre des paires de noeuds de chaque graphe, afin d'éviter un sur-apprentissage dû l'ordre d'apparition des arêtes.

Comme nous l'avons expliqué précédemment, la classe 0 est sur-représentée dans notre jeu de données. Pour remédier à ce problème, nous avons pondéré la fonction de coût en assignant un poids de 0.1 à classe 0, contre un poids de 0.8 pour les autres classes. Ces poids ont été choisis de façon empirique, après avoir testé un panel de valeurs différentes. Au cours de l'entraînement, nous avons utilisé une nouvelle métrique, appelée *balanced accuracy* (moyenne des valeurs de rappel obtenues pour la prédiction de chaque classe contre l'ensemble des autres), à la place de l'*accuracy* classique (pourcentage de prédictions correctes). Cela nous permet, en plus de la pondération de la fonction de coût, de prendre en compte le déséquilibre des classes [130, 131], et nous donne un score plus représentatif de la capacité de notre modèle à identifier correctement les assemblages deux-à-deux dans toutes les directions. Nous avons également calculé le score F1 pour la prédiction de chaque classe contre l'ensemble des autres.

Après 30 *epochs*, notre modèle atteint un plateau, avec une *balanced accuracy* de 86% sur le jeu de données de validation. Les résultats obtenus avec notre jeu de données sont présentés dans la Table 6.3. la première chose que nous pouvons noter est le fait que la classe 0 (absence d'assemblage) est la classe qui est le plus souvent prédite correctement par notre modèle (colonne F1_0 dans le tableau). Nous pouvons supposer qu'une partie des informations calculées par le modèle de **Comparaison Deux-à-deux** sont liées au gradient de valeurs de pixels au niveau des zones de jonctions possibles. Cela pourrait expliquer que la classe 0 reste la plus simple à identifier, étant donné qu'il est aisé de faire la différence entre un fort gradient (cas où l'assemblage n'est pas possible) et un faible gradient (cas où un assemblage est possible), mais qu'il est difficile de prédire la direction de l'assemblage à partir de ces informations.

	Loss	Acc	F1_1	F1_2	F1_3	F1_4	F1_0
Entraînement	0.46	0.86	0.65	0.63	0.50	0.51	0.81
Validation	0.43	0.86	0.69	0.67	0.54	0.52	0.83
Test	0.42	0.85	0.69	0.68	0.53	0.52	0.81

TABLE 6.5 – Valeurs de la fonction de coût (*loss*), de la *balanced accuracy*, et du score F1 par classe, obtenues après un entraînement de 30 *epochs*

Les résultats que nous obtenons avec notre modèle AssemblyGraphNet sont encourageants, mais nous n'avons pu évaluer que les assemblages deux-à-deux, et non les reconstitutions globales des images. En effet, nous ne pouvons pas évaluer directement ces reconstitutions globales, étant donné que, dans le graphe que nous renvoie notre modèle, chaque noeud est susceptible d'avoir plusieurs voisins dont l'arête incidente correspond à la même direction d'assemblage.

Une solution est de filtrer automatiquement les résultats, en seuillant les probabilités associées à chaque classe, afin de supprimer une partie des arêtes. Cependant il existe un risque de supprimer des arêtes correspondant à des assemblages corrects, mais prédits avec une probabilité insuffisante par noter modèle.

Ainsi, nous avons choisi de proposer une visualisation interactive des graphes d'assemblage, permettant à un utilisateur de décider lui-même de supprimer ou non certaines arêtes, en se basant sur sa propre expertise.

C.5. Interface graphique

Affichage des propositions d'assemblage

Après avoir utilisé notre modèle AssemblyGraphNet pour l'inférence sur un groupe de *patchs*, et éventuellement filtré une partie des arêtes, le graphe résultant est sauvegardé dans un fichier contenant l'ensemble des noeuds avec leurs attributs (tableaux de pixels représentant les *patchs*) et l'ensemble des arêtes dirigées avec leurs attributs (classe prédite et probabilité associée).

Nous avons utilisée la bibliothèque JavaScript Cytoscape.js [132] afin de proposer à l'utilisateur une interface minimaliste, mais fonctionnelle, qui lui permet d'afficher le graphe, et d'interagir avec ses éléments, par exemple pour déplacer des noeuds ou supprimer des arêtes. Le code est disponible à cette adresse : *https ://github.com/CeciliaOstertag/VisuGraph.*

A l'écran, chaque noeud est représenté par l'image du *patch* qui lui correspond. Le graphe est dessiné de façon itérative. Premièrement, les composantes connexes sont identifiées, puis, pour chaque composante connexe, un noeud est choisi arbitrairement comme origine.

Une paire de coordonnées (x,y) est affectée à ce noeud, puis les coordonnées des noeuds suivants sont calculées les unes après les autres en utilisant les relations spatiales entre noeuds (c'est-à-dire les assemblages entre *patchs*), d'après un algorithme de parcours en profondeur.

Cela peut conduire à l'obtention de noeuds qui possèdent les mêmes coordonnées, et ces noeuds sont donc automatiquement mis en valeur à l'écran pour avertir l'utilisateur. Sur l'écran, les différentes composantes connexes, correspondant à des reconstitutions partielles, sont clairement séparées les unes des autres. L'utilisateur peut ensuite choisir entre une représentation compacte, avec les arêtes et les labels invisibles, ou une version étendue montrant toutes les arêtes ainsi que leur label et probabilité de prédiction associée.

Exemples de reconstructions globales

Pour tester le fonctionnement de notre interface graphique, nous avons utilisé des images provenant de notre jeu de données de test avec notre AssemblyGraphNet. Après avoir récupéré les graphes correspondant aux propositions de reconstructions globales, nous avons utilisé notre outil de visualisation pour évaluer la qualité de ces reconstitutions.

La Figure 6.10 montre une proposition de reconstitution globale correcte, obtenue après avoir supprimé les arêtes correspondant à une absence d'assemblage, et filtré les arêtes restantes avec un seuil de 0.8 sur les probabilités de prédiction des classes.

Cette figure illustre bien le cas où nous obtenons une reconstitution parfaite, alors que toutes les arêtes n'ont pas été correctement classifiées (ou classifiées avec une probabilité trop faible) dans le graphe correspondant.

Elle montre également l'intérêt de tester les assemblages dans les deux sens pour chaque direction : l'interface affiche les probabilités associées aux deux arêtes réciproques, et permet ainsi à l'utilisateur de choisir en toute connaissance de cause l'assemblage qu'il considère comme étant le plus juste. A titre d'exemple, on voit sur cette figure que les *patchs* 7 et 4 ont pu être assemblés dans un sens mais pas dans l'autre.

On voit également dans cette figure que l'organisation globale de l'image permet à l'utilisateur d'identifier des assemblages qui n'ont pas été détecté, par exemple entre les *patchs* 13 et 10.

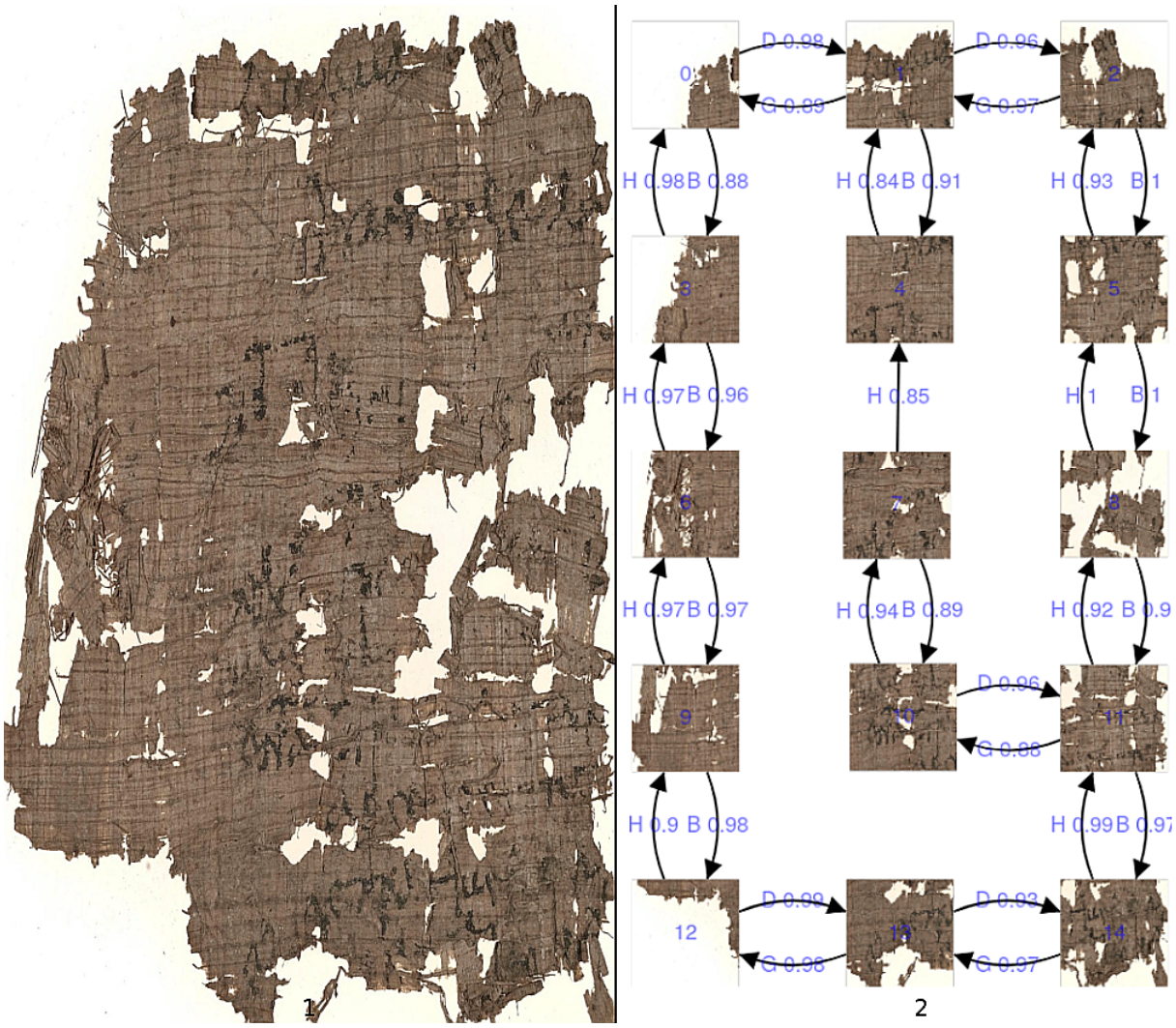


FIGURE 6.10 – Exemple de reconstitution correcte d'une image à partir de 15 *patchs*. 1 : Vérité terrain, 2 : Graphe d'assemblage produit par notre AssemblyGraphNet

Après avoir montré la capacité de notre modèle à fournir des reconstructions correctes d'une image à partir de la collection de *patchs* qui la composent, nous avons cherché à voir si notre modèle était également capable de différencier des *patchs* provenant de deux images différentes. La Figure 6.11 est un exemple de reconstruction obtenue après avoir fourni à notre modèle une liste de 30 *patchs*, provenant de deux images. Ces deux images proviennent de deux papyri distincts, mais dont les différences sont assez subtiles (couleurs très proches, différentes textures, et différents scribes). Comme pour la figure précédente, les arêtes ont été filtrées, pour n'afficher que les plus pertinentes. Nous pouvons voir ici que le graphe d'assemblage produit est composé de quatre composantes connexes, c'est-à-dire de quatre assemblages partiels. De plus, les *patchs* provenant d'images différentes ne sont pas groupés ensembles : les deux premières composantes connexes ne contiennent que des *patchs* provenant de la première image, et les deux dernières ne contiennent que des *patchs* provenant de la seconde image. Ainsi, même si ces deux images ne sont pas

entièrement reconstituées, des assemblages partiels corrects sont tout de même proposés à l'utilisateur.

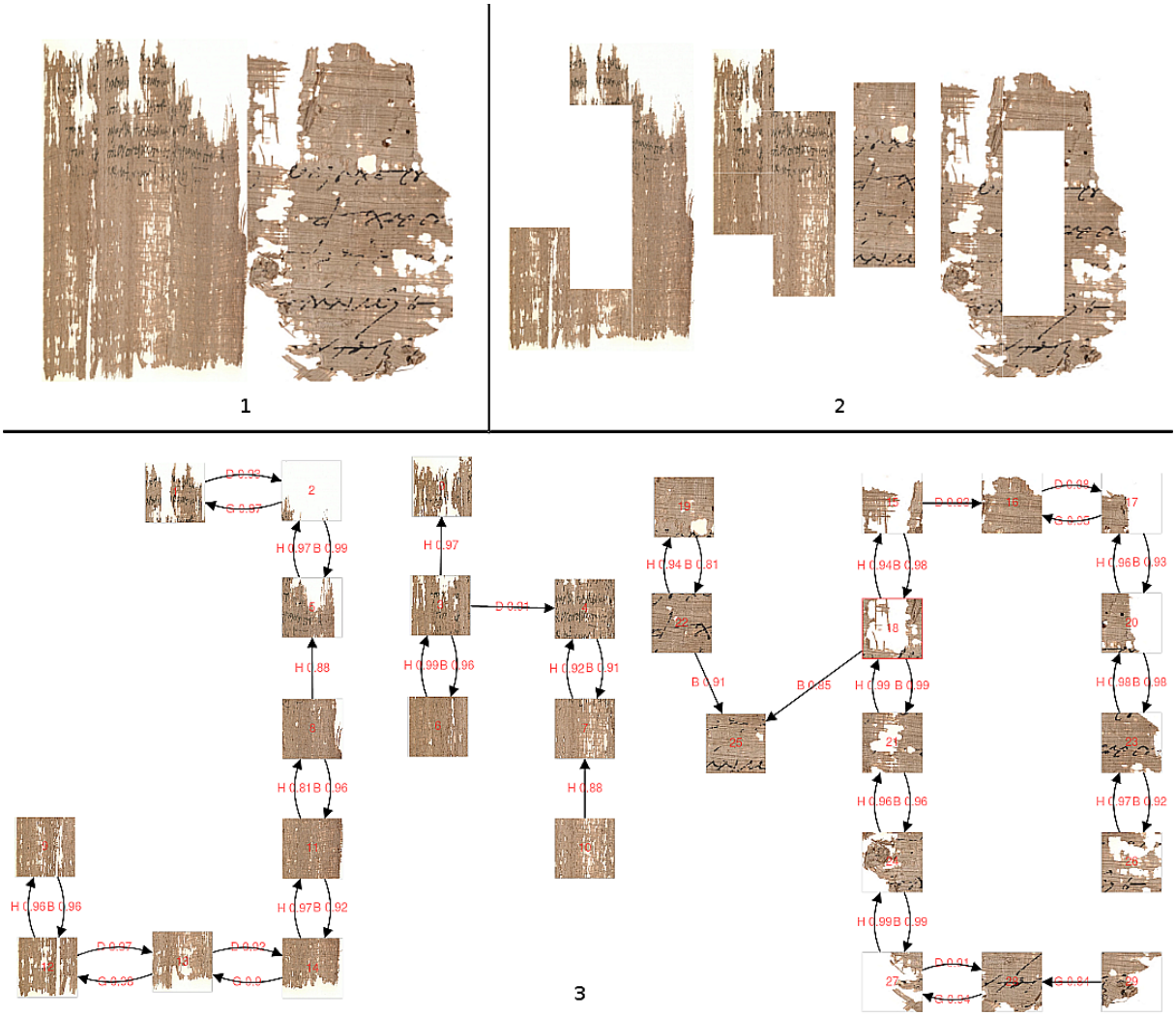


FIGURE 6.11 – Exemple de reconstitution correcte d'une image à partir de 30 *patchs*, provenant de deux images différentes. 1 : Vérité terrain (deux papyri), 2 : Reconstructions partielles, 3 : Graphe d'assemblage produit par notre AssemblyGraphNet

Notre modèle AssemblyGraphNet prédit correctement 85% des relations spatiales entre paires de *patchs* (résultats moyens, obtenus sur les données de test). Ces résultats sont significativement meilleurs que ceux obtenus par Paumard *et. al.* [125], qui ont obtenu 68.8% de fragments correctement placés avec leur jeu de données.

De plus, les auteurs de cet article disent n'obtenir une reconstruction globale correcte que pour 28.8% de leurs images. Dans notre cas, nous n'avons pas de moyens objectifs d'estimer le nombre de reconstitutions globales correctes que notre modèle peut produire, étant donné que nous nous appuyons sur l'intervention de l'utilisateur pour filtrer les graphes

d'assemblage. Cependant, étant donné nos résultats pour la classification des arêtes, nous pouvons supposer qu'un nombre plus grand de reconstructions globales peut être obtenu en utilisant notre modèle.

Enfin, dans une situation où les *patchs* fournis au modèle proviennent d'images différentes, le modèle proposé par Paumard *et. al.* ne sera pas adapté, étant donné que, contrairement à nous, les auteurs n'ont pas prévu de classe pour modéliser l'absence d'assemblage.

	Paumard et. al	**AssemblyGraphNet**
Prédiction des relations spatiales entre *patchs* correctes	68.8%	86%
Reconstitutions globales correctes	28.8%	Basées sur l'intervention de l'utilisateur
Modélisation de l'absence d'assemblage entre deux *patchs*	Non	Oui (classe 0)
Reconstructions à partir de *patchs* provenant de plusieurs images	Non	Oui

TABLE 6.6 – Comparaison entre notre modèle et le modèle proposé par Paumard *et. al.* [125]

C.6. Discussion et Conclusion

Dans ces travaux, nous avons utilisé l'architecture de type réseaux siamois pour un problème de reconstitution de documents anciens. Nous avons commencé par présenter un modèle, OstraNet, prenant en entrée une paire de *patchs* et capable de prédire l'existence et la direction de l'assemblage entre ces deux *patchs*. Ce modèle obtient une *accuracy* de 96% après avoir été entraîné sur un jeu de données de 7000 *patchs* dont 1000 ont servi à la validation. Nous avons également proposé un procédé itératif pour reconstituer une image entière à partir d'une suite de *patchs*, basé sur la création de graphes représentant les reconstructions possibles. Cependant, nous avons montré que ce procédé est lourd en terme de temps de calcul et de quantité de résultats générés, ce qui nous a conduit à changer de stratégie.

Nous avons ensuite étudié une approche complémentaire, basée cette fois sur les réseaux de neurones adaptés aux graphes, et permettant d'identifier les alignements deux-à-deux ainsi que l'organisation globale de l'image complète de façon simultanée. Ce modèle, appelé AssemblyGraphNet, produit directement en sortie un graphe correspondant à une reconstitution globale, dans lequel chaque *patch* peut avoir plusieurs *patchs* adjacents possibles.

De plus, si l'on considère uniquement l'étape d'inférence, le temps d'exécution de notre modèle est faible : sur un GPU GeForce RTX 2080 Ti, il faut en moyenne 0.8 ms pour obtenir un graphe d'assemblage à partir d'un groupe de 15 *patchs*.

Nous avons également montré que notre AssemblyGraphNet est capable de distinguer des *patchs* provenant d'images différentes, en utilisant des informations comme la couleur, la texture, et les inscriptions présentes sur le document. Ce résultat est particulièrement intéressant, car cela veut dire que pour une application concrète il n'y aurait pas besoin de trier préalablement les documents.

Dans l'état actuel de ces travaux, notre modèle n'utilise pas beaucoup de propriétés inhérentes aux graphes, à l'exception de la connectivité, qui permet de traiter toutes les comparaisons deux-à-deux en une seule fois. Cette approche basée sur les GraphNN a été proposée avec l'intention de l'utiliser pour reconstruire les images entières en plusieurs étapes. Par la suite, au lieu d'une simple passe comme nous l'avons présenté ici, il serait possible d'assembler progressivement des groupes de *patchs* de plus en plus larges.

Nous fournissons sur GitHub l'implémentation de nos deux modèles, ainsi que les modèles entraînés, de même que le code source pour notre interface de visualisation interactive des résultats. Grâce à celle-ci, un utilisateur possédant une certaine connaissance du sujet peut affiner et corriger les graphes d'assemblage. Pour cela, il pourra se baser aussi bien sur les probabilités associées à chaque label d'arête, que sur la réciprocité des arêtes entre deux noeuds, ou sur les noeuds voisins. Nous pouvons donc espérer que le *pipeline* que nous proposons paraîtra intuitif aux chercheurs du domaine de l'héritage culturel.

Analyse des pathologies neuro-dégénératives par apprentissage profond

Résumé :
Le suivi et l'établissement de pronostiques sur l'état cognitif des personnes affectées par une maladie neurologique sont cruciaux, car ils permettent de fournir un traitement approprié à chaque patient, et cela le plus tôt possible. Ces patients sont donc suivis régulièrement pendant plusieurs années, dans le cadre d'études longitudinales. A chaque visite médicale, une grande quantité de données est acquise : présence de facteurs de risque associés à la maladie, imagerie médicale (IRM ou PET-scan), résultats de tests cognitifs, prélèvements de molécules identifiées comme bio-marqueurs de la maladie, *etc.* Ces différentes modalités apportent des informations sur la progression de la maladie, certaines complémentaires et d'autres redondantes.
De nombreux modèles d'apprentissage profond ont été appliqués avec succès aux données bio-médicales, notamment pour des problématiques de segmentation d'organes ou de diagnostique de maladies. Ces travaux de thèse s'intéressent à la conception d'un modèle de type "réseau de neurones profond" pour la prédiction du déclin cognitif de patients à l'aide de données multimodales.
Ainsi, nous proposons une architecture composée de sous-modules adaptés à chaque modalité : réseau convolutif 3D pour les IRM de cerveau, et couches entièrement connectées pour les données cliniques quantitatives et qualitatives. Pour évaluer l'évolution du patient, ce modèle prend en entrée les données de deux visites médicales quelconques. Ces deux visites sont comparées grâce à une architecture siamoise. Après avoir entraîné et validé ce modèle en utilisant comme cas d'application la maladie d'Alzheimer, nous nous intéressons au transfert de connaissance avec d'autres maladies neuro-dégénératives, et nous utilisons avec succès le transfert d'apprentissage pour appliquer notre modèle dans le cas de la maladie de Parkinson.
Enfin, nous discutons des choix que nous avons pris pour la prise en compte de l'aspect temporel du problème, aussi bien lors de la création de la vérité terrain en fonction de l'évolution au long terme d'un score cognitif, que pour le choix d'utiliser des paires de visites au lieu de plus longues séquences.

Mots clés :
apprentissage profond, multimodal, maladies neuro-dégénératives, réseaux siamois

Analysis of neuro-degenerative disorders using deep learning

Summary:
Monitoring and predicting the cognitive state of a subject affected by a neuro-degenerative disorder is crucial to provide appropriate treatment as soon as possible. Thus, these patients are followed for several years, as part of longitudinal medical studies. During each visit, a large quantity of data is acquired: risk factors linked to the pathology, medical imagery (MRI or PET scans for example), cognitive tests results, sampling of molecules that have been identified as bio-markers, *etc.* These various modalities give information about the disease's progression, some of them are complementary and others can be redundant.
Several deep leraning models have been applied to bio-medical data, notably for organ segmentation or pathology diagnosis. This PhD is focused on the conception of a deep neural network model for cognitive decline pediction, using multimodal data, here both structural brain MRI images and clinical data.
In this thesis we propose an architecture made of sub-modules tailored to each modality: 3D convolutional network for the brain MRI, and fully connected layers for the quantitative and qualitative clinical data. To predict the patient's evolution, this model takes as input data from two medical visits for each patient. These visits are compared using a siamese architecture. After training and validating this model with Alzheimer's disease as our use case, we look into knowledge transfer to other neuro-degenerative pathologies, and we use transfer learning to adapt our model to Parkinson's disease.
Finally, we discuss the choices we made to take into account the temporal aspect of our problem, both during the ground truth creation using the long-term evolution of a cognitive score, and for the choice of using pairs of visits as input instead of longer sequences.

Keywords :
deep learning, multimodality, neuro-degenerative disorders, siamese networks

L3i

Laboratoire L3i
Faculté des Sciences et Technologies
Bâtiment Pascal
Avenue Michel Crépeau

17042 LA ROCHELLE