# Approche semi-supervisée pour l'analyse automatique de la moelle épinière sur imagerie par résonance magnétique

par

Nicolas M. GIGNAC

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLEÀ L'OBTENTION DE
LA MAÎTRISE AVEC MÉMOIRE EN GÉNIE LOGICIEL
M. Sc. A.

MONTRÉAL, LE 27 JUIN 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

**PRÉSENTATION DU JURY**

CE OU MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Luc Duong, directeur de mémoire
Génie logiciel et TI à l'École de technologie supérieure

M. Jean-Marc Mac-Thiong, codirecteur de mémoire
Centre de recherche de l'Hôpital du Sacré-Cœur de Montréal

M. Simon Drouin, président du jury
Génie logiciel et TI à l'École de technologie supérieure

Mme Sylvie Ratté, membre du jury
Génie logiciel et TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 10 JUIN 2022

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

## REMERCIEMENTS

J’aimerais tout d’abord remercier mon directeur de mémoire, Luc Duong. Merci d’avoir partagé votre expertise sur de nombreux sujets et de m’avoir guidé judicieusement tout au long de ce processus. Je me sens sincèrement privilégié de vous avoir eu comme directeur de mémoire.

Merci à mon codirecteur de mémoire, Jean-Marc Mac Thiong, de m’avoir confié les données nécessaires à ce projet et de m’avoir guidé à travers le volet médical avec lequel j’étais moins confortable. Vos idées et conseils sont très appréciés.

Merci à Simon Drouin d’avoir accepté de présider le jury. Merci aussi à Sylvie Ratté d’avoir accepté d’être membre du jury.

Finalement, merci à ma conjointe de m’avoir supporté dans cette démarche, qui a fini par être un peu plus longue que prévue, mais qui se termine très bien.

**Approche semi-supervisée pour l'analyse automatique de la moelle épinière sur imagerie par résonnance magnétique**

Nicolas M. GIGNAC

## RÉSUMÉ

Cette étude utilise des réseaux de neurones convolutifs 2D afin d'automatiser la segmentation de la moelle épinière à partir d'IRM sagittaux pondérés en T2. La base de données est constituée de 106 IRM provenant de 94 patients ayant subi des lésions traumatiques à la moelle épinière. Deux méthodes complètes intégrant le prétraitement des données ainsi que la segmentation automatique ont été développées.

Une de ces méthodes a comme objectif de produire la meilleure segmentation possible de la moelle épinière d'un point de vue clinique. Celle-ci emploie une approche novatrice où les IRM sont analysées en cascade sous le plan axial et sagittal par deux réseaux différents. Les résultats sont comparés avec ceux de Deepseg 2D du Spinal Cord Toolbox, qui représente l'état de l'art. Notre méthode arrive à des résultats significativement supérieurs à Deepseg 2D, avec un coefficient Dice moyen de 0.95 contre 0.88 pour Deepseg 2D ($P < 0.001$).

L'autre approche a comme objectif le développement et l'évaluation d'une méthode d'apprentissage semi-supervisée. Dans cette approche, deux réseaux sont entrainés en parallèle où chaque réseau utilise une version binarisée des segmentations produites par l'autre réseau afin d'améliorer son propre entrainement. Pour évaluer cette approche, quatre paires de réseaux sont entrainés avec 25%, 50%, 75% et 100% des données annotées respectivement, puis les résultats sont comparés avec ceux de réseaux utilisant seulement l'apprentissage supervisé. Les résultats montrent une amélioration considérable pour les réseaux utilisant l'apprentissage semi-supervisé, amélioration qui est certainement plus marquée lorsque moins de données sont annotées, mais qui demeure néanmoins présente avec 100% des données annotées.

Deux approches ont donc été développées avec succès pour segmenter la moelle épinière, l'une présentant une technique supervisée qui arrive à des résultats de segmentation dépassant l'état de l'art, et l'autre présentant une technique semi-supervisée qui performe bien avec des quantités de données annotées variées.

**Mots-clés** : Apprentissage profond, apprentissage semi-supervisé, segmentation, moelle épinière, réseau de neurones convolutif

**Approche semi-supervisée pour l'analyse automatique de la moelle épinière sur imagerie par résonnance magnétique**

Nicolas M. GIGNAC

## ABSTRACT

This study uses 2D convolutional neural networks to automate spinal cord segmentation from T2-weighted MRIs. The database consists of 106 sagittal MRI scans from 94 patients with traumatic spinal cord injuries. Two complete methods integrating data preprocessing and automatic segmentation have been developed.

One of these methods aims for the best possible segmentation of the spinal cord from a clinical point of view. This method uses an innovative approach where the MRIs are analyzed in series under the axial and sagittal plane by two different networks. The results are compared with those of Deepseg 2D from the Spinal Cord Toolbox, which is a method representing the state of the art. Our method achieves significantly better results than Deepseg 2D, with an average Dice coefficient of 0.95 against 0.88 for Deepseg 2D ($P < 0.001$).

The other approach aims to develop and evaluate a semi-supervised learning method. In this approach, two networks are trained in parallel, where each network uses a binarized version of the segmentations produced by the other network to improve its own training. To evaluate this approach, four pairs of networks are trained with 25%, 50%, 75% and 100% of the annotated data respectively, then the results are compared with those of networks using only supervised learning. The results show a considerable improvement over the supervised baseline, an improvement that is certainly more marked when a lesser portion of the data is annotated, but which nevertheless remains with 100% of the annotated data.

Two approaches have therefore been successfully developed to segment the human spinal cord, one presenting a supervised technique which achieves segmentation results exceeding the state of the art, and the other presenting a semi-supervised technique which performs well with variable amounts of data.

**Keywords** : Deep learning, semi-supervised learning, segmentation, spinal cord, convolutional neural networks (CNNs)

**TABLE DES MATIÈRES**

Page

INTRODUCTION .....1

CHAPITRE 1 REVUE DE LA LITTÉRATURE .....5
1.1 La moelle épinière.....5
1.1.1 Lésions médullaires ..... 6
1.2 Imagerie par résonance magnétique.....8
1.2.1 Séquences..... 8
1.2.2 Autres paramètres ..... 9
1.3 Pronostic des lésions médullaires .....10
1.4 Apprentissage profond : contexte actuel et applications.....14
1.5 Segmentation de la moelle épinière .....14
1.5.1 Défis..... 14
1.5.2 Approches ..... 16
1.6 Spinal Cord Toolbox.....22

CHAPITRE 2 MÉTHODOLOGIE.....25
2.1 Mise en contexte .....25
2.1.1 Description et prétraitement des données ..... 25
2.1.2 Architecture du réseau convolutif..... 29
2.1.3 Métriques pour la validation des résultats ..... 31
2.1.3.1 Mesure de similitude basée sur la ligne centrale ..... 31
2.1.4 Méthode de validation croisée ..... 34
2.2 Approche en deux étapes .....35
2.2.1 Étape 1: segmentation sur les coupes axiales ..... 37
2.2.2 Étape 2 : segmentation sur les coupes sagittales..... 38
2.2.3 Méthode d'analyse statistique des résultats ..... 40
2.2.3.1 Test-t ..... 40
2.2.3.2 Limitations ..... 42
2.2.3.3 Implémentation sur les résultats..... 43
2.2.3.4 Analyse de la distribution des résultats..... 44
2.3 Approche semi-supervisée.....45
2.3.1 Apprentissage pseudo-supervisé en croisé..... 45
2.3.2 Prétraitement des données..... 48
2.3.3 Méthode de validation croisée ..... 48
2.3.4 Pertes non-supervisées..... 49
2.3.5 Résumé des paramètres..... 50
2.3.6 Procédé d'entrainement ..... 50

CHAPITRE 3 RÉSULTATS ET DISCUSSION .....53
3.1 Approche en deux étapes .....53
3.1.1 Étape 1 : segmentation sur les coupes axiales ..... 53
3.1.2 Étape 2 : segmentation sur les coupes sagittales..... 62
3.1.2.1 Comparaison de complexité..... 74

3.2 Approche semi-supervisée ..... 76

CONCLUSION…. ..... 85

ANNEXE I Prétraitement des données ..... 87

ANNEXE II Apprentissage profond pour l'analyse d'image automatique ..... 97

ANNEXE III Métriques pour la validation des approches ..... 111

ANNEXE IV Comparaison de Deepseg 2D et Deepseg 3D ..... 115

ANNEXE V Analyse de l'architecture U-Net ..... 117

BIBLIOGRAPHIE ..... 126

## LISTE DES TABLEAUX

Page


Tableau 2.1 Résumé des hyperparamètres du réseau axial....37

Tableau 2.2 Résumé des hyperparamètres du réseau sagittal....39

Tableau 2.3 Résume des hyperparamètres....50

Tableau 3.1 Pertes et coefficients de validation Dice....55

Tableau 3.2 Résultats Dice sur les 20 IRM de test pour les dix itérations de validation croisée....56

Tableau 3.3 Résultats Jaccard sur les 20 IRM de test pour les dix itérations de validation croisée....57

Tableau 3.4 Taux de vrais positifs sur les 20 IRM de test pour les dix itérations de validation croisée....58

Tableau 3.5 Résultats pour la métrique de la ligne centrale pour les dix itérations de validation croisée....59

Tableau 3.6 Distances Hausdorff moyennes pour les dix itérations de validation croisée....60

Tableau 3.7 Tableau récapitulatif des résultats pour l'étape axiale....61

Tableau 3.8 Pertes et coefficients de validation Dice....64

Tableau 3.9 Moyenne des coefficients Dice sur les IRM de test pour les dix itérations de validation croisée....65

Tableau 3.10 Moyennes des coefficients Jaccard sur les IRM de test pour les dix itérations de validation croisée....66

Tableau 3.11 Moyennes des taux de vrais positifs sur les IRM de test pour les dix itérations de validation croisée....67

Tableau 3.12 Moyennes des résultats de la métrique Ligne centrale sur les IRM de test pour les dix itérations de validation croisée....68

Tableau 3.13 Moyennes des distances Hausdorff sur les IRM de test pour les dix itérations de validation croisée....70

Tableau 3.14 Récapitulatif des résultats pour l'étape sagittale....71

Tableau 3.15 Moyennes de la méthode abrégée et la méthode .....76

Tableau 3.16 Époque avec la perte de validation moyenne.....78

Tableau 3.17 Moyennes des coefficients Dice sur les IRM de test pour les dix itérations de validation croisée .....78

Tableau 3.18 Moyennes des coefficients Jaccard sur les IRM de test pour les dix itérations de validation croisée.....79

Tableau 3.19 Moyennes des taux de vrai positif sur les IRM de test pour les dix itérations de validation croisée.....79

Tableau 3.20 Moyennes des lignes centrales sur les IRM de test pour les dix itérations de validation croisée .....80

Tableau 3.21 Moyennes des distances Hausdorff sur les IRM de test pour les dix itérations de validation croisée.....80

Tableau 3.22 Tableau récapitulatif des résultats. Les valeurs représentent les moyennes des dix itérations de validation croisée sur les IRM de test.....81

## LISTE DES FIGURES

Page


Figure 0.1 IRM sagittale d'un patient atteint d'une lésion ..... 2

Figure 1.1 Processus de segmentation de la moelle épinière et des lésions Tirée de Gros et al., 2019 ..... 17

Figure 1.2 Réseau Deepseg 2D utilisé pour segmenter les IRM pondérées en T2 ..... 24

Figure 2.1 Vue de la segmentation manuelle d'une coupe sagittale ..... 26

Figure 2.2 Étapes de prétraitements pour les réseaux axiaux et sagittaux ..... 27

Figure 2.3 Illustration de transformations utilisées sur les tranches sagittales ..... 28

Figure 2.4 Illustration des transformées utilisées sur les tranches axiales ..... 29

Figure 2.5 Schéma du réseau axial utilisé ..... 30

Figure 2.6 Centre de masse *(x, y)* d'une tranche *z* ..... 32

Figure 2.7 Visualisation de la ligne centrale d'une segmentation (noir) ..... 34

Figure 2.8 Schéma des dix sous-ensembles de validation (jaune). Les IRM ..... 35

Figure 2.9 Schéma global de l'approche en deux étapes ..... 36

Figure 2.10 Créations des échantillons ..... 41

Figure 2.11 Schéma du système de pertes semi-supervisées ..... 47

Figure 2.12 Visualisation des données pour deux des dix itérations de validation croisée. Le jaune représente les données de validation, le bleu foncé les données annotées et le bleu pâle les données non-annotées. La base de données est mélangée aléatoirement précédant ces séparations. ..... 49

Figure 3.1 Courbes de pertes de validation pour chaque sous-ensemble (gauche) et moyenne des pertes de validation et d'entrainement pour tous les sous-ensembles (droite). La moyenne des pertes de validation atteint sa valeur minimale à l'époque 49 ..... 53

Figure 3.2 Estimations par noyau et valeurs-p des tests Agostino-Pearson pour les cinq métriques pour le réseau Deepseg ..... 54

Figure 3.3 Estimations par noyau des résultats pour les dix itérations croisées. Chaque courbe représente un des dix réseaux et est constituée des 20 résultats obtenus sur les IRM de test. On peut observer que les distributions du graphique Centroïde (ligne centrale) sont biaisées vers la droite, ce qui explique pourquoi plusieurs distributions échouent le test de normalité .....60

Figure 3.4 Courbes de pertes de validation pour chaque sous-ensemble (gauche) et moyenne des pertes de validation et d'entrainement pour tous les sous-ensembles (droite). La moyenne des pertes de validation atteint sa valeur minimale à l'époque 118.....63

Figure 3.5 Estimations par noyau des résultats pour les cinq métriques. Chaque courbe représente un des dix réseaux de validation croisée .....69

Figure 3.6 Comparaisons visuelles de notre approche avec Deepseg 2D .....73

Figure 3.7 Exemple d'une coupe sagittale en bordure de la moelle épinière.....74

Figure 3.8 Pertes de validation moyennes pour l'approche avec 25% des données annotées. Puisque chacune des dix itérations est constituée de deux réseaux, ces courbes moyennes sont calculées à partir des données de 20 réseaux. .....77

Figure 3.9 Moyennes des cinq métriques pour les quatre approches.....82

Figure 3.10 Segmentation de la même coupe sagittale avec différentes proportions de données annotées .....83

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

| | |
|---|---|
| AIS | *ASIA Impairment Scale* |
| AMS | *ASIA Motor Scale* |
| ASIA | *American Spinal Injury Association* |
| CPU | *Central processing unit* |
| CNN | *Convolutive neural network* |
| IRM | Imagerie par résonance magnétique |
| ISNCSCI | *International Standards for Neurological Classification of Spinal Cord Injury* |
| IoU | *Intersection over union* |
| FLOPS | *Floating point operations per second* |
| GPU | *Graphics processing unit* |
| LASSO | *Least absolute shrinkage and selection operator* |
| NIfTI | *Neuroimaging Informatics Technology Initiative* |
| RPI | *Right posterior inferior* |
| SCT | *Spinal Cord Toolbox* |
| TPR | *True positive rate* |
| VRAM | *Video random access memory* |

# INTRODUCTION

La moelle épinière est une structure anatomique hautement complexe faisant partie du système nerveux central. Sa fonction principale est d'assurer la conduction des signaux moteurs du cerveau vers le système nerveux périphérique et la conduction des signaux sensoriels du système nerveux périphérique vers le cerveau. La dégénération causée à la moelle épinière par diverses maladies et les dommages induits par des accidents traumatiques peuvent donc altérer drastiquement le fonctionnement du corps humain. Or, l'imagerie par résonnance magnétique est l'outil moderne de choix pour évaluer les lésions à la moelle épinière. La segmentation de la moelle épinière par IRM est le plus souvent utilisée pour mesurer sa surface transversale à divers niveaux, ce qui permet de réaliser des analyses quantitatives pour caractériser les lésions (De Leener et al., 2016). La segmentation automatique de la moelle épinière sert aussi à construire des atlas standardisés comme dans Spinal Cord Toolbox et facilite les études longitudinales où la segmentation manuelle est trop longue à accomplir.

Plusieurs approches existent pour la segmentation automatique de la moelle épinière à partir d'IRM. Certains réseaux, tel que Propseg (De Leener et al., 2014), utilisent des approches algorithmiques sans réseaux de neurones, tandis que les approches plus récentes telles que Deepseg 2D (Spinal Cord Toolbox), Deepseg 3D (Gros et al., 2019) ou BASICseg (McCoy et al., 2019) utilisent des méthodes d'apprentissage profond basées sur des architectures de type U-Net. Les approches Deepseg sont capables de segmenter des moelles épinières saines ou comportant des lésions non-traumatiques avec une précision élevée (coefficient Dice > 95%). Cependant, la qualité des segmentations chez les patients atteints de lésions médullaires (lésions à la moelle épinière) traumatiques est considérablement inférieure (coefficient Dice < 90%). Ceci est dû au fait que les caractéristiques normales des structures entourant la moelle épinière dans les zones blessées sont perdues ou déformées, comme on peut le voir sur la Figure 0.1. Les approches Deepseg ont également de la difficulté à segmenter la moelle épinière de certains patients plus âgés, puisque les démarcations l'entourant peuvent être moins claires.

Figure 0.1 IRM sagittale d'un patient atteint d'une lésion traumatique à la moelle épinière. Tirée de Andrew Dixon, 2019

La première contribution de ce mémoire est de développer une approche automatique pour la segmentation des blessures médullaires. Ainsi, nous avons développé une approche robuste capable de segmenter non-seulement les moelles épinières saines, mais aussi celles atteintes de blessures traumatiques ou dégradées par l'âge. À cette fin, une méthode automatique incorporant la standardisation des données IRM ainsi que l'entrainement de réseaux convolutifs a été développée. Cette méthode utilise des IRM sagittaux de blessés médullaires pondérés en T2 afin d'entrainer deux réseaux convolutifs placés en cascade. Les IRM sont d'abord acheminées au premier réseau, qui les analyse dans le plan axial et produit une première segmentation. Ensuite, cette segmentation ainsi que l'IRM originale sont acheminées au deuxième réseau, qui les analyse dans le plan sagittal et produit la segmentation finale. Les réseaux sont évalués à l'aide de plusieurs métriques, dont une spécialement conçue dans le

cours de cette étude pour évaluer les lignes centrales des segmentations. Les résultats sont comparés à ceux du réseau Deepseg 2D, qui représente l'état de l'art.

La deuxième contribution de ce mémoire est de développer une technique de segmentation de la moelle épinière incorporant l'apprentissage semi-supervisé. L'apprentissage semi-supervisé vise à réduire la quantité de données annotées requises pour entrainer des réseaux. Ceci est important car l'annotation de données en général, et particulièrement dans le domaine médical, est un processus qui est long et demande une connaissance spécifique du domaine visé. La variété des données annotées est de prime importance lors de l'entrainement supervisé d'un réseau, car sa capacité à inférer sur des nouvelles données par la suite est directement proportionnelle à celle-ci. Il est donc utile de développer des techniques visant à réduire la dépendance du processus d'apprentissage sur les données annotées afin de maximiser l'utilité de toutes les données disponibles, ce qui est l'objectif de l'apprentissage semi-supervisé.

L'approche semi-supervisée développée dans ce mémoire utilise deux réseaux convolutifs entrainés en parallèle. Ces réseaux sont entrainés avec un mélange de pertes supervisées et non-supervisées. Pour les pertes non-supervisées, chacun des deux réseaux utilise une version binarisée des segmentations de l'autre réseau à la place de segmentations manuelles. Plusieurs versions des réseaux sont entrainées avec entre 25% et 100% des données d'annotées. Les résultats sont évalués en comparant les réseaux semi-supervisés à des réseaux supervisés entrainés avec les mêmes quantités de données annotées.

L'organisation de ce mémoire est divisée en trois chapitres, dont le premier constitue la revue de littérature. Celui-ci débute avec une introduction à l'anatomie de la moelle épinière et des lésions médullaires. Ensuite, un survol des différents paramètres associés aux IRM et des facteurs influant dans les pronostics des lésions médullaires est présenté. Puis, un bref aperçu du contexte actuel entourant l'apprentissage profond est présenté et l'architecture de type U-Net, qui est utilisée dans ce mémoire, est vue en détail. Finalement, les approches existantes visant la segmentation de la moelle épinière sont présentées.

Le deuxième chapitre présente les méthodologies utilisées lors de l'approche en deux étapes et de l'approche semi-supervisée. Les éléments communs aux deux approches sont d'abord présentés, suivi des éléments spécifiques à l'approche en deux étapes et puis des éléments spécifiques à l'approche semi-supervisée. Le troisième chapitre présente les résultats et discussions pour les deux approches. La conclusion revient d'abord sur les résultats plus notables, rappelant leurs forces et limites, puis discute des améliorations et travaux futurs possibles.

# CHAPITRE 1

# REVUE DE LA LITTÉRATURE

## 1.1 La moelle épinière

La moelle épinière ou moelle spinale est la structure anatomique la plus importante pour la communication entre le cerveau et le reste du corps. Elle est de forme cylindrique avec une coupe transversale ovale et s'étend de la moelle allongée dans le cerveau jusqu'à la première ou deuxième vertèbre lombaire. Elle est caractérisée par deux sillons extérieurs ou sulci, soit un, plus volumineux, sur son côté ventral (vers le corps), et l'autre sur son côté dorsal (vers l'extérieur). Elle mesure environ 45 cm chez l'homme et 42 cm chez la femme. À l'exemple des vertèbres, on la divise en cinq régions qui ont les mêmes noms, soit la région cervicale, thoracique, lombaire, sacrée et coccygienne. Il est à remarquer, cependant, que ces régions ne correspondent pas nécessairement exactement aux régions vertébrales dans l'espace.

La moelle épinière est composée de matière nerveuse blanche et grise. La matière blanche se trouve en périphérie et contient les axones myélinisés des neurones, tandis que la matière grise se trouve au centre et contient le corps des neurones. La matière grise peut être subdivisée en trois colonnes qui lui donnent sa forme de papillon (3 colonnes sur chaque aile). Elle entoure le canal central, une cavité remplie de liquide cérébrospinal qui se propage sur la longueur de la moelle épinière. Ce canal est une extension du quatrième ventricule du cerveau.

La matière blanche en périphérie peut être subdivisée en plusieurs sous-voies, qui sont composées d'agglomérations d'axones assurant la communication entre des parties spécifiques du corps et du cerveau (Ahuja et al., 2017). Ces agglomérations nerveuses forment une série de racines nerveuses, soit 31 paires, qui s'étendent de chaque côté de la moelle épinière. Chaque racine de chaque paire de racines nerveuses comporte une branche ventrale, qui est responsable pour la communication moteur, et une branche dorsale, qui est responsable pour la communication sensorielle.

La moelle épinière est caractérisée par deux élargissements. Le premier est au niveau cervical, entre les vertèbres C5 et T1, et correspond à la section de la moelle épinière responsable pour la communication avec les bras et le tronc du corps. Le deuxième élargissement est au niveau lombaire, entre L1 et S3, et correspond à la section de la moelle responsable pour la communication avec les jambes.

Un canal formé par les vertèbres entoure et protège la moelle épinière sur toute sa longueur. Ce canal, qu'on nomme cavité spinale, est rempli de liquide cérébrospinal. Ce liquide a plusieurs fonctions, dont le support physique de la moelle, l'absorption de chocs, la nutrition, l'immunité, ainsi que plusieurs fonctions homéostatiques telles que la régularisation de la température et l'équilibre biochimique.

Trois membranes protègent aussi la moelle épinière. On nomme ce groupe de membranes les méninges. La membrane extérieure, nommée dure-mère, forme une enveloppe solide protectrice. La membrane suivante, nommée arachnoïde, contient des espaces ouverts comme une toile d'araignée. Ces deux membranes sont séparées par l'espace épidural, qui contient du tissu adipeux et des vaisseaux sanguins. La dernière couche est la plus délicate et se nomme pie-mère. Elle est séparée de la couche arachnoïde par l'espace subarachnoïdien, qui contient du liquide cérébrospinal.

### 1.1.1 Lésions médullaires

Une blessure à la moelle épinière, ou lésion médullaire, peut être définie comme un dommage à la moelle épinière qui porte atteinte à ses fonctions de façon temporaire ou permanente (Ahuja et al., 2017). Il existe deux catégories de lésions médullaires, soit les lésions traumatiques et non-traumatiques. Les lésions traumatiques surviennent à la suite d'un impact externe, habituellement dans le contexte d'un accident comme une chute ou un accident d'auto, tandis que les lésions non-traumatiques sont causées par des maladies comme la discopathie dégénérative ou la sclérose en plaques.

Suivant une lésion traumatique, une cascade de réactions biologiques se produit qui peut venir grandement aggraver la blessure initiale. Cette cascade produit la mort de plusieurs neurones et cellules gliales additionnelles, ainsi que de l'ischémie et de l'inflammation. Ces effets ont pour conséquence des changements dans l'organisation de la moelle épinière menant à des cicatrices gliales et des cavités cystiques, qui contribuent à garder le potentiel de guérison suite à une lésion traumatique assez faible et peuvent entrainer des dommages en excès des dommages initiaux (Ahuja et al., 2017).

Les lésions à la moelle épinière sont décrites par rapport aux différentes régions d'où sortent les racines nerveuses de la moelle épinière (région cervicale, etc.), alors que les fractures de la colonne vertébrale sont décrites selon les régions vertébrales qui portent les mêmes noms. Or, ces deux ensembles de régions, bien qu'ils portent les mêmes noms et sont placés dans le même ordre, ne coïncident pas nécessairement dans l'espace. En effet, l'écart entre les deux devient de plus en plus apparent à partir du centre de la région thoracique en descendant, où une fracture T8 peut causer une lésion au niveau T12 de la moelle épinière (Ahuja et al., 2017).

Une lésion à la moelle épinière peut causer une perte partielle ou totale des fonctions sensorimotrices associées aux niveaux en-dessous de la lésion. L'étendue des pertes fonctionnelles dépend du niveau de dommages neurologiques et de la quantité de tissu spinal préservé. En plus d'affecter les fonctions sensorimotrices, les lésions à la moelle épinière peuvent également affecter le système nerveux sympathique, donnant ainsi lieu à des problèmes reliés aux fonctions automatiques du corps.

La classification moderne des lésions spinales se fait aujourd'hui selon l'*International Standards for Neurological Classification of Spinal Cord Injury* (ISNCSCI). Ces standards comprennent trois échelles de classification, soit (1) l'échelle moteur du *American Spinal Injury Association* (ASIA), qui évalue la force motrice de chaque groupe de muscles innervé par une même racine nerveuse spinale, (2) l'échelle sensitive ASIA, qui évalue les sensations associées au toucher ou piqûre légère de 28 zones de la peau qui sont chacune innervées par

une racine nerveuse différente, et (3) l'échelle de déficience ASIA (AIS), qui attribue une mesure sensorimotrice globale à la blessure médullaire.

## 1.2 Imagerie par résonance magnétique

La technique d'imagerie par résonance magnétique se base sur l'excitation magnétique des atomes et molécules composant les différents tissus du corps. Lorsque ces atomes sont excités par un champ magnétique variable, ils émettent de l'énergie sous différentes formes dépendant de leur structure atomique et des paramètres du champs magnétique appliqué. Ainsi, différents tissus émettent différents signaux en réponse à différents champs, et l'on peut mesurer ces réponses pour construire une image de l'intérieur du corps.

### 1.2.1 Séquences

Les champs magnétiques appliqués pour exciter les atomes suivent certaines séquences de pulses et gradients prédéfinis. Chacune de ces séquences possède différentes caractéristiques et permet de visualiser les tissus différemment. Un des types de séquences les plus utilisés en clinique est la séquence pondérée en T1. Dans cette séquence, les spins des protons de tous les atomes sont initialement alignés par un champ magnétique fixe. Ensuite, ils sont excités par un pulse magnétique de façon à ce que leurs spins se retrouvent dans le plan transversal par rapport au champ initial. Une fois que le pulse est terminé, les spins reviennent progressivement au plan du champ initial, cependant, ce retour ne se fait pas à la même vitesse pour tous les types d'atomes composant les tissus biologiques. La séquence T1 consiste donc à mesurer le temps de retour des spins de chaque section d'une image afin de déterminer le ton (de blanc à noir) que prendra cette section sur l'image. Ce type de séquence utilise un temps de répétition (TR) et un temps écho (TE) relativement courts. Le temps de répétition correspond au temps entre les pulses générés pour exciter les spins protoniques des atomes vers le plan transversal, tandis que le temps écho correspond au temps entre ce pulse excitatif et la mesure du signal de résonance magnétique émis par les atomes excités.

La séquence pondérée en T2 est un autre type qui est très utilisé en clinique. Dans ce type de séquence, on mesure le temps auquel les spins des protons excités par un pulse perdent leur cohérence de phase dans le plan transversal. La cohérence de phase correspond à la synchronisation dans les spins protoniques entre les atomes. Cette mesure requiert un temps de répétition et un temps écho plus longs que la séquence T1. Il est à noter que ces temps peuvent varier même à l'intérieur d'un certain type de séquence, à la discrétion du radiologiste opérant le scan et de plusieurs protocoles existants.

Chaque type de séquence possède un spectre (de noir à blanc) particulier pour les différents tissus du corps. La séquence T2 rend un teint très clair pour le liquide cérébrospinal et un teint foncé pour le tissu de la moelle épinière. Ceci permet une bonne démarcation entre la moelle épinière et le liquide cérébrospinal l'entourant, et c'est pour cette raison que c'est ce type d'IRM qui est habituellement utilisé pour segmenter la moelle épinière (McCoy et al., 2019, Gros et al., 2019). La séquence T2 permet aussi de voir plusieurs types de lésions sur la moelle épinière, et elle est souvent utilisée pour identifier et mesurer ces lésions (Gupta et al., 2014; Magu et al., 2015; Miyanji et al., 2007; Martineau et al., 2019). Il existe plusieurs autres types de séquences, cependant, seulement les scans de séquence T2 ont été utilisés dans le cadre de cette étude.

### 1.2.2 Autres paramètres

Plusieurs paramètres sont indépendants du type de séquence utilisée et ajoutent à la variété des IRM. On retrouve parmi ces paramètres le plan du scan. Il y a deux plans communément utilisés pour visualiser la moelle épinière, soit le plan axial et le plan sagittal. Un plan axial indique que le scan est composé d'une série d'images prises perpendiculairement à la moelle épinière, où chaque image représente en quelque sorte une tranche du corps vu du haut. Un plan sagittal indique que le scan est composé d'une série d'images prises sur un plan parallèle à la moelle épinière, où chaque image représente en quelque sorte une tranche du corps vu de côté. Les images individuelles des scans axiaux ou sagittaux, qui couvrent chacune une certaine

épaisseur du corps qu'on appelle l'épaisseur de coupe, sont ensuite superposées pour construire une vue tri-dimensionnelle de l'intérieur du corps.

On appelle les scans axiaux ou sagittaux *anisotropes*. Ceci indique qu'ils sont basés sur un seul plan deux dimensionnel. La résolution de ces scans est d'ordinaire nettement supérieure dans le plan où les images ont été prises, la résolution dans les autres plans dépendant de l'épaisseur de coupe et de l'incrément entre les coupes. La résolution tridimensionnelle d'une IRM anisotrope dépend donc de la résolution dans le plan de l'image, de l'épaisseur de coupe, et de l'incrément entre les coupes. Ces trois facteurs se combinent pour donner la résolution d'un *voxel*, qui est le terme pour désigner le volume unitaire tridimensionnel utilisé pour définir la résolution d'une IRM. À l'opposition des scans anisotropes, il existe aussi des scans où la résolution est la même dans les trois plans. On appelle ces scans *isotropes* ou encore des scans tridimensionnels.

### 1.3 Pronostic des lésions médullaires

Le critère clinique le plus important dans le pronostic de lésions spinales est l'échelle de déficience ASIA (AIS). En effet, plusieurs études démontrent que l'évaluation AIS initiale de patients atteints de lésions spinales traumatiques a une corrélation significative avec leur degré de rétablissement (Al-Habib et al., 2011; Wilson et al., 2012; Martineau et al., 2019). Cependant, plusieurs études ont tenté d'améliorer les prédictions basées sur cette échelle en se basant sur des lectures IRM de patients atteints, et certains autres indicateurs potentiels ont été découverts (Al-Habib et al., 2011; Wilson et al., 2012).

Certaines études suggèrent que la compression extrinsèque de la moelle épinière pouvant être visualisée par IRM pourrait être corrélée avec la sévérité initiale de blessures médullaires cervicales et thora-lombaires, cependant, la relation avec un pronostic possible demeure incertaine (Martineau et al., 2019). En effet, plusieurs analyses de régression multivariables n'ont su démontrer une association significative entre ce critère et le rétablissement prospectif des patients, surtout lorsque le statut neurologique des patients a été tenu en ligne de compte

(Miyanji et al., 2007; Skeers et al., 2018). Cependant, d'autres études suggèrent que certaines anomalies observées par IRM initialement suite à une blessure pourraient être associées avec le rétablissement prospectif des patients (Martineau et al., 2019).

On retrouve parmi celles-ci l'étude (Miyanji et al., 2007), qui, en utilisant une régression multivariable, a su démontrer une relation significative entre le score moteur ASIA (AMS) prospectif des patients et la présence d'une hémorragie initiale dans la moelle épinière. Cependant, certains facteurs, tels qu'un temps de suivi pouvant être très court et variable (entre 1 et 35 mois suivant l'incident) ainsi que l'inclusion dans l'échantillon de 22 patients asymptomatiques pour le contrôle, viennent considérablement limiter les conclusions de cette étude (Martineau et al., 2019).

Une autre étude (Gupta et al., 2014) a aussi trouvé une relation possible entre la présence d'hémorragie spinale et le AMS subséquent des patients. Cependant, dans ce cas-ci, les suivis ont été conduits peu de temps après les incidents (entre trois et six mois), ce qui vient limiter l'étendue des résultats, étant donné que la période de rétablissement pour ce type de blessure peut aller jusqu'à six mois (Martineau et al., 2019). Les auteurs de cette étude suggèrent aussi que la longueur des hémorragies observées sur des IRM pondérés en T2 pourrait potentiellement être un indicateur du rétablissement futur des patients, puisque cette variable a presque atteint le seuil de signification dans leur analyse de régression multivariable pour l'AMS.

Une autre étude (Wilson et al., 2012) a trouvé que la présence d'œdème et d'hémorragie initiales de la moelle épinière était reliée au statut fonctionnel dégénératif des patients entre 6 et 12 mois suivant les incidents. Cependant, les auteurs de cette étude ont n'ont pas évalué si une telle relation existait avec le rétablissement neurologique.

Afin de clarifier la question, l'étude (Martineau et al., 2019) s'est basée sur trois critères visibles par IRM, soit la présence d'hémorragie intramédullaire, la compression maximale de la moelle épinière (MSCC), qui est une mesure de la réduction de la surface axiale, et la

longueur de la lésion intramédullaire. La cohorte de cette étude était constituée de 82 patients provenant du même centre de trauma spécialisé dans les blessures spinales. Les critères d'inclusion pour tous les sujets étaient (1) âgés de 16 ans et plus, (2) présence de trauma au niveau cervical entre C0 et C8 avec un niveau de blessure neurologique, (3) opération performée à l'intérieur de cinq jours suivant l'accident (4) IRM et examen neurologique réalisé à l'intérieur de 72 heures suivant l'accident (5) aucun déficit neurologique antérieur à l'accident, (6) suivi réalisé entre 6 et 12 mois après l'accident.

Les évaluations initiales des patients ont été faites selon l'échelle de déficience ASIA (AIS) et selon le score moteur ASIA (AMS) normalisé de façon à produire un pourcentage. Les évaluations lors du suivi des patients ont aussi été réalisées selon le AIS et le AMS normalisé, en plus d'inclure la conversion de grade AIS binaire (pas de changement ou bien changement de 1 grade et plus) et l'atteinte ou non d'un niveau AIS de grade D ou E.

Les IRM pré-opératifs considérés lors de cette étude ont tous été performé sur une machine de 1.5 Tesla du même modèle en utilisant les séquences standardisées sagittal T2 TSE, sagittal T1 TSE, sagittal STIR, sagittal 2D MEDIC, axial 2D MEDIC et axial T2 TSE. La présence d'hémorragie intramédullaire a été diagnostiquée en identifiant des régions hypointenses avec un mince contour hyperintense sur des images T2 (Gupta et al., 2014; Magu et al., 2015; Miyanji et al., 2007). La longueur des lésions intramédullaires a été mesurée en millimètres sur des image sagittales pondérées en T2 en considérant la lésion de longueur maximale sur la moelle épinière et pouvait inclure des lésions d'œdème, de contusion ou d'hémorragie (Gupta et al., 2014; Magu et al., 2015). La compression maximale a été calculée en utilisant des images sagittales pondérées en T2 et correspond à la mesure du rétrécissement du diamètre de la moelle épinière par rapport à la moyenne des sections immédiatement au-dessus et en-dessous de la compression.

Les résultats de cette étude confirment une corrélation entre certaines des anomalies observées sur les IRM et les scores AMS et AIS des patients lors de leurs suivis, surtout lorsque les IRM sont utilisées conjointement avec les résultats AIS initiaux pour effectuer les prédictions. En

effet, en utilisant la méthode LASSO pour construire leurs modèles, les auteurs ont trouvé un coefficient de détermination R-carré de 0.662 en prédisant le score moteur AMS à l'aide de la présence d'hémorragie intramédullaire et du grade AIS initial. La même mesure de détermination obtenue en utilisant seulement le grade AIS initial était de 0.636, indiquant une amélioration de 0.026 pour la prévision avec le critère IRM. Les auteurs ont aussi trouvé un coefficient de validation avec index-c de 0.903 en prédisant la probabilité que les patients obtiennent un grade AIS de D ou E à leur suivi. Cette prédiction a été obtenue en utilisant les mêmes variables d'entrée, soit la présence d'hémorragie intramédullaire et le grade AIS initial. Dans ce cas, l'index-c obtenu en utilisant seulement le grade AIS initial pour effectuer la prédiction était de 0.873, indiquant une amélioration de 0.03 pour la prédiction avec l'addition du critère d'imagerie. Les auteurs ont aussi trouvé un index-c de 0.704 en prédisant l'amélioration d'au moins un grade de l'AIS initial à l'aide de la présence d'hémorragie interne, de la longueur de la lésion et de l'AIS initial. Cependant, l'index-c obtenu en utilisant seulement l'AIS initial était de 0.727, soit 0.023 supérieur à ceci, indiquant une diminution de performance lorsque les critères IRM étaient ajoutés.

On peut donc conclure de ces résultats que la présence d'hémorragie intramédullaire et la longueur de la lésion peuvent contribuer à établir des pronostics pour les patients atteints de lésions traumatiques à la moelle épinière. Ces résultats sont d'autant plus importants étant donné que les études précédentes (Gupta et al., 2014; Miyanji et al., 2007) ont considéré uniquement les critères IRM pour tirer des conclusions vis-à-vis le rétablissement des patients, tandis que cette étude évalue ces facteurs en combinaison avec le statut neurologique initial, se rapprochant ainsi davantage d'un résultat qui pourrait être utilisé dans un contexte clinique (Martineau et al., 2019). Cependant, seule la présence d'hémorragie intramédullaire améliore réellement certaines prédictions par rapport à l'utilisation du grade AIS seul. Le grade AIS préopératoire demeure donc le facteur le plus important dans la prédiction de l'évolution des blessures spinales.

## 1.4 Apprentissage profond : contexte actuel et applications

L'intelligence artificielle connait aujourd'hui une croissance fulgurante dans de nombreux secteurs d'industrie. Cette croissance est principalement poussée par les développements et avancées en apprentissage profond, qui sont à leur tour poussés par le développement des processeurs, des framework logiciels, et la disponibilité des données massives. Les avancées sont particulièrement spectaculaires dans le domaine de la vision par ordinateur, où des réseaux d'apprentissage profond arrivent maintenant à des taux d'erreurs se rapprochant du taux d'erreur minimal de Bayes (Lundervold et Lundervold, 2019) pour des taches de classification d'objets. Ceci représente une avancée énorme, considérant que de telles tâches étaient considérées jusqu'à récemment comme très difficiles et loin d'être résolues.

Les progrès ne sont cependant pas restreints au domaine de la vision par ordinateur. Dans le domaine de la santé, l'utilisation de l'apprentissage profond mène présentement au développement de réseaux informatiques intelligents, à la possibilité de prédire des séquences génétiques, à l'enregistrement verbal de dossiers et à des diagnostics automatiques à partir d'imagerie médicale (Yang Lu, 2019). L'apprentissage profond est maintenant aussi utilisé dans le traitement du langage naturel, la reconnaissance et la synthèse vocale, et l'analyse de données non-structurées (Lundervold et Lundervold, 2019), ainsi que plusieurs autres domaines, comme l'agriculture, où il utilisé pour analyser le contenu complexe des sols, ou en éducation, où il est utilisé par les chercheurs pour mieux comprendre les processus d'apprentissage et créer des outils éducatifs, ainsi qu'en sécurité informatique, en vente, transport, manufacture, robotique et finance (Yang Lu, 2019).

## 1.5 Segmentation de la moelle épinière

### 1.5.1 Défis

Il existe plusieurs défis associés à la segmentation de la moelle épinière. Parmi ces défis, bon nombre ont rapport à l'obtention des données et leur variété. En effet, les IRM sont presque toujours protégées par le droit à la confidentialité des patients, ce qui rend leur accès limité.

Les chercheurs voulant travailler sur des projets liés aux IRM doivent donc généralement le faire en partenariat avec des institut médicaux. Cela limite donc de prime abord le nombre de scans auxquels un projet donné peut avoir accès. Certains projets réussissent à réunir des scans de plusieurs instituts différents (Gros et al., 2019), cependant, cela requiert plus d'organisation et ce n'est pas tous les projets qui y parviennent. De plus, même dans les cas où des projets utilisent de multiples sources de données, ce n'est souvent pas suffisant pour tirer des conclusions généralisables (Paugam et al., 2019), car la variété des données IRM que l'on retrouve dans le monde réel est très grande.

Le besoin d'une grande quantité et variété de données pour l'apprentissage profond à partir d'IRM provient de multiples facteurs. D'abord, un réseau d'apprentissage profond en soi requiert naturellement une variété de données représentant la variété retrouvée dans le monde réel. Dans le cas de l'IRM, ce critère peut être difficile à rencontrer, car les IRM ont de nombreux paramètres (contraste, résolution, temps de répétition, etc.) qui peuvent varier d'un scan à l'autre et d'une machine ou d'un institut à l'autre. De plus, les sujets composant les échantillons sont souvent biaisés (âge, sexe, ethnie, type de lésions, sévérite des lésions, etc.) lorsqu'on prend des données d'une seule ou de quelques institutions. Tous ces facteurs s'accumulent et posent de sérieux défis pour la création de réseaux intelligents à partir d'IRM. En effet, dans les rares cas où le code source de ces réseaux est disponible, des essais avec des données d'autres institutions donnent habituellement des résultats décevants, bien que certaines exceptions existent (McCoy et al., 2019).

Une autre source de défis pour la segmentation de la moelle épinière et la segmentation d'IRM en général est le temps requis par les cliniciens pour annoter les échantillons (Gros et al., 2019; McCoy et al, 2019). Ceci est dû au fait que les cliniciens doivent annoter manuellement chaque coupe de chaque IRM, ce qui peut représenter un très grand nombre d'images. Vient s'ajouter à cette problématique le fait que les annotations manuelles comprennent toujours de la variabilité intra et inter-observateur, voulant dire que les annotations faites par un même clinicien ne seront pas nécessairement les mêmes à un autre moment, et ces annotations ne correspondront pas nécessairement aux annotations d'un autre clinicien. On peut partiellement

pallier ce problème en augmentant le nombre de cliniciens annotant les données, mais ceci vient encore ajouter au temps requis pour l'annotation.

### 1.5.2 Approches

Une étude récente (Gros et al., 2019) s'est basée sur des IRM de 6 centres différents afin de construire une série de réseaux convolutifs visant à segmenter la moelle épinière et identifier les lésions reliées à la sclérose en plaques. L'échantillon pour cette étude était constitué de 1042 patients avec un total de 1943 scans comportant une grande variété de pathologies, contrastes, résolutions, dimensions et orientations. En effet, l'échantillon comprenait 459 patients en santé, 471 patients atteints de sclérose en plaques, ainsi que des patients atteints d'autres types de maladies dégénératives et quatre patients atteints de blessures à la moelle épinière. Les scans ont été obtenus sur des machines 3T et 7T de plusieurs plateformes (Siemens, Philips, GE) et les contrastes incluent des séquences pondérées en T2 (n = 904), T1 (n = 151) et T2* (n = 888). La surface de couverture variait largement entre les scans et pouvait inclure la région cervicale avec ou sans le cerveau, la région thoracique, la région lombaire, ou une combinaison de celles-ci. L'échantillon était composé de 451 scans isotropes et 1492 scans anisotropes dont 1010 étaient d'orientation axiale et 482 d'orientation sagittale.

La méthodologie de cette approche est divisée en trois étapes. D'abord, un réseau convolutif est utilisé afin d'identifier la ligne centrale de la moelle épinière. Ensuite, un algorithme d'optimisation vient cerner un volume autour de la moelle épinière. La troisième étape peut prendre deux formes. Soit un réseau convolutif utilise le volume atour de la moelle épinière afin de segmenter celle-ci, soit un autre réseau convolutif utilise ce même volume pour segmenter les lésions. Ces deux réseaux sont indépendants l'un de l'autre et peuvent être utilisés conjointement.

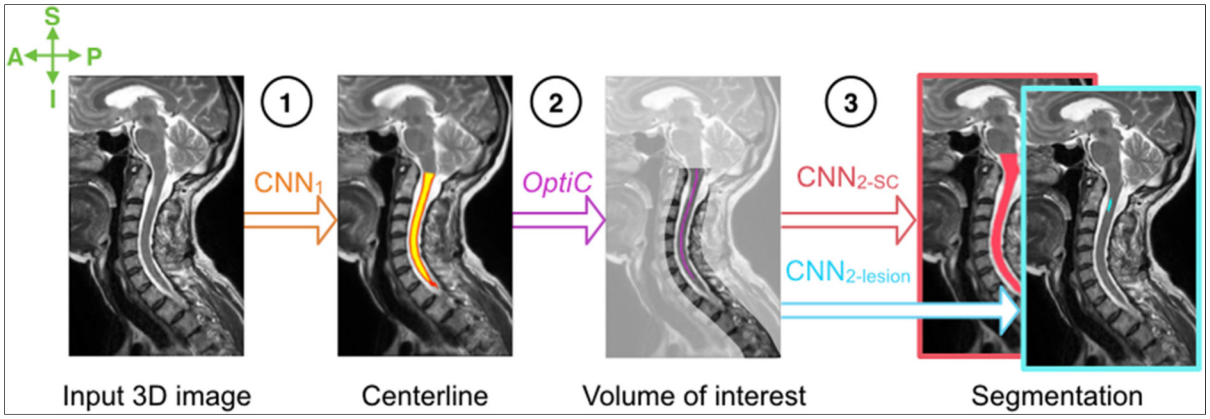



Figure 1.1 Processus de segmentation de la moelle épinière et des lésions
Tirée de Gros et al., 2019

Le réseau convolutif qui identifie la ligne centrale de la moelle épinière, nommé $CNN_1$, est un réseau 2D qui analyse chaque tranche des volumes d'entrée. Avant de débuter l'entrainement, une surface de 96 × 96 pixels est d'abord extrait de chaque tranche puis normalisée avec la normalisation de batch (Ioffe et Szegedy, 2015) afin d'obtenir une distribution standard normale. Ce réseau est basé sur une architecture U-Net modifiée (voir ANNEXE V). Le nombre de couches contractives a été réduit de quatre à deux et les convolutions dans ces couches ont été replacé par des convolutions dilatées avec un taux de dilation de trois. Chacune de ces convolutions est suivie d'une normalisation de batch, une fonction d'activation ReLU, et une couche dropout (Srivastava et al., 2014) de 20%.

Un entrainement différent a été effectué pour chaque type de contraste (T2, T2*, T1), donnant ainsi trois modèles différents pour $CNN_1$. Le processus d'entrainement a inclus l'utilisation d'un optimisateur Adam (Kingma et Ba, 2014) avec un taux d'apprentissage de $1\times10^{-05}$, une taille de batch de 32, et 100 époques. Une fonction de perte basée sur le coefficient Dice (Milletari et al., 2016) a été utilisée comme fonction objective pour entrainer le réseau, étant donné les trouvailles récentes (Drozdzal et al., 2018; Perone et al., 2017; Sudre et al., 2017) indiquant son insensibilité aux déséquilibres des classes. De plus, une augmentation poussée des données d'entrainement a été effectuée, incluant l'inversion des images, un déplacement

de ±10 voxels dans chaque direction, une rotation de ±20º dans chaque direction, et des déformations élastiques avec un coefficient de 100 et un écart-type de 16.

Le réseau pour la segmentation de la moelle épinière, $CNN_{2\text{-}SC}$, et le réseau pour la segmentation des lésions de sclérose en plaques, $CNN_{2\text{-}Lesion}$, suivent des procédés d'entrainement et ont des architectures semblables. Les deux sont des réseaux convolutifs 3D avec des volumes d'entrée différents ($64 \times 64 \times 48$ pour $CNN_{2\text{-}SC}$ et $48 \times 48 \times 48$ pour $CNN_{2\text{-}Lesion}$). Ces dimensions sont les résultats de tests préliminaires et représentent un compromis en le déséquilibre des classes, le risque de overfitting et le coût de calcul. Un algorithme de normalisation des intensités est appliqué sur ces volumes d'entrée afin d'homogénéiser la distribution des intensités sur une plage standardisée d'intensités (Pereira et al., 2016; Shah et al., 2011). Suivant ceci, une normalisation de batch est appliquée avant que les données soient envoyées dans le réseau. L'architecture de ces réseaux est basée sur l'architecture 3D U-Net avec une réduction de la profondeur des couches contractives et expansives de trois à deux. Cette réduction sert à réduire le nombre de paramètres et la mémoire requise lors de l'entrainement.

Un entrainement séparé a été effectué pour les trois contrastes de chacun des deux réseaux, donnant un total de six modèles. Un optimisateur Adam a été utilisé avec un taux d'apprentissage de $5\times10^{-5}$, une taille de batch de 4, et 300 époques au total. La même fonction de perte basée sur le coefficient de Dice que dans le réseau $CNN_1$ a été utilisée et un dropout de 40% a été appliqué. L'augmentation des données a principalement été restreinte à l'inversion des images, avec l'addition des petites érosions et dilatations locales des contours des lésions annotées manuellement, qui ont été ajoutées afin de tester la confiance du réseau par rapport aux bordures subjectives.

Pour ce qui est de la segmentation de la moelle épinière, les auteurs jugent que plusieurs aspects des résultats sont très prometteurs. Ils citent notamment des exemples où ils ont segmenté avec succès des moelles comprimées ou atrophiées, des images avec des contrastes pauvres entre la moelle et les structures environnantes, et des images avec des couvertures peu fréquentes. Ils

comparent leur approche, qu'ils baptisent *Deepseg*, avec une approche précédente ayant connu beaucoup de succès, appelée *Propseg* (De Leener et al., 2014). En comparant avec toutes les données de test, ils obtiennent une moyenne pour le coefficient de similarité Dice de 0.946 avec un écart-interquartile de 4.6 pour Deepseg et un coefficient Dice de 0.879 avec un écart-interquartile de 18.3 pour Propseg. Ce résultat est significatif ($p < 0.001$) et montre que Deepseg est supérieur, du moins sur leur échantillon de test. Les auteurs comparent aussi les deux approches sur des patients présentant une atrophie sévère de la moelle épinière et trouvent un coefficient Dice de 0.929 pour Deepseg contre 0.820 pour Propseg, montrant encore une fois la supériorité de Deepseg. Le modèle de Deepseg a aussi été testé sur les données de deux sites qui n'avaient pas été utilisés durant l'entrainement. Pour ce test, les auteurs trouvent un coefficient Dice de 0.933, indiquant une capacité de généralisation de leur réseau semblable à sa performance avec les données des sites utilisés lors de l'entrainement (0.946). Les résultats de Propseg sur cet échantillon ne sont cependant pas discutés.

Une autre approche (McCoy et al., 2019) s'est basée sur des scans de 47 patients du Zuckerberg San Francisco General Hospital entre 2015 et 2017. Dans cette étude, tous les sujets avaient subi une blessure spinale traumatique et étaient âgés de plus de 18 ans, avec une moyenne de 55 ans et un écart-type de 19 ans. Il y avait environ deux fois plus de patients masculins que de patients féminins, et seuls les patients présentant des contusions traumatiques cervicales ou thoraciques ont été inclus dans l'échantillon, excluant ainsi les patients atteints de traumas pénétrants. Tous les scans pré-opérationnels et ont été effectués moins de 24 heures suivant les traumas et ont utilisé la même séquence axiale (T2 fast spin-echo), le même temps de répétition, temps d'écho, épaisseur de coupe, longueur de train echo, cadre de vue, grandeur de pixel maximale dans le plan, et provenaient de la même machine.

Dans cette étude, les chercheurs ont bâti trois réseaux légèrement différents pour tenter de segmenter la moelle épinière et les contusions intramédullaires. Chaque réseau a été testé pour segmenter la moelle épinière ainsi que les lésions. Le premier réseau est nommé BASICseg-1 et est basé sur une architecture U-Net modifiée. Dans la section contractive du réseau, les auteurs ont utilisé des convolutions 3×3 *zero-padées* au lieu de convolutions 3×3 *non-paddées*,

en plus d'ajouter une couche dropout (Srivastava et al., 2014) de 20% à chaque convolution des quatre premières couches de cette section (sur cinq couches). Leurs modifications à la section expansive du réseau sont plus étendues. Plutôt que d'utiliser quatre couches comportant chacune deux convolutions 3×3 non paddées et séparées par une *up-convolution* 2×2, les auteurs ont choisi d'utiliser deux convolutions 3×3 zéro-paddées pour chaque couche (une après le upsampling/concaténation et une pour séparer les couches). De plus, les auteurs ont ajouté une fonction d'activation sigmoïde suivant la convolution finale 1×1 à la sortie du réseau. Leur réseau a donc un total de 19 couches convolutives. L'entrainement a été effectué à l'aide d'un gradient de descente stochastique avec un taux d'apprentissage de 1e-5 et un optimisateur Adam (Kingma et Ba, 2014). Ce réseau est le plus performant des trois pour la segmentation de la moelle épinière, ayant atteint un coefficient Dice de 0.93 contre 0.91 et 0.90 pour BASICseg-2 et BASICseg-3 respectivement.

Le deuxième réseau, BASICseg-2, reprend l'architecture de BASICseg-1 avec deux changements. Premièrement, une normalisation de batch est placée entre chaque convolution et sa fonction ReLU subséquente, et ce, pour la couche contractive aussi bien que la couche expansive. Cette addition vient annuler le besoin de dropout dans la couche expansive; le dropout est donc éliminé. Deuxièmement, le nombre de canaux créés à chaque couche a été doublé. Le reste de l'architecture n'a pas été modifiée.

Le troisième réseau, BASICseg-3, reprend l'architecture de BASICseg-2 et ajoute un filtre dans la couche de sortie. Ce filtre sert à atténuer le bruit causé par les régions où la moelle épinière se trouve comprimée par des lésions. Il prend la forme d'un tenseur de dimension 16×16 situé juste avant la convolution finale 1×1 du réseau. Les poids composant le filtre sont ajustés durant l'entrainement de la même façon que les autres poids, c'est-à-dire en minimisant le coefficient Dice du système.

Ce réseau est le plus performant pour la segmentation des contusions à la moelle épinière. Pour mesurer la performance sur cette tâche, les auteurs ont corrélé le volume des lésions segmentées avec le score moteur inférieur des patients à leur admission et à leur décharge de

la clinique. Les corrélations ont été effectuées avec une simple régression linéaire. La corrélation avec le score moteur à l'admission a donné un coefficient de -4.583 ave P = 0.002, tandis que la corrélation avec le score moteur à la décharge a donné un coefficient de -4.030 avec P = 0.009. Les coefficients négatifs indiquent que, pour chaque point de dégradation du score moteur, les lésions augmentent de plusieurs millimètres cubes. Il est intéressant de noter ici que les auteurs ne précisent pas à quel moment après l'admission les décharges ont lieu. Cela aurait été intéressant à savoir, car si les décharges ont eu lieu à un moment suffisamment éloigné dans le temps, un lien entre les lésions et le pronostic des patients aurait potentiellement pu être établi (voir la section 1.3). De plus, le fait que les auteurs ne spécifient pas d'intervalle entre l'admission et la décharge rend incertaine l'interprétation des résultats reliés à la décharge, limitant ainsi notre capacité à en tirer des conclusions. Il est aussi intéressant de noter que les auteurs ne présentent pas de résultats de coefficient Dice pour la segmentation des lésions, tel qu'ils l'ont fait pour la segmentation de la moelle épinière entière. Cette mesure de corrélation entre les annotations des radiologistes et les résultats du réseau aurait pu nous donner une meilleure idée de la performance réelle de la segmentation des lésions.

Les auteurs comparent la performance de BASICseg-1, leur réseau le plus performant pour la segmentation de la moelle épinière, aux approches *Deepseg* et *Propseg* discutées précédemment. En comparant les résultats de ces trois réseaux sur leurs données de test, ils trouvent une différence estimée entre les coefficients Dice de 0.12, $P < 0.001$ pour la comparaison avec Propseg, et une différence de 0.03, $P < 0.056$ pour la comparaison avec Deepseg. De ceci, les auteurs concluent une différence significative avec Propseg et une différence marginalement significative avec Deepseg, en faveur de leur réseau. En comparant les résultats des réseaux avec seulement les coupes présentant des lésions, ils trouvent une différence de 0.220, P = 0.001 pour la comparaison avec Propseg, et une différence de 0.102, P = 0.020 pour la comparaison avec Deepseg. Les auteurs concluent de ceci une différence significative avec les deux réseaux, en faveur de leur réseau. Finalement, les auteurs comparent les résultats des réseaux avec seulement les coupes ne présentant pas de lésion. Dans ce cas, ils trouvent une différence significative avec Propseg de 0.0953, P = 0.001 mais ne trouvent pas de différence significative avec Deepseg (0.013, P = 0.43).

Les auteurs concluent de ces comparaisons que leur réseau est au moins aussi performant que Deepseg pour segmenter la moelle épinière, ce qui peut sembler une conclusion quelque peu hâtive. Le fait que Deepseg ait été entrainé sur une base de données complètement différente et plus variée est un facteur très important dans la détermination de sa vraie mesure de performance relativement à BASICseg, facteur que les auteurs ne tiennent pas en ligne de compte dans leur article. En effet, les réseaux BASICseg ont tous été entrainés avec des scans provenant de la même machine et utilisant exactement les mêmes paramètres. Ceci indique que ces réseaux risquent fortement d'être plus performants sur des données identiques à ces données, et c'est précisément avec de telles données que les tests et comparaisons avec Deepseg sont effectués. On peut donc conclure qu'il existe presque certainement un biais dans les résultats en général et plus particulièrement dans les comparaisons avec Deepseg. Les auteurs eux-mêmes admettent partiellement ce fait en mentionnant que les trois réseaux BASICseg montrent des signes de *overfitting*. Donc, étant donné que les résultats de BASICseg ne sont que marginalement meilleurs que ceux de Deepseg, cela pourrait porter à croire que Deepseg est probablement supérieur dans la segmentation sur une plus grande variété de scans.

## 1.6 Spinal Cord Toolbox

Le Spinal Cord Toolbox (SCT) est un logiciel open-source destiné à l'analyse et au traitement d'IRM de la moelle épinière. Le SCT vise la standardisation des modèles et procédures d'analyse entourant l'imagerie de la moelle épinière (De Leener et al., 2017). Pour ce faire, il propose non seulement une variété de modèles et d'atlas, mais aussi des algorithmes intégrés pour segmenter les images et adapter les résultats aux modèles proposés. Le SCT vise donc une grande variété d'applications, pouvant aller de la segmentation de la moelle épinière entière à la segmentation de lésions de sclérose en plaques intramédullaires, en passant par des mesures de surfaces et la quantification précise de métriques spinales.

Le SCT offre trois algorithmes différents pour segmenter la moelle épinière, dont deux ont été introduits dans la section 1.5.2. Le premier algorithme à voir le jour, baptisé Propseg (De

Leener et al., 2014), est basé sur des méthodes d'analyse mathématiques et ne fait pas usage d'apprentissage machine. Sa performance semble avoir été dépassée (McCoy et al., 2019) par l'algorithme d'apprentissage profond développé plus récemment, qui est baptisé Deepseg 3D (Gros et al., 2019). Cet algorithme semble atteindre de très bonnes performances pour la segmentation de moelles épinières ne présentant pas de lésions traumatiques, atteignant des coefficients de Dice au-dessus de 93% pour des échantillons de cliniques non-inclues dans l'entrainement du réseau (Gros et al., 2019). Sa performance sur des sujets atteints de lésions traumatiques médullaires est cependant moins élevée. En effet, selon (McCoy et al., 2019), le coefficient de Dice atteint par Deepseg pour des sections de moelles présentant des lésions était légèrement supérieur à 70%, avec un coefficient de Dice pour la moelle entière de ces patients de 90%. Plus de travail est donc requis afin d'améliorer la capacité de Deepseg à segmenter des moelles épinières présentant des lésions traumatiques.

Le troisième algorithme pour segmenter la moelle épinière utilise un réseau baptisé Deepseg 2D. C'est cet algorithme qui est utilisé par défaut dans la version la plus récente du Spinal Cord Toolbox (version 5.5) pour segmenter la moelle épinière. Les segmentations de nos IRM produites avec Deepseg 2D ont été comparées aux segmentation produites par Deepseg 3D. Lors de ces comparaison, les segmentations de Deepseg 2D étaient visuellement supérieures à celles des autres réseaux. Ces comparaisons sont présentées en détail à l'ANNEXE IV.

L'architecture du réseau 2D de Deepseg, visible sur la Figure 1.2, est une adaptation de l'architecture U-Net. Elle comprend deux couches contractives/expansives et une couche bottleneck, pour un total de trois couches. Les blocs convolutifs utilisent des convolutions $3 \times 3$ avec padding = 1 afin de conserver la taille des canaux, et la régularisation est effectuée avec des couches dropout dans tous les blocs convolutifs. L'échantillonnage vers le haut est réalisé à l'aide de convolutions transposées $2 \times 2$ avec une stride = 2, de façon à ce que la taille des canaux soit égale dans les couches contractives et expansives. La couche de sortie est composée d'une convolution $1 \times 1$, suivie d'une couche sigmoïde afin de ramener les valeurs entre 0 et 1.

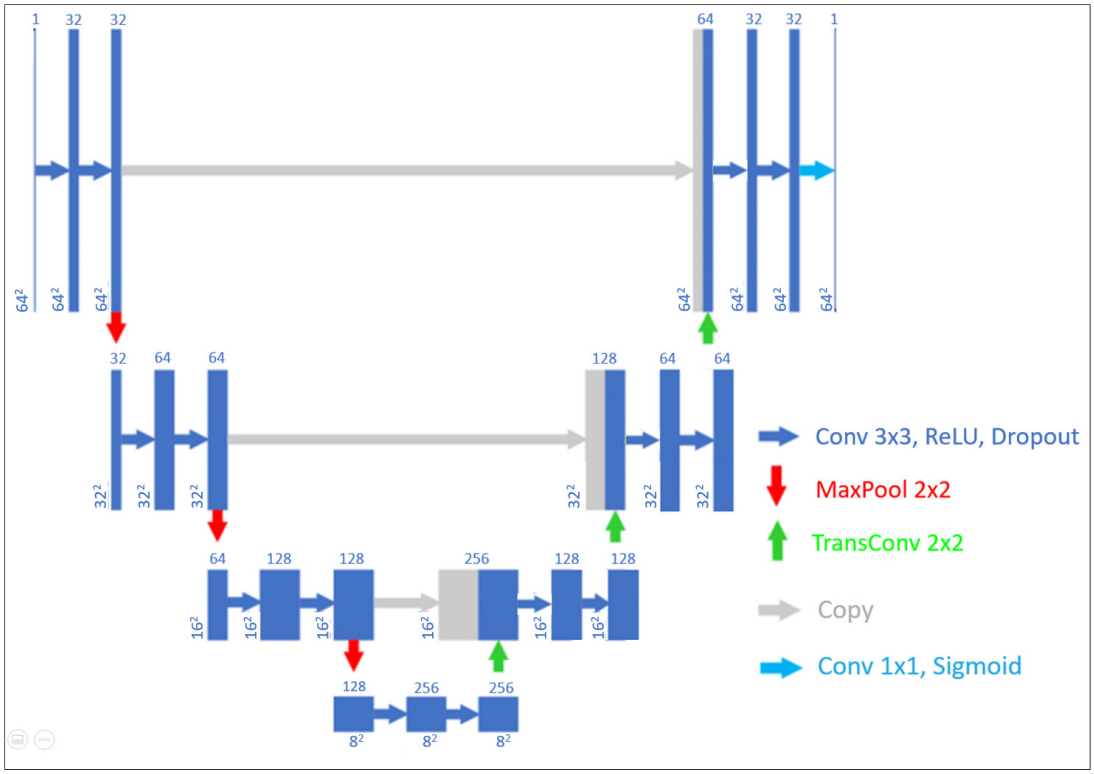



Figure 1.2 Réseau Deepseg 2D utilisé pour segmenter les IRM pondérées en T2

Le prétraitement des données lors de cette approche comprend une réorientation RPI, un rééchantillonnage à 0.5 mm dans le plan axial, un recadrage autour de la moelle épinière et une normalisation des tranches axiales. En fait, les prétraitements utilisés lors de l'étape axiale(ANNEXE I) de l'approche en deux étapes sont largement modelés sur les prétraitements de Deepseg 2D, la seule différence majeure étant que Deepseg 2D normalise les tranches axiales à des valeurs entre 0 et 383, tandis que notre approche normalise ces tranches à des valeurs flottantes entre 0 et 1.

Le post-traitement de Deepseg 2D comprend d'abord une étape où les trous sont remplis dans les segmentations de chaque tranche axiale. Ensuite, seulement l'objet le plus grand est conservé sur chaque tranche axiale. Puis, la segmentation est binarisée avec un seuil de signification de 0.5 et les deux étapes précédentes sont répétées.

# CHAPITRE 2

# MÉTHODOLOGIE

## 2.1 Mise en contexte

Cette section présente les principaux éléments qui sont communs aux deux approches principales, soit l'approche en deux étapes et l'approche semi-supervisée. L'ensemble des approches décrites sont réalisées avec le langage de programmation Python 3.6.9, en utilisant la librairie d'apprentissage profond PyTorch (version 1.9.0). Les réseaux sont entrainés sur la plateforme Colaboratory, avec un GPU NVDIA Tesla P100.

### 2.1.1 Description et prétraitement des données

Les données proviennent du Centre de recherche de l'Hôpital du Sacré-Cœur de Montréal. Elles sont composées d'imagerie à résonance magnétique provenant de patients atteints de lésions traumatiques à la moelle épinière. Cette étude s'intéresse uniquement aux scans sagittaux pondérés en T2, un type de scan pour lequel 94 patients dans la base de données possèdent au moins un scan. De ces 94 patients, huit possèdent plus d'un scan sagittal T2, correspondant à différents niveaux de la moelle épinière (cervical, dorsal, lombaire). Les scans utilisés totalisent donc 106. Pour les patients ne possédant seulement qu'un scan sagittal T2, la grande majorité de ces scans sont de type cervical.

La segmentation manuelle des images a été réalisée avec le logiciel ITK-Snap sur Windows 10. Pour accélérer le processus, les segmentations automatiques produites par le réseau Deepseg 2D du Spinal Cord Toolbox ont été utilisées comme point de départ pour les segmentations manuelles. Les images ont été segmentées en peinturant manuellement chaque coupe sagittale contenant des pixels appartenant à la moelle épinière. Pour chaque scan, il y avait en moyenne de quatre à cinq coupes contenant de telles pixels, pour un total de 473 coupes segmentées.

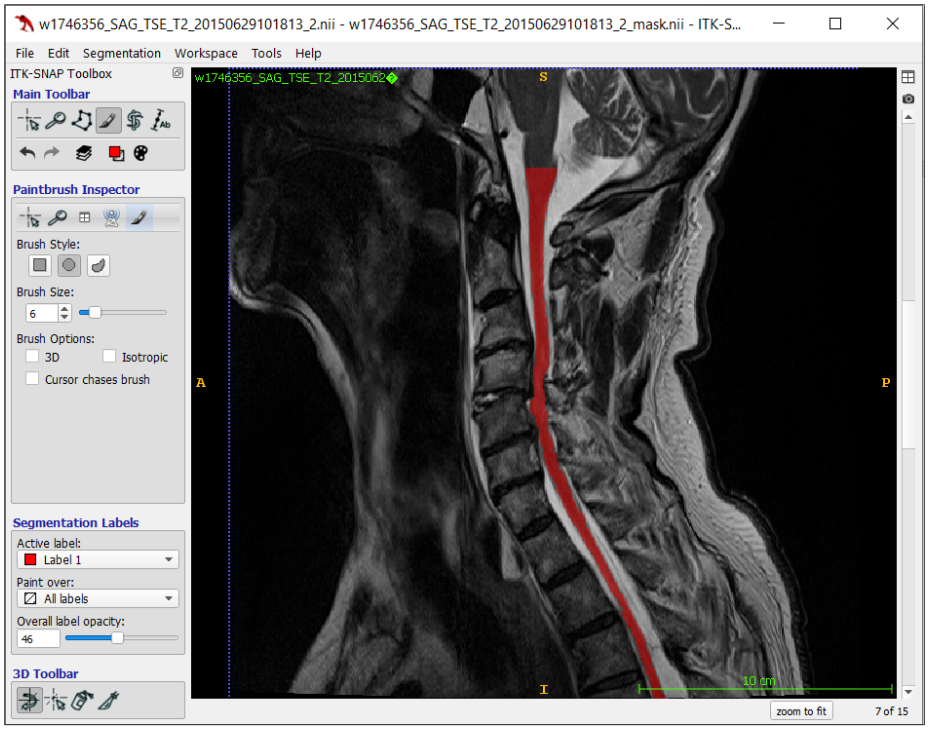



Figure 2.1 Vue de la segmentation manuelle d'une coupe sagittale cervicale avec le logiciel ITK-Snap

Les données ont été séparées aléatoirement en données d'entrainement et données de test. Les données de 19 patients, soit 20 scans, ont été réservés pour les tests, tandis que les données des autres 75 patients, soit 86 scans, ont été consacrés à l'entrainement. Ceci représente une séparation d'environ 20/80 pour les tests et l'entrainement. Étant donné que plusieurs patients ont plus d'un scan, il est important de veiller à ce tous les scans d'un même patient soient dans la même catégorie, afin de ne pas effectuer les tests avec un patient ayant aussi été utilisé pour l'entrainement, ce qui pourrait biaiser les résultats de façon favorable.

Les données doivent subir un prétraitement afin de standardiser certaines de leurs caractéristiques. Ceci est nécessaire pour que le réseau de neurones qui sera bâti par la suite puisse apprendre ces caractéristiques et généraliser à de nouvelles données ayant subi elles aussi les mêmes prétraitements et possédant les même caractéristiques clés.

Il existe deux procédures de prétraitement distinctes, soit une pour le réseau de segmentation axial et une pour le réseau sagittal. Ces deux procédures partagent cependant plusieurs étapes en commun, comme on peut le voir sur la Figure 2.2. Les différentes étapes du prétraitement des données sont décrites en détail à l'ANNEXE I.

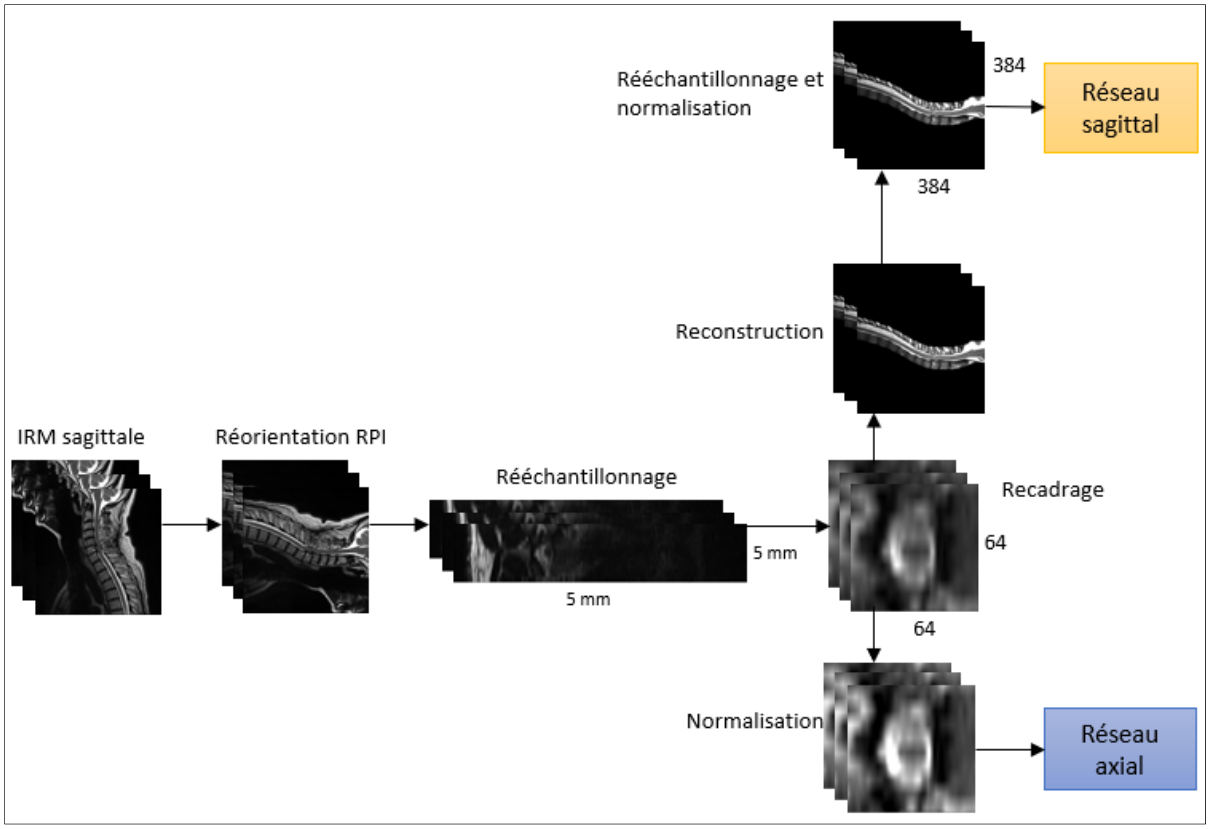



Figure 2.2 Étapes de prétraitements pour les réseaux axiaux et sagittaux

Les données sont augmentées durant l'entrainement afin de prévenir le *overfitting*. Cette augmentation est réalisée en opérant des transformations identiques sur les IRM et leurs masques correspondants. Ces transformations sont réalisées en utilisant le module *transforms.affine* de PyTorch. Elles incluent des transformations élastiques et des rotations. La sélection de ces types de transformations s'est faite de façon expérimentale, en essayant aussi plusieurs autres types de transformations telles que le recadrage, le miroitage et la translation.

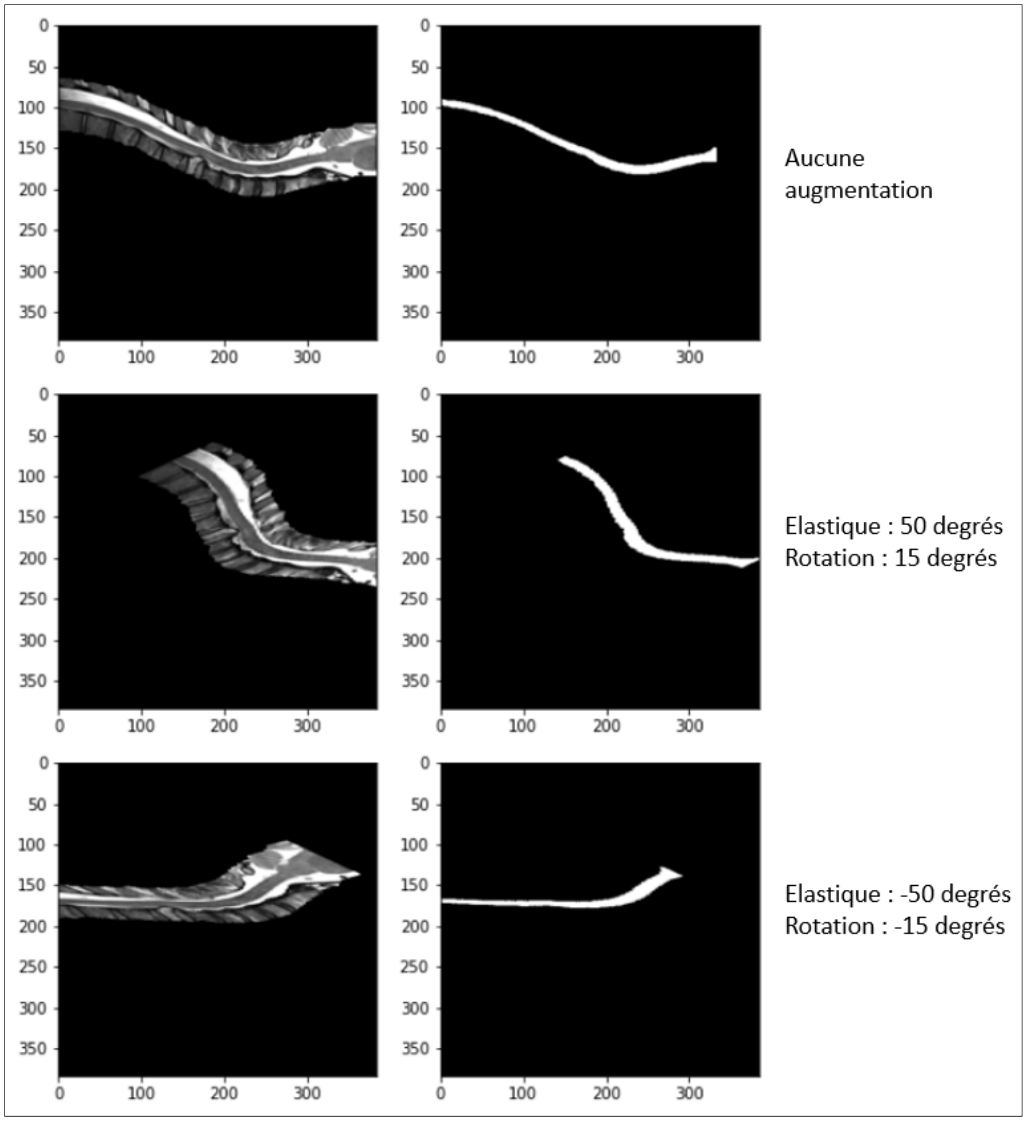



Figure 2.3 Illustration de transformations utilisées sur les tranches sagittales

Les valeurs permises pour les deux types de transformations retenues sont mesurées en degrés, permettant des valeurs entre -180 et 180. Durant l'entrainement, chaque coupe sagittale est augmentée avec des valeurs en degrés aléatoires comprises entre deux bornes prédéfinies. Par exemple, si les deux bornes pour les transformations élastiques sont de -50 et 50 degrés, chaque coupe sera augmentée selon une valeur aléatoire entre ces deux bornes. Ces bornes ont été ajustées séparément pour chacune des approches suivant un processus d'optimisation empirique.

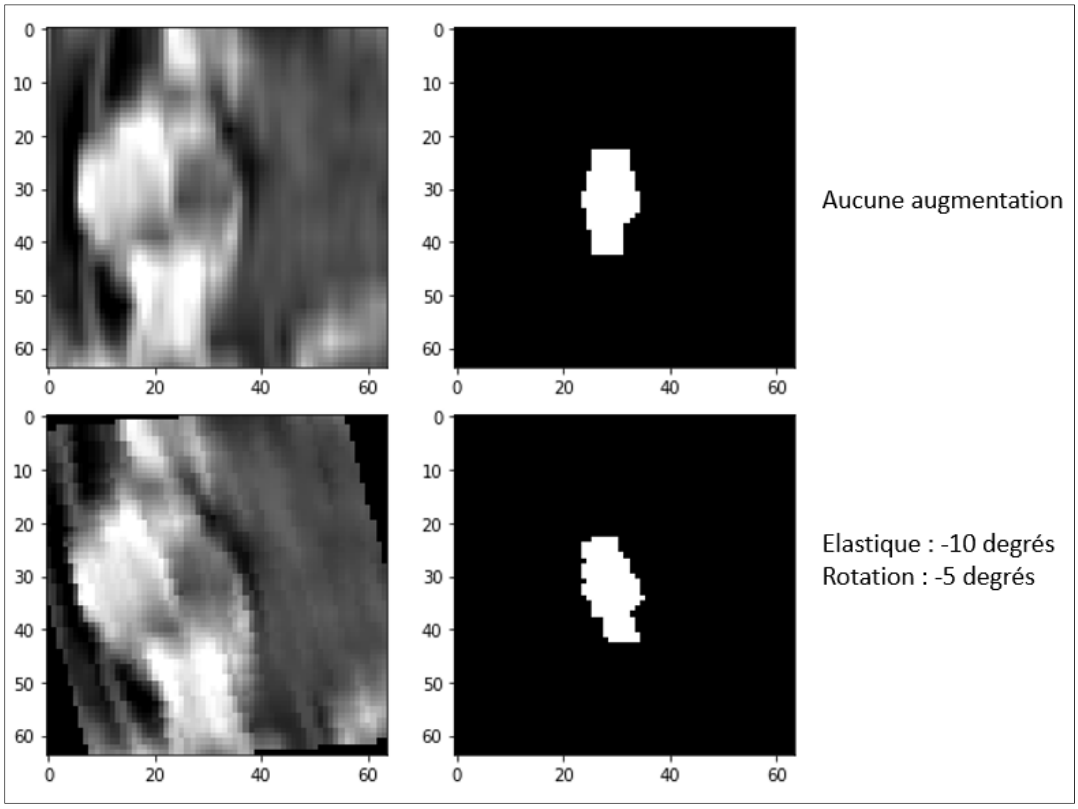



Figure 2.4 Illustration des transformées utilisées sur les tranches axiales

Les coupes originales sont aussi conservées dans les ensembles d'entrainement. De cette façon, la quantité de données pour l'entrainement est doublée. De plus, les coupes augmentées sont remplacées par des nouvelles versions augmentées à chaque époque, de façon à ce qu'il y ait toujours la moitié de l'ensemble de données d'entrainement qui corresponde aux coupes originales et l'autre moitié qui soit nouvellement augmentée. De cette façon, un apport constant de nouvelles données sont fournies au réseau, prévenant ainsi le *overfitting* et améliorant les résultats sur l'ensemble de validation.

### 2.1.2 Architecture du réseau convolutif

La même architecture est utilisée pour l'ensemble des approches exposées dans ce document. Suivant les considération exposées à l'ANNEXE V, cette architecture est de type U-Net avec

trois couches contractives/expansives et une couche *bottleneck*. La normalisation de batch n'a finalement pas été retenue, puisque le réseau performe mieux avec seulement des couches *dropout*. Des portes d'attention (voir l'ANNEXE V) ont été placées entre les couches expansives et contractives, tel qu'illustré à la Figure 2.5.

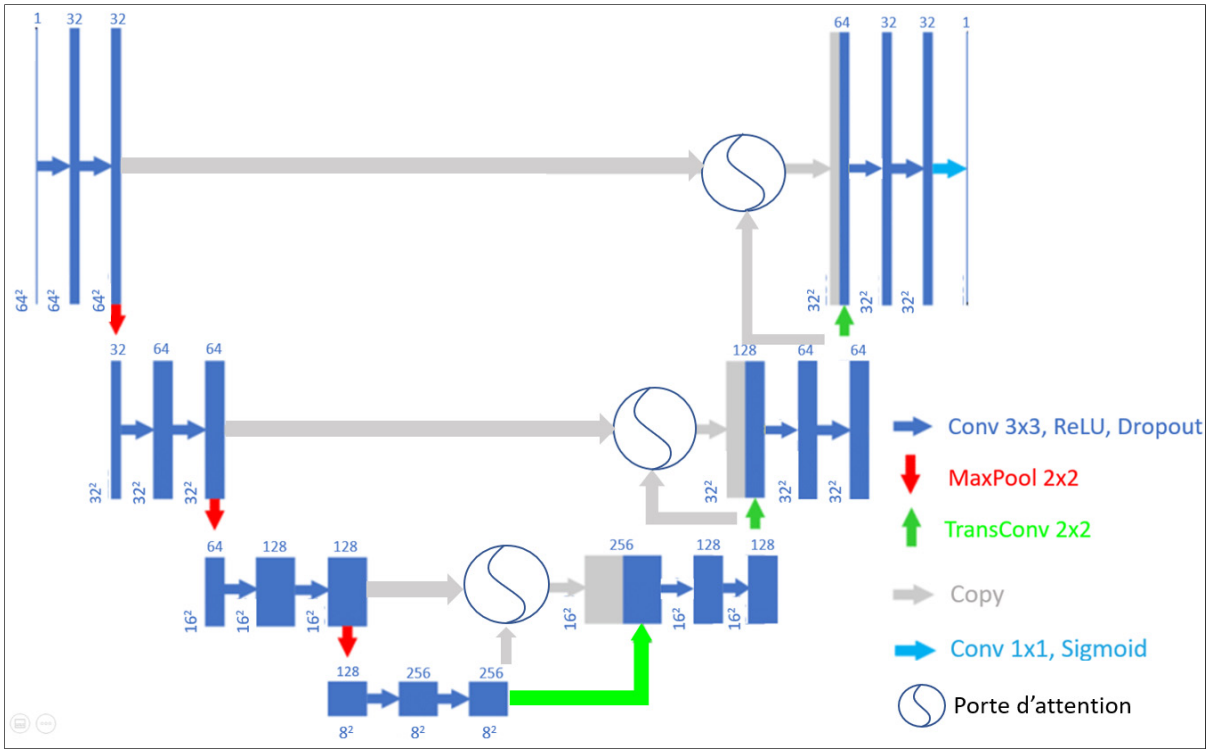



Figure 2.5 Schéma du réseau axial utilisé

La figure ci-haut illustre le réseau axial utilisé lors de l'approche en deux étapes. L'architecture demeure cependant la même pour l'étape de segmentation sagittale et l'approche semi-supervisée. Ces réseaux diffèrent seulement dans la taille des données aux différentes couches dans les réseaux, qui à ce moment est de $384^2$, $192^2$, $96^2$ et $48^2$ plutôt que de $64^2$, $32^2$, $16^2$ et $8^2$. De plus, dans le cas du réseau sagittal de l'approche en deux étapes, il y a deux canaux d'entrée plutôt qu'un. Ce sont donc les seules différences entre ces réseaux et le réseau affiché à la Figure 2.5.

### 2.1.3 Métriques pour la validation des résultats

La qualité de la relation entre un objet et sa segmentation est une entité complexe qui peut être difficile à évaluer de façon précise. C'est pourquoi plusieurs métriques sont habituellement utilisées afin de capturer les diverses facettes de cette relation et offrir une évaluation plus compréhensive. Dans le cadre de cette étude, quatre métriques communément utilisées ont été sélectionnées afin d'évaluer les segmentations produites. Ces métriques sont le coefficient Dice, le coefficient Jaccard, le taux de vrai positif et la distance Hausdorff (ANNEXE III).

#### 2.1.3.1 Mesure de similitude basée sur la ligne centrale

Cette métrique a été élaborée pour comparer la ligne centrale de la segmentation à la ligne centrale du masque de segmentation afin de donner une idée de la similarité des deux structures par rapport à la forme naturelle de la moelle épinière. La moelle épinière étant une structure allongée, et donc une sorte de « ligne » dans l'espace, cette métrique est particulièrement adaptée pour rendre une évaluation de la qualité des segmentations par rapport à la forme naturelle de la moelle épinière.

Pour former cette métrique, les centre de masse de chaque coupe axiale est d'abord calculé séparément pour la segmentation et le masque. À ce stade-ci, les coordonnées (*x, y, z*) de chaque centre de masse sont calculées, le symbole *z* correspondant ici à l'index de la tranche axiale dans la segmentation, comme on peut le voir sur la Figure 2.6. Si une tranche axiale ne possède pas de pixels segmentés, aucune donnée n'est enregistrée pour son index, cette tranche n'ayant à toute fin pratique pas de centre de masse.

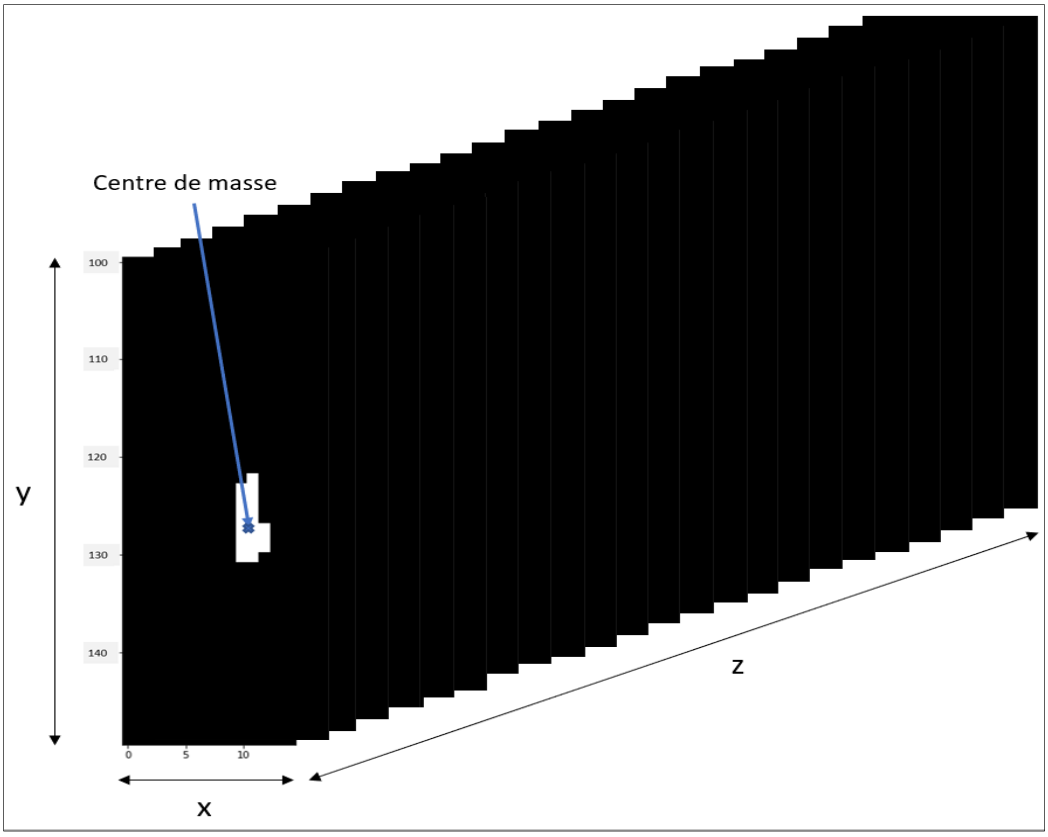



Figure 2.6 Centre de masse *(x, y)* d'une tranche *z*

L'étape suivante consiste à adapter une courbe dans un espace tridimensionnel afin qu'elle traverse le plus étroitement possible toutes les coordonnées des centres de masse qui ont été enregistrés. Ceci est nécessaire afin de pouvoir attribuer un centre de masse aux tranches vides mentionnées précédemment. Lors de ce processus, une courbe est adaptée pour la segmentation et une autre pour le masque.

Les coordonnées d'un point $p$ d'une courbe dans un espace tridimensionnel peuvent généralement être décrits de la façon suivante :

$$p(x, y, z) = (f(z), g(z), z) \tag{2.1}$$

Ici, *f(z)* et *g(z)* sont deux fonctions distinctes décrivant chaque point de la courbe tridimensionnelle dans les plan x et y, respectivement. Or, dans notre cas, la forme de la

colonne vertébrale peut être décrite dans les deux plans par deux équations de troisième degré de la forme :

$$f(x) = ax^3 + bx^2 + cx + d \tag{2.2}$$

Ceci a été trouvé empiriquement, en expérimentant avec des fonctions quadratiques de degrés plus élevées. Lors de ces expériences, les fonctions à quatre ou cinq degrés ne semblaient pas présenter des meilleurs résultats visuellement, et le module utilisé, *optimize* de *SciPy*, avait fréquemment de la difficulté à adapter les courbes. Z[2]

Suite à l'adaptation des deux courbes, la distance entre elles est calculée pour chaque point *z* à l'aide de la formule suivante :

$$d(z) = \sqrt{(f(z)_{seg} - f(z)_{mask})^2 + (g(z)_{seg} - g(z)_{mask})^2} \tag{2.3}$$

Ensuite, la moyenne des distances est retenue afin de produire le résultat final. La métrique peut donc être résumée sous la forme suivante :

$$LC = \frac{1}{z_f - z_0} \sum_{i=z0}^{zf} d(i) \tag{2.4}$$

Où $z_f$ et $z_0$ sont respectivement les index de la première et de la dernière tranche axiale du masque de segmentation possédant un centre de masse.

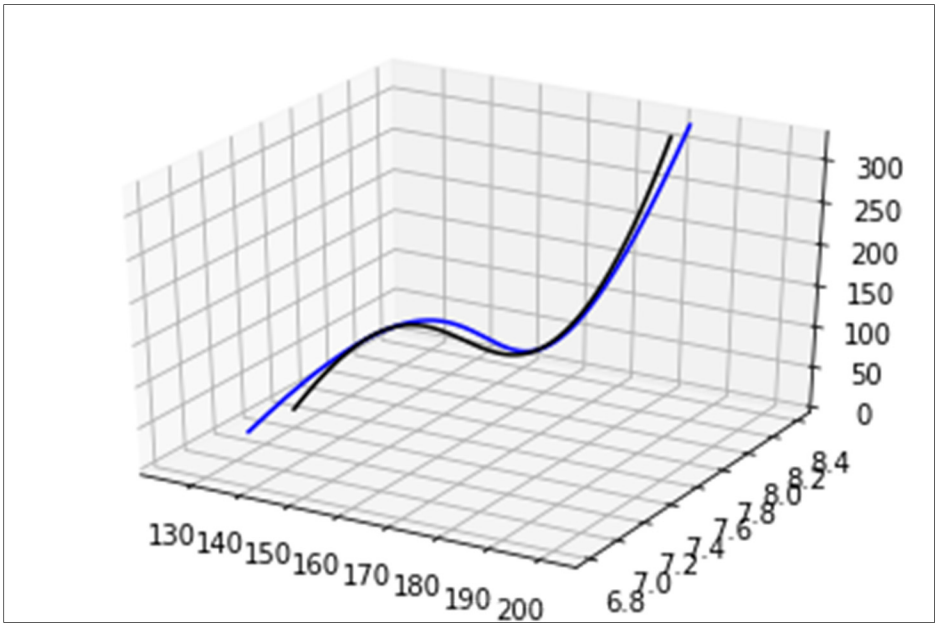



Figure 2.7 Visualisation de la ligne centrale d'une segmentation (noir) et du masque correspondant (bleu)

Le fait d'adapter les centres de masses à des lignes centrales continues permet d'extraire une information spécifique sur les lignes centrales en évitant de contaminer la métrique avec des pénalités non-voulues lorsque seulement une tranche de segmentation ou de masque contient un centre de masse, ce type de pénalité étant suffisamment représenté dans les autres métriques.

### 2.1.4 Méthode de validation croisée

L'entrainement se fait en suivant le processus de validation croisée K-Fold pour l'ensemble des approches décrites dans ce document. Lors de ce processus, la base de données réservée à l'entrainement est divisée aléatoirement en dix sous-ensembles de validation distincts. Chaque sous-ensemble de validation possède ses propres IRM qui ne sont partagées avec aucun autre sous-ensemble. Sur un total de 86 IRM réservées à l'entrainement, huit ou neuf d'entre elles sont donc allouées à chaque sous-ensemble de validation, tandis que les IRM restantes sont utilisées pour entrainer le réseau à proprement parler. Sur dix sous-ensembles, il y en a donc six qui ont neuf IRM de réservées à la validation et quatre qui en ont huit.

Il est à noter que la séparation des sous-ensembles se fait d'une telle façon que chaque IRM se retrouve dans un seul ensemble de validation. Il est aussi à noter que les dix sous-ensembles sont identiques pour l'ensemble des approches décrites dans ce document. Cette particularité est nécessaire afin de permettre la comparaison des sous-ensemble à travers les différentes approches.

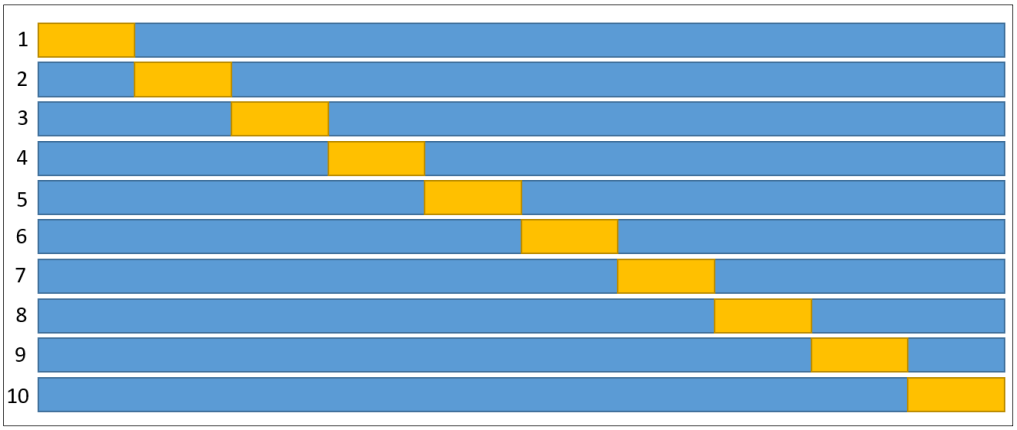



Figure 2.8 Schéma des dix sous-ensembles de validation (jaune). Les IRM sont mélangées aléatoirement précédant la séparation en sous-ensembles.

## 2.2 Approche en deux étapes

Lors de cette approche, le réseau axial segmente d'abord toutes les coupes axiales d'une IRM. Puis, ces coupes segmentées sont réorganisées pour former une segmentation tridimensionnelle de la moelle épinière. Suivant ceci, les coupes sagittales de l'IRM ainsi que celles de la segmentation sont utilisées comme données d'entrée pour l'entrainement du réseau sagittal.

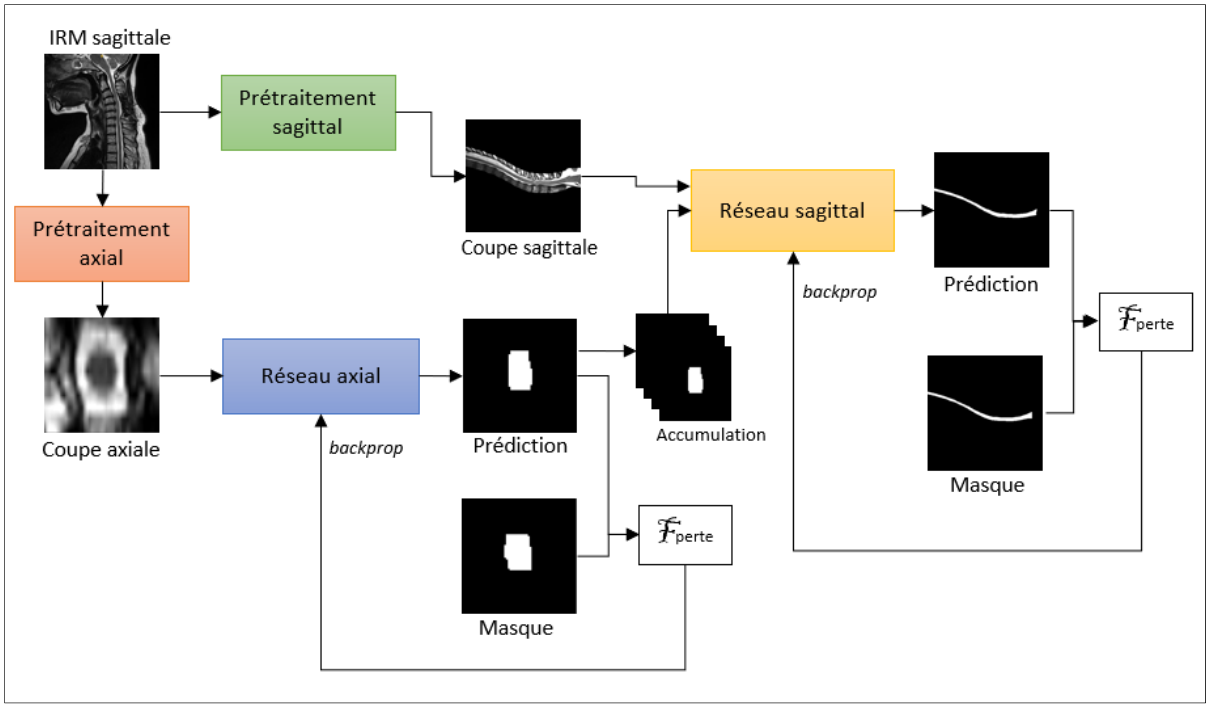



Figure 2.9 Schéma global de l'approche en deux étapes

Cette approche, qui vise la segmentation de la moelle épinière entière, est donc divisée en deux étapes. La première étape consiste à entrainer un réseau convolutif à partir de coupes axiales des afin de produire une première segmentation de la moelle épinière. Cette étape miroite l'approche de McCoy et al. vue précédemment, dans le sens où elle utilise uniquement des coupes axiales. Elle diffère cependant de cette approche dans la mesure où elle utilise seulement des IRM sagittales, tandis que cette approche utilise des IRM axiales ou un mélange des deux types. Rappelons ici que la différence entre une IRM axiale et sagittale est qu'une IRM axiale a une meilleure résolution dans le plan axial, tandis qu'une IRM sagittale a une meilleure résolution dans le plan sagittal.

La deuxième étape de cette approche prend comme données d'entrée les coupes sagittales des IRM ainsi que les segmentations obtenues lors de la première étape. Son objectif est d'entrainer un second réseau convolutif à partir de ces données, celui-ci pour segmenter les coupes sagittales. La combinaison des réseaux axiaux et sagittaux permet au système d'analyser les IRM sous différents angles, permettant ainsi de capturer davantage d'information contextuelle.

### 2.2.1 Étape 1: segmentation sur les coupes axiales

Cette approche consiste à entrainer le réseau de type U-Net avec portes d'attention décrit à la section 2.1.2 pour segmenter des coupes axiales dont le processus de prétraitement est décrit en détail à l'ANNEXE I. Le réseau prend donc en entrée des coupes axiales recadrées et normalisées de dimensions 64 × 64 et produit des segmentations de la même taille.

Les divers paramètres d'entrainement ainsi que l'architecture utilisés lors de ce processus ont été sélectionnés suivant les considérations décrites à l'ANNEXE V. Le Tableau 2.1 résume les paramètres spécifiques à l'entrainement.

Tableau 2.1 Résumé des hyperparamètres du réseau axial

| **Paramètre** | **Valeur** |
|---|---|
| Taille de batch | 4 |
| Fonction de perte | Dice |
| Taux de dropout | 0.4 |
| Optimisateur | Adam |
| Betas optimisateur | (0.5, 0.999) |
| Taux d'apprentissage | 5e-5 |
| Augmentation élastique (degrés) | 10 |
| Augmentation rotation (degrés) | 5 |

Le nombre total de coupes axiales contenues dans les 86 IRM d'entrainement est de 39 728. On obtient donc une séparation entrainement/validation moyenne de 35 755/3973 coupes pour les dix sous-ensembles de validation croisée. Les IRM n'ayant pas toutes les mêmes dimensions, et les sous-ensembles de validation n'ayant pas tous le même nombre d'IRM, la proportion des coupes axiales dédiées à la validation et à l'entrainement pour chaque sous-ensemble peut varier légèrement.

Durant l'entrainement, chaque coupe axiale est augmentée avec un coefficient élastique choisi aléatoirement entre -10 et 10 degrés et un coefficient de rotation entre -5 et 5 degrés. Les coupes non-modifiées sont aussi conservées dans l'ensemble d'entrainement, de façon à doubler la quantité de coupes axiales dédiées à l'entrainement. De plus, les coupes augmentées sont renouvelées à chaque époque de façon à maintenir un apport constant de nouvelles données au réseau. Ce processus est discuté plus en détail à la section 2.1.1.

Suivant l'augmentation, les sous-ensembles d'entrainement se retrouvent avec une moyenne de 71 510 coupes axiales. En tenant en compte la taille de batch de 4, cela résulte en une moyenne de 17 878 itérations par époque d'entrainement. En multipliant ce chiffre par 50 époques, qui est le nombre total d'époques d'entrainement (voir section 3.1.1), on obtient un total de 893 900 itérations de propagation arrière pour la durée entière de l'entrainement.

### 2.2.2 Étape 2 : segmentation sur les coupes sagittales

Suite à l'entrainement des dix réseaux axiaux à la section précédente, ceux-ci sont utilisés tour à tour pour segmenter les IRM d'entrainement et de validation. Les segmentations produites sont ensuite reconstruites sagittalement et jumelées avec leurs IRM originaux en vue d'être acheminées au réseau sagittal. Il est à noter que les mêmes séparations entrainement/validation qui ont été utilisées pour chaque itération de l'étape axiale sont maintenues. Le processus de reconstruction, rééchantillonnage et normalisation des images sagittales est décrit en détail à l'ANNEXE I.

Le réseau de segmentation sagittale utilise la même architecture que le réseau axial, seulement avec deux canaux d'entrée plutôt qu'un. Les divers paramètres utilisés lors du processus d'entrainement ont été sélectionnés suivant un processus d'optimisation basé sur le principe de sélection génétique. Le Tableau 2.2 résume les paramètres spécifiques à l'entrainement du réseau sagittal.

Tableau 2.2 Résumé des hyperparamètres du réseau sagittal

| **Paramètre** | **Valeur** |
|---|---|
| Taille de batch | 2 |
| Fonction de perte | Dice |
| Taux de dropout | 0.4 |
| Optimisateur | Adam |
| Betas optimisateur | (0.5, 0.999) |
| Taux d'apprentissage | 5e-5 |
| Augmentation élastique (degrés) | 50 |
| Augmentation rotation (degrés) | 15 |

Le nombre total de coupes sagittales contenues dans les 86 IRM est de 1177. Cependant, seules les coupes comportant des pixels segmentés sont conservées pour l'entrainement. Cette façon de procéder a été trouvée expérimentalement et vient grandement réduire le nombre de coupes sagittales utilisées. En effet, en éliminant les coupes ne contenant pas de moelle épinière, chaque IRM est réduite à quatre ou cinq coupes plutôt que de 13 ou 15. Cela permet au réseau de se concentrer uniquement sur les coupes pertinentes, évitant ainsi de gaspiller des ressources à tenter d'identifier les caractéristiques des tranches ne contenant pas de moelle épinière. Il est cependant à noter que suite à l'entrainement, lors de l'inférence sur les données de test, l'intégralité des coupes sont acheminées au réseau. Comme on le verra à la section suivante, les réseaux sagittaux possèdent donc la capacité de reconnaitre les coupes vides même si elles n'ont pas été utilisées lors de l'entrainement.

Suite à cette réduction des coupes, on obtient une séparation entrainement/validation moyenne de 348/39 coupes pour les dix sous-ensembles de validation croisée. Les IRM n'ayant pas toutes les mêmes dimensions, et les sous-ensembles de validation n'ayant pas tous le même nombre d'IRM, la proportion des coupes axiales dédiées à la validation et à l'entrainement pour chaque sous-ensemble peut varier légèrement (voir section 2.1.4).

Durant l'entrainement, chaque coupe sagittale est augmentée avec un coefficient élastique choisi aléatoirement entre -50 et 50 degrés, et en coefficient de rotation entre -15 et 15 degrés. Les coupes non-modifiées sont aussi conservées dans l'ensemble d'entrainement, de façon à doubler la quantité de coupes axiales dédiées à l'entrainement. Les coupes augmentées sont renouvelées à un intervalle de dix époques, faisant contraste avec les autres approches développées dans ce document, qui renouvellent les coupes augmentées à chaque époque. Cette façon de procéder a été trouvée expérimentalement. Le processus d'augmentation est discuté plus en détail à la section 2.1.1.

Suivant l'augmentation, les sous-ensembles d'entrainement se retrouvent avec une moyenne de 696 coupes axiales. En tenant en compte la taille de batch de deux, cela résulte en une moyenne de 348 itérations par époque d'entrainement. En multipliant ce chiffre par 125 époques, qui est le nombre total des époques d'entrainement (voir section 3.1.2), on obtient un total de 43 500 itérations de propagation arrière.

Suite à l'entrainement, lorsque les réseaux sont utilisés en inférence pour segmenter les IRM de test, un post-traitement est appliqué aux segmentations produites. Ce post-traitement consiste à effacer tous les pixels segmentés sur une coupe sagittale si elle contient moins que 20 pixels de segmentés. Ceci est un nombre très limité de pixels, puisqu'une coupe sagittale contient plus de 145 000 pixels. L'impact de ce post-traitement, qui vient en quelque sorte vider les tranches vides, n'est donc pas habituellement visible à l'œil nu. Cependant, vu la façon dont certaines métriques sont calculées, il peut avoir un impact significatif sur elles. Il est à noter pour les comparaisons avec Deepseg à la section suivante que celui-ci utilise un post-traitement plus étendu et complexe que le nôtre, tel que discuté à la section 1.6.

### 2.2.3 Méthode d'analyse statistique des résultats

#### 2.2.3.1 Test-t

L'un des objectifs principaux lors de l'analyse des résultats est d'établir une comparaison significative entre les différents réseaux entrainés et aussi entre ceux-ci et un réseau

représentant l'état de l'art. Il est donc nécessaire d'établir un processus pour la comparaison de deux réseaux. Ce processus aura comme objectif l'obtention d'une valeur-p exprimant la probabilité que les deux réseaux aient la même erreur moyenne par rapport à leur population respective composée de leurs résultats sur une métrique. Dans ce calcul, chaque population est représentée par un échantillon composé des 20 résultats obtenus lorsque la métrique est appliquée aux segmentations produites par le réseau sur les 20 IRM de test. Ce processus est visible sur la Figure 2.10.

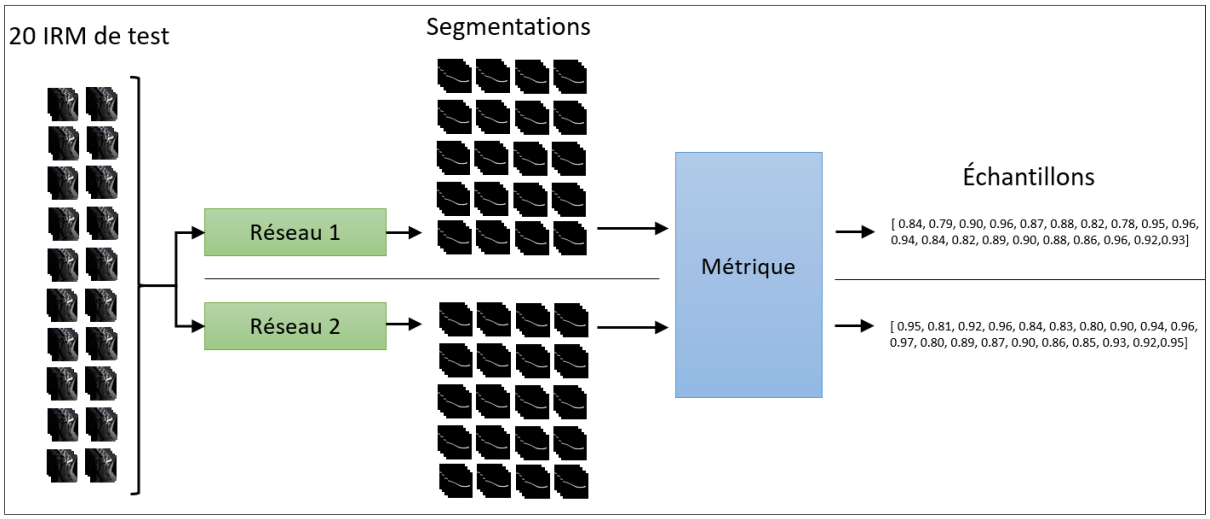



Figure 2.10 Créations des échantillons

Une fois les échantillons obtenus, l'étape suivante est de produire la matrice erreur d'un réseau par rapport à l'autre. Ceci se fait tout simplement en soustrayant une matrice d'échantillons à l'autre. On se retrouve alors avec une seule matrice erreur de longueur 20.

L'étape suivante est de trouver la valeur-t de la matrice erreur. Celle-ci peut se calculer avec la formule suivante :

$$t = \frac{\bar{e}\sqrt{n}}{\sqrt{\frac{\sum_{i=0}^{n}(\bar{e}-e_i)^2}{n-1}}} \tag{2.5}$$

Où $\bar{e}$ est la moyenne de la matrice erreur et $n$ est longueur de la matrice erreur. On peut voir que le dénominateur de cette formule représente l'écart-type échantillon de la matrice erreur.

À partir de la valeur-t, la valeur-p peut être trouvée facilement à l'aide de la librairie *SciPy*, en s'assurant de spécifier un nombre de degrés de liberté de 20 – 1 = 19 et une hypothèse à deux queues, voulant dire qu'on retient comme possibilité qu'un réseau ou l'autre ait une erreur moyenne plus grande ou plus petite que l'autre.

L'hypothèse que l'on a fixé au départ, en choisissant de soustraire les deux matrices échantillons pour obtenir la matrice erreur, est que les deux erreurs moyennes des deux populations auxquelles appartiennent ces échantillons sont égales. Donc, en fixant le seuil de signification à $\alpha = 0.05$, on peut rejeter cette hypothèse si $p < 0.05$ et affirmer que les erreurs moyennes des populations ne sont pas égales, et donc que la différence entre les deux moyennes obtenues est significative.

#### 2.2.3.2 Limitations

La méthode de test-t décrite à la section précédente comporte certains aspects dont il est important de tenir compte. En effet, cette méthode perd en grande partie sa validité lorsque les données utilisées pour entrainer les réseaux à comparer sont les mêmes ou bien comportent des éléments partagés (Dietterich et al., 1998; Salzberg et al., 1997). En particulier, ceci cause un risque élevé d'erreur de type I, c'est-à-dire un risque que le test détecte une différence significative dans la performance des réseaux lorsqu'il n'y en a pas. Les résultats de ces tests ne doivent alors être interprétés que comme des heuristiques et non que comme des résultats statistiquement vérifiés (Dietterich et al., 1998).

L'emploi de la validation croisée K-Fold peut être utilisé pour atténuer les effets négatifs du partage de données, en augmentant le nombre d'essais et en assurant l'indépendance des données de test entre les différents essais. Cependant, les essais conservent tout de même une grande proportion de données partagées, et le risque d'erreur de type demeure significatif

(Dietterich et al., 1998; Salzberg et al., 1997). Le risque d'erreur de type II (que le test ne détecte pas de différence lorsqu'il y en a une) devient cependant acceptable (Dietterich et al., 1998).

Il est important de noter que dans le cadre de cette étude, lorsque des test-t sont utilisés pour comparer un réseau au réseau Deepseg, ce type d'erreur ne devrait pas être en cause, puisque les données utilisées pour l'entrainement des deux réseaux sont complètement indépendantes, et qu'il n'y a pas de chevauchement entre les données de test et d'entrainement. Les résultats de ces tests-t devraient donc être statistiquement valides, pour autant que la distribution des échantillons soit normale, tel que discuté à la section plus bas.

#### 2.2.3.3 Implémentation sur les résultats

Lors des comparaisons avec le réseau Deepseg, chacun des réseaux résultant des dix itérations croisées est comparé indépendamment à Deepseg à l'aide d'un test-t. Cette façon de procéder est plus valable que l'alternative, qui comporterait la combinaison des dix résultats préalablement au test-t, car dans ce cas, le test-t serait contaminé par des données d'entrainement chevauchées, tel que discuté à la section précédente.

Donc, suite à ces tests-t, dix valeurs-p sont obtenues. Prises indépendamment les unes des autres, ces valeurs sont statistiquement valides. Cependant, vu le fait que les données se chevauchent entre les essais, leur combinaison comporte la même problématique que les test-t. Il importera donc, lors de la présentation des résultats, de présenter les dix résultats séparément ainsi de leur moyenne. Bien que la moyenne ne soit pas strictement valide comme méthode de combinaison des dix résultats, elle représente néanmoins une alternative plus pessimiste que la méthode de combinaison Fisher, qui utilise une approche multiplicative, réduisant la valeur-p obtenue à mesure que le nombre d'échantillons augmente et augmentant donc la probabilité d'erreur de type I.

Certaines méthodes existent pour tenter de combiner des valeurs-p dépendantes (Kost et McDermott, 2002; Poole et al., 2016; Wilson D., 2019; Vovk V., 2012), cependant, il ne semble pas y avoir de consensus clair en regard de leur validité. De plus, les données doivent répondre à plusieurs conditions différentes dépendant de la méthode utilisée. Tenter d'implanter une de ces méthodes à partir d'articles scientifiques est donc un processus qui risquerait fortement d'être en proie à des erreurs. Voilà pourquoi la simple moyenne a été retenue à titre d'heuristique.

#### 2.2.3.4 Analyse de la distribution des résultats

En plus des considérations exposées précédemment, il est nécessaire de vérifier que les données suivent une distribution normale ou gaussienne afin de produire une analyse statistique valide. Il existe plusieurs types de tests conçus pour évaluer la normalité d'une population. Dans le cadre de cette étude, c'est le test d'Agostino-Pearson qui est utilisé. Ce test compare des caractéristiques concernant le *kurtosis* et l'asymétrie d'un échantillon à celles d'une distribution gaussienne, puis utilise ces mesures afin de produire une valeur-p indiquant la probabilité que l'échantillon provienne d'une distribution gaussienne. Un échantillon ayant une valeur-p $> 0.05$ est accepté comme ayant une distribution normale. Ce test requiert un échantillon de taille minimale $n = 8$; notre taille d'échantillon de 20 IRM est donc suffisamment élevée. Ce test est effectué en utilisant la librairie *SciPy*.

Une autre façon d'évaluer une distribution est en visualisant l'estimation par noyau (kernel density estimation) de l'échantillon. L'estimation par noyau est une méthode non-paramétrique qui permet d'estimer la densité en tout point d'une population à l'aide d'un échantillon. Le fait que cette méthode soit non-paramétrique indique qu'elle est indépendante de la distribution de la population, ce qui la rend bien adaptée pour visualiser des populations dont la distribution est incertaine.

## 2.3 Approche semi-supervisée

L'apprentissage semi-supervisé est une technique qui tente de combiner l'apprentissage supervisé et l'apprentissage non-supervisé. Ce processus utilise donc habituellement une combinaison de données annotées et non-annotées. De nombreuses méthodes existent pour combiner les deux types d'apprentissage, plusieurs d'entre elles étant liées des domaines d'application spécifiques (Qi et Luo, 2022).

### 2.3.1 Apprentissage pseudo-supervisé en croisé

La méthode utilisée dans ce chapitre consiste à entrainer deux réseaux parallèlement en vue de tirer avantage du partage des segmentations entre eux. Elle est en partie adaptée de la méthode développée dans (Chen et al., 2021), que les auteurs appellent la méthode *pseudo-supervisée en croisé*.

Dans cette méthode, deux réseaux identiques sont d'abord initialisés avec des poids aléatoires différents. Puis, ils sont entrainés en parallèle avec les mêmes données d'entrée et les mêmes hyperparamètres. Le volet d'apprentissage supervisé utilise la fonction de perte d'entropie croisée par pixel (pixel-wise cross entropy) entre la segmentation et le masque, tandis que le volet d'apprentissage semi-supervisé utilise aussi cette fonction de perte, mais puisqu'aucun masque n'existe, la segmentation est comparée à une version encodée *one-hot* de la segmentation de l'autre réseau. L'encodage one-hot est une méthode de binarisation d'image où chaque classe dans une segmentation se trouve à être représentée par une suite binaire particulière. La perte semi-supervisée ainsi obtenue est ensuite multipliée par un coefficient lambda et puis additionnée à la perte supervisée pour former la perte totale qui est propagée vers l'arrière dans le réseau.

Notre approche conserve le principe général de cette approche, c'est-à-dire d'entrainer deux réseaux en parallèle et d'utiliser une version transformée de la segmentation de l'un pour entrainer l'autre. Elle diffère cependant sur plusieurs points. D'abord, l'approche originale utilise une architecture Deeplabv3+ avec *backbone* Resnet, tandis que notre méthode utilise

plutôt une architecture de type U-Net avec portes d'attention (voir section 2.1.2). De plus, l'approche originale analyse des photographies provenant de bases de données bien connues telles que PASCAL VOC et Cityscapes, tandis que notre approche analyse des IRM inédits. Cela est significatif, car les images médicales possèdent des caractéristiques bien différentes des photographies ordinaires.

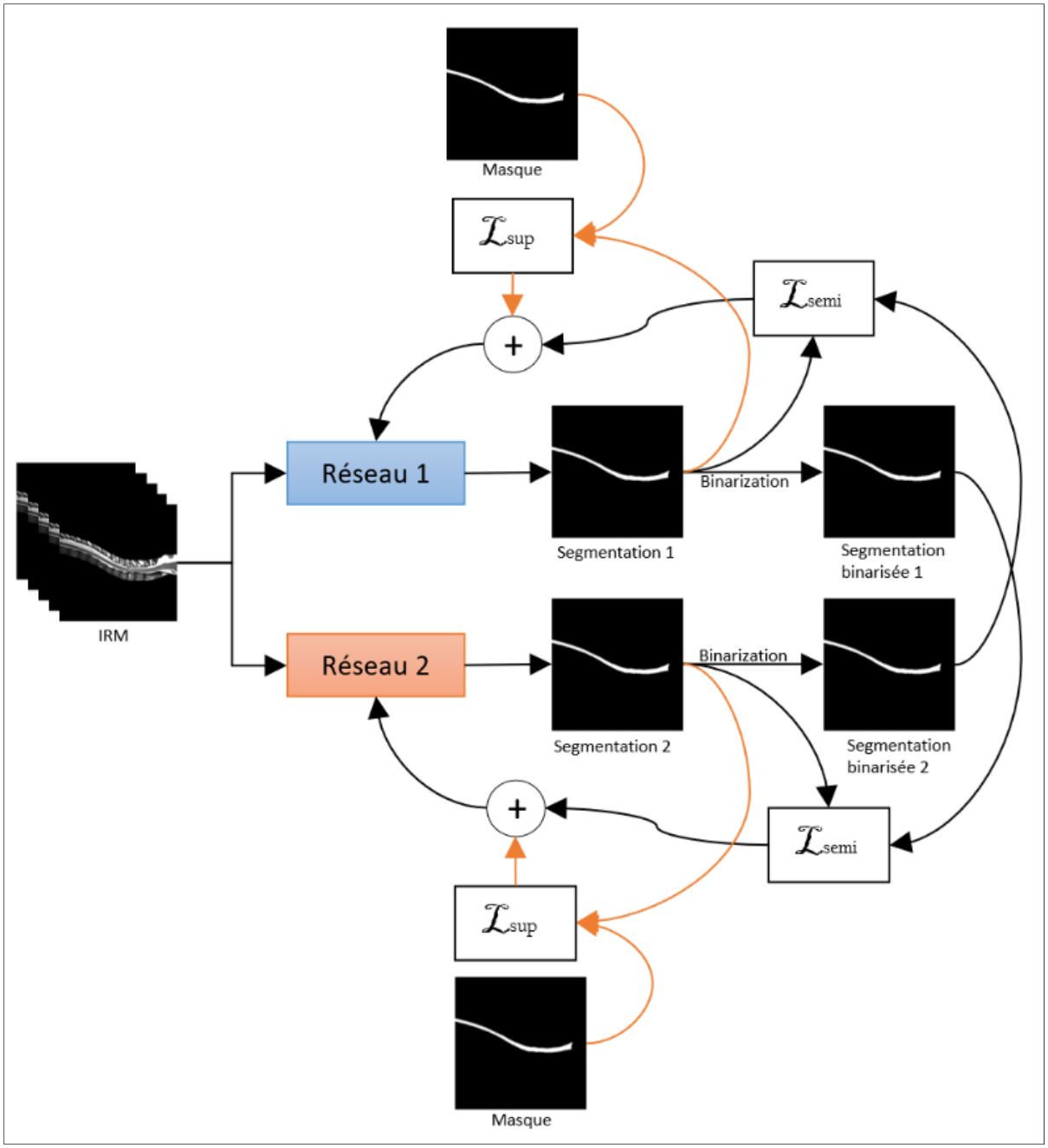



Figure 2.11 Schéma du système d'apprentissage semi-supervisé

La fonction de perte utilisée pour l'entrainement supervisé dans notre approche n'est pas l'entropie croisée comme dans l'approche originale, mais plutôt la fonction de perte Dice. De plus, puisque nos segmentations ont une seule classe, le processus d'encodage *one-hot* de l'étude originale est remplacé par une binarisation des segmentations. Le seuil de signification

pour ces binarisations a été trouvé expérimentalement avec une technique basée sur la sélection génétique et est fixé à 0.1.

### 2.3.2 Prétraitement des données

Les réseaux sont entrainés avec des images sagittales de dimension 384 × 384 ayant été soumises aux prétraitements décrits à l'ANNEXE I. Ces prétraitements incluent le recadrage autour de la ligne centrale, le rééchantillonnage et la normalisation des images. Les réseaux sont entrainés avec des images sagittales plutôt qu'axiales puisque les IRM manuellement annotés sont de type sagittal.

Tandis que l'approche originale d'entrainement semi-supervisé en croisé utilise l'augmentation CutMix (Yun et al., 2019), notre approche utilise des rotations et des transformations élastiques (voir section 2.1.1). Ainsi, chaque coupe sagittale est augmentée avec un coefficient élastique choisi aléatoirement entre -50 et 50 degrés et en coefficient de rotation entre -15 et 15 degrés. Les coupes non-modifiées sont aussi conservées dans l'ensemble d'entrainement, de façon à doubler la quantité de coupes dédiées à l'entrainement. De plus, les coupes augmentées sont renouvelées à chaque époque, de façon à maintenir un apport constant de nouvelles données au réseau.

### 2.3.3 Méthode de validation croisée

Les données sont séparées conformément à l'approche de validation K-Fold décrite à la section 2.1.4. Elles sont donc séparées en dix sous-ensembles d'entrainement/validation où chaque ensemble de validation ne contient aucun élément en commun avec les autres sous-ensembles de validation. Les ensembles d'entrainement partagent cependant la plus grande partie de leurs données. Or, puisque lors de l'entrainement semi-supervisé seulement un certain pourcentage des données est annoté, il est important que les différentes itérations de validation croisée n'aient pas toutes les mêmes données d'annotées. Les données qui sont annotées pour une itération donnée sont donc choisies aléatoirement, tel qu'illustré sur la Figure 2.12.

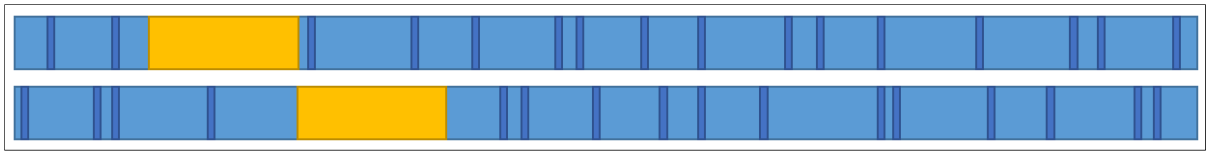


Figure 2.12 Visualisation des données pour deux des dix itérations de validation croisée. Le jaune représente les données de validation, le bleu foncé les données annotées et le bleu pâle les données non-annotées. La base de données est mélangée aléatoirement précédant ces séparations.

### 2.3.4 Pertes non-supervisées

La fonction de perte pour l'entrainement semi-supervisé est l'entropie croisée binaire par pixel (BCE). Cette fonction de perte compare chaque pixel de la segmentation au pixel du masque correspondant et retourne le négatif du logarithme de 1 moins la différence entre les valeurs des pixels. Pour un masque $m$ et une segmentation $s$ de longueur $L$ et de largeur $H$, elle peut être formulée ainsi :

$$\mathcal{L}_{bce} = \frac{1}{L \times H} \sum_{i=0}^{L \times H} -(m_i \cdot \log(s_i) + (1 - m_i) \cdot \log(1 - s_i)) \tag{2.7}$$

Durant l'entrainement, les pertes non-supervisées pour une batch donnée sont multipliées par un coefficient lambda et additionnées aux pertes supervisées Dice. Cependant, puisque les données sont augmentées et mélangées aléatoirement à chaque époque, les batchs contiennent un mélange de coupes annotées et non-annotées. Les coupes annotées sont donc utilisées pour calculer la perte Dice, tandis que toutes les coupes non-annotées sont utilisées dans le calcul de la perte non-supervisée. Cette façon de fonctionner implique que le nombre de coupes annotées dans une batch est variable, ce qui fait en sorte que le calcul de la perte Dice pour une batch est basé sur un nombre variable de coupes.

$$\mathcal{L} = \mathcal{L}_{dice} + \lambda \mathcal{L}_{bce} \tag{2.8}$$

Cette façon de faire, qui a été trouvée expérimentalement, aide à balancer la perte totale en assurant que la perte Dice ait un apport constant peu importe le nombre de coupes annotées dans une batch. Le coefficient lambda pour la perte non-supervisée est fixé à 0.05, une valeur qui a aussi été trouvée expérimentalement.

### 2.3.5 Résumé des paramètres

Les hyperparamètres utilisés lors du processus d'entrainement ont été sélectionnés suivant les considérations décrites à la section 2.3 ainsi qu'en utilisant un processus d'optimisation basé sur la sélection génétique. Le Tableau 2.3 résume les paramètres spécifiques à l'entrainement des réseaux.

Tableau 2.3 Résume des hyperparamètres

| Paramètre | Valeur |
|---|---|
| Taille de batch | 8 |
| Fonction de perte supervisée | Dice |
| Fonction de perte non-supervisée | BCE |
| Lambda de perte non-supervisée | 0.05 |
| Taux de dropout | 0.3 |
| Optimisateur | Adam |
| Betas optimisateur | (0.5, 0.999) |
| Taux d'apprentissage | 5e-5 |
| Augmentation élastique (degrés) | 50 |
| Augmentation rotation (degrés) | 15 |
| Seuil binarisation | 0.1 |

### 2.3.6 Procédé d'entrainement

Préalablement à l'entrainement semi-supervisé, les réseaux sont entrainés de façon supervisée pendant 40 époques. Ceci est effectué afin que les réseaux produisent des segmentations qui

ont un certain sens avant de démarrer l'apprentissage non-supervisé, ce qui est nécessaire puisque, lors de celui-ci, chacun des réseaux se base en partie sur les segmentations de l'autre réseau pour ajuster ses paramètres. Suite à la quarantième époque, une version de chaque réseau continue d'être entrainée de façon supervisée, tandis qu'une autre version démarre l'entrainement semi-supervisée. La version sans semi-supervision servira de contrôle pour mesurer l'amélioration produite par l'entrainement semi-supervisé lors de l'analyse des résultats.

# CHAPITRE 3

# RÉSULTATS ET DISCUSSION

## 3.1 Approche en deux étapes

### 3.1.1 Étape 1 : segmentation sur les coupes axiales

Les résultats contenus dans cette section servent principalement à comparer la présente étape axiale avec le réseau Deepseg. Cette comparaison est intéressante puisque ce dernier utilise aussi un réseau axial seul pour segmenter la moelle épinière. De plus, ce réseau est aussi de type U-Net, avec une configuration similaire à notre réseau (voir sections 1.6 et 2.1.2). Il est cependant à noter que ces résultats ne sont pas les résultats finaux de l'approche en deux étapes.

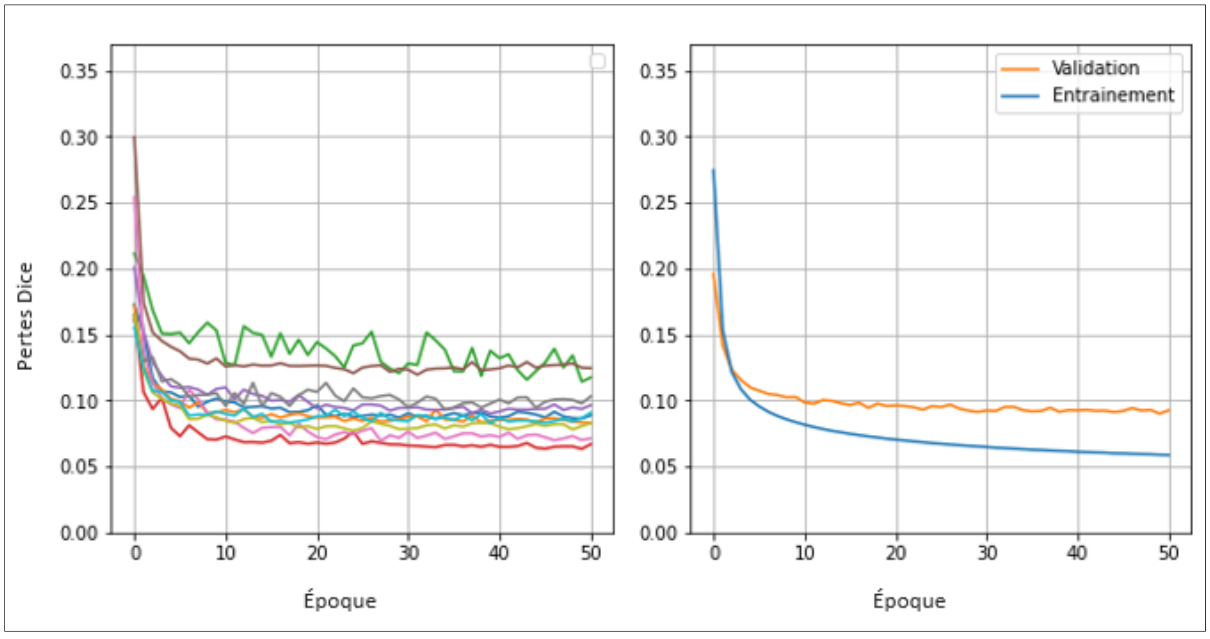



Figure 3.1 Courbes de pertes de validation pour chacun des 10 sous-ensembles de validation croisée (gauche) et moyenne des pertes de validation et d'entrainement pour tous les sous-ensembles (droite). La moyenne des pertes de validation atteint sa valeur minimale à l'époque 49

Pour déterminer la période optimale d'entrainement, les dix réseaux de validation croisée sont entrainés pour une durée 50 époques. Ce nombre d'époques correspond à un nombre d'époques suffisamment élevé pour que les courbes des pertes de validation des sous-ensembles s'aplanissent ou bien commencent à remonter. Ceci correspond généralement au moment où les réseaux commencent à faire du *overfitting*, surtout lorsque les pertes d'entrainement continuent de baisser. Ce phénomène d'aplanissement est bien visible sur la Figure 3.1, qui montre la moyenne des pertes de validation pour la totalité des sous-ensembles.

La Figure 3.2 illustre la distribution des cinq métriques pour le réseau Deepseg. Chaque courbe est composée de 20 points correspondants aux 20 IRM de test. On voit au coup d'œil que les distributions s'apparentent à des distributions normales. De plus, les valeurs-p Agostino-Pearson obtenues pour l'ensemble des métriques sont inférieures à 0.05 (voir section 2.2.3.4). On peut donc conclure que les distributions des cinq métriques pour le réseau Deepseg sont normales. Ceci est fortuit, car ce réseau servira comme référence pour comparer les réseaux construits dans l'approche en deux étapes.

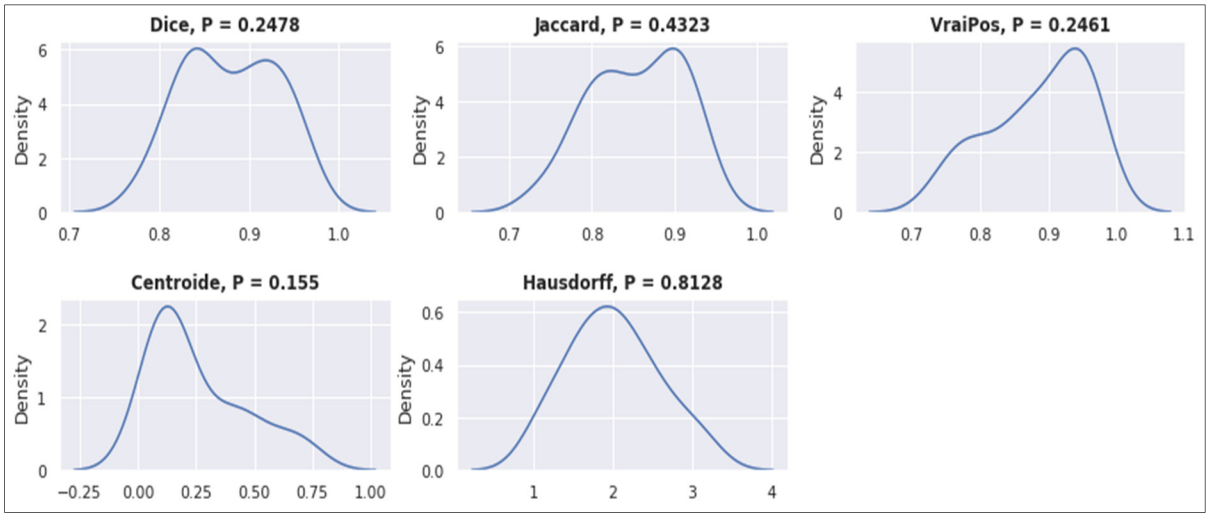



Figure 3.2 Estimations par noyau et valeurs-p des tests Agostino-Pearson pour les cinq métriques pour le réseau Deepseg

**Coefficient Dice**

Le Tableau 3.1 montre les valeurs Dice pour l'époque finale des dix itérations de validation croisée. Il est à noter que ces valeurs représentent les valeurs Dice moyennes des tranches axiales et non des tranches sagittales, car les réseaux sont entrainés avec des tranches axiales. De plus, ces valeurs sont calculées durant l'entrainement à partir de tranches axiales recadrées à 64 × 64. La comparaison entre les valeurs dans ce tableau et celles dans le Tableau 3.2 montrant les valeurs sagittales sur les IRM de test ne peut donc qu'être qualificative.

Tableau 3.1 Pertes et coefficients de validation Dice axiales à l'époque 50

| Indice k | Pertes Dice | Coefficients Dice |
|---|---|---|
| 1 | 0.089 | 91.1 |
| 2 | 0.083 | 91.7 |
| 3 | 0.118 | 88.3 |
| 4 | 0.067 | 93.3 |
| 5 | 0.096 | 90.4 |
| 6 | 0.125 | 87.6 |
| 7 | 0.071 | 92.9 |
| 8 | 0.103 | 89.7 |
| 9 | 0.083 | 91.7 |
| 10 | 0.091 | 90.9 |
| **Moyenne** | 0.092 | 90.8 |
| Écart-type | 0.019 | 1.90 |

Ceci étant dit, les coefficients Dice axiaux sont nettement supérieurs aux valeurs sagittales du Tableau 3.2. Ceci peut s'expliquer par le fait que les réseaux axiaux sont entrainés de façon à minimiser les pertes axiales et non sagittales. Les pertes sagittales ne sont donc minimisées qu'indirectement. De plus, l'ensemble des IRM sont de type sagittal, donc leur résolution d'origine dans le plan axial est pauvre, et bien qu'elles soient rééchantillonnées préalablement à l'entrainement des réseaux, les caractéristiques qu'elles contiennent demeurent forcément plus floues que celles sur le plan sagittal, ce qui simplifie grandement le travail de segmentation.

Tableau 3.2 Résultats Dice sur les 20 IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Coefficient Dice** | 1 | 83.4 | 0.248 | 2.658 | 0.016 |
| | 2 | 83.1 | 0.101 | 2.923 | 0.009 |
| | 3 | 83.3 | 0.021 | 2.451 | 0.024 |
| | 4 | 83.0 | 0.789 | 2.617 | 0.017 |
| | 5 | 82.9 | 0.213 | 3.072 | 0.006 |
| | 6 | 82.8 | 0.322 | 2.840 | 0.010 |
| | 7 | 82.7 | 0.144 | 2.844 | 0.010 |
| | 8 | 82.3 | 0.928 | 2.939 | 0.008 |
| | 9 | 82.4 | 0.302 | 2.969 | 0.008 |
| | 10 | 83.3 | 0.121 | 2.467 | 0.023 |
| | **Moyenne** | 82.9 ***deepseg = 87.9*** | 0.319 | 2.778 | 0.012 |
| | Écart type | 0.4 | 0.301 | 0.217 | 0.006 |

Le réseau Deepseg est nettement supérieur à nos réseaux sur le plan sagittal. En effet, on peut voir dans le Tableau 3.2 que Deepseg a un coefficient Dice moyen de 87.9 sur les 20 IRM de test, tandis que nos réseaux ont un coefficient moyen de 82.9. En comparant nos réseaux avec Deepseg selon le processus décrit à la section 2.2.3, l'on obtient des valeurs-p $< 0.05$ pour l'ensemble des dix itérations, indiquant que le réseau Deepseg est significativement supérieur à l'ensemble de nos réseaux.

Seulement un de nos dix réseaux possède une distribution qui n'est pas classifiée comme normale par le test d'Agostino-Pearson (en rouge dans le Tableau 3.2). La valeur-p qui est associée à ce réseau ne peut donc pas être considérée comme valide. Cependant, cela importe peu, puisque le réseau Deepseg est aussi clairement supérieur.

**Coefficient Jaccard**

Le coefficient Jaccard s'apparente au coefficient Dice, tel que discuté à l'ANNEXE III. Il est présenté afin que les deux coefficients, pris ensemble, se renforcent et se confirment. On s'attend donc ici à ce que les résultats de Deepseg soient encore supérieurs, ce qui est le cas.

Tableau 3.3 Résultats Jaccard sur les 20 IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Coefficient Jaccard** | 1 | 80.5 | 0.297 | 2.850 | 0.010 |
| | 2 | 80.2 | 0.152 | 3.160 | 0.005 |
| | 3 | 80.4 | 0.059 | 2.648 | 0.016 |
| | 4 | 80.1 | 0.870 | 2.812 | 0.011 |
| | 5 | 79.9 | 0.449 | 3.315 | 0.004 |
| | 6 | 79.9 | 0.358 | 3.002 | 0.007 |
| | 7 | 79.8 | 0.265 | 3.063 | 0.006 |
| | 8 | 79.3 | 0.851 | 3.157 | 0.005 |
| | 9 | 79.5 | 0.462 | 3.136 | 0.005 |
| | 10 | 80.4 | 0.078 | 2.658 | 0.016 |
| | **Moyenne** | 80.0 ***deepseg = 85.4*** | 0.384 | 2.980 | 0.009 |
| | Écart-type | 0.4 | 0.287 | 0.228 | 0.004 |

On peut noter sur le Tableau 3.3 que toutes les distributions sont normales, donc toutes les valeurs-p sont valides. Puisque l'ensemble des valeurs-p est inférieur à 0.05, on peut conclure avec confiance que Deepseg en supérieur à nos réseaux.

**Taux de vrais positifs**

Tel que discuté à l'ANNEXE III, le taux de vrais positifs est une métrique statistique indiquant la proportion des pixels segmentés par le réseau qui sont aussi segmentés sur le masque. Comme on peut le voir sur le Tableau 3.4, le réseau Deepseg est hautement supérieur à nos réseaux en regard de cette métrique, avec un taux moyen de 0.89, tandis que nos réseaux ont un taux moyen de 0.57. Cette différence est mise en évidence par les valeurs-p, qui sont dans l'ordre de $1x10^{-14}$, ne laissant aucun doute quant à supériorité de Deepseg.

Tableau 3.4 Taux de vrais positifs sur les 20 IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Taux vrai positif** | 1 | 0.56 | 0.926 | 21.607 | 7.78E-15 |
| | 2 | 0.57 | 0.868 | 19.428 | 5.39E-14 |
| | 3 | 0.57 | 0.775 | 19.420 | 5.43E-14 |
| | 4 | 0.57 | 0.754 | 19.851 | 3.65E-14 |
| | 5 | 0.58 | 0.627 | 19.124 | 7.18E-14 |
| | 6 | 0.57 | 0.416 | 20.169 | 2.73E-14 |
| | 7 | 0.57 | 0.452 | 20.624 | 1.82E-14 |
| | 8 | 0.57 | 0.769 | 20.912 | 1.41E-14 |
| | 9 | 0.57 | 0.765 | 19.905 | 3.47E-14 |
| | 10 | 0.58 | 0.465 | 18.336 | 1.53E-13 |
| | **Moyenne** | 0.57 ***deepseg = 0.89*** | 0.682 | 19.938 | 4.72E-14 |
| | Écart-type | 0.01 | 0.181 | 0.944 | 4.24E-14 |

**Ligne centrale**

Tel que décrit à la section 2.1.3.1, la métrique de la ligne centrale est une métrique élaborée au cours de la présente étude afin de donner une mesure adaptée à la forme naturelle de la moelle épinière.

Tableau 3.5 Résultats pour la métrique de la ligne centrale pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Ligne centrale** | 1 | 0.60 | 0.622 | -4.001 | 0.0008 |
| | 2 | 0.61 | 0.614 | -4.200 | 0.0005 |
| | 3 | 0.64 | 0.000 | -4.037 | 0.0007 |
| | 4 | 0.63 | 0.171 | -4.740 | 0.0001 |
| | 5 | 0.60 | 0.000 | -4.176 | 0.0005 |
| | 6 | 0.59 | 0.022 | -4.483 | 0.0003 |
| | 7 | 0.59 | 0.617 | -4.387 | 0.0003 |
| | 8 | 0.60 | 0.175 | -3.898 | 0.0010 |
| | 9 | 0.58 | 0.000 | -4.521 | 0.0002 |
| | 10 | 0.62 | 0.049 | -4.018 | 0.0007 |
| | **Moyenne** | 0.61 ***deepseg = 0.30*** | 0.227 | -4.246 | 0.0005 |
| | Écart-type | 0.02 | 0.277 | 0.275 | 0.0003 |

Comme on peut le voir sur le Tableau 3.5, Deepseg semble encore une fois être supérieur à nos réseaux, avec une distance moyenne de 0.30 tandis que nos réseaux affichent une distance moyenne de 0.61, soit 100% de plus que Deepseg. Les valeurs-p doivent cependant être interprétées avec prudence ici, puisque cinq des dix distributions ne sont pas normales selon le test Agostino-Pearson. Néanmoins, compte tenu de la grande différence entre les moyennes et les cinq valeurs-p valides, il est clair que le réseau Deepseg est significativement supérieur.

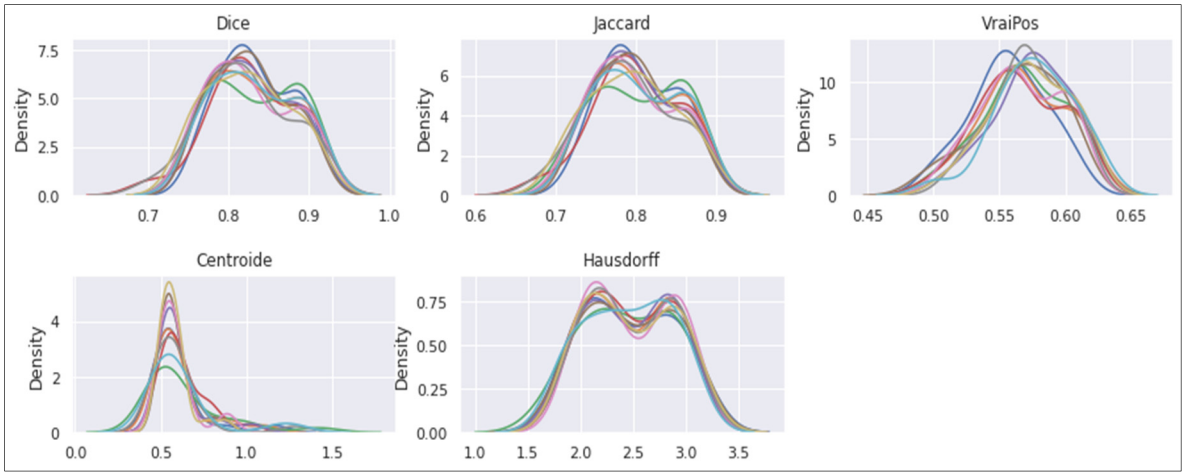



Figure 3.3 Estimations par noyau des résultats pour les dix itérations croisées. Chaque courbe représente un des dix réseaux et est constituée des 20 résultats obtenus sur les IRM de test. On peut observer que les distributions du graphique Centroïde (ligne centrale) sont biaisées vers la droite, ce qui explique pourquoi plusieurs distributions échouent le test de normalité

**Distance Hausdorff**

Tel que discuté à l'ANNEXE III, la distance Hausdorff est une métrique mesurant la distance maximale d'un ensemble de données au point le plus près d'un autre ensemble de données.

Tableau 3.6 Distances Hausdorff moyennes pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Distance Hausdorff** | 1 | 2.47 | 0.097 | -4.822 | 0.0001 |
| | 2 | 2.47 | 0.020 | -4.761 | 0.0001 |
| | 3 | 2.44 | 0.324 | -4.529 | 0.0002 |
| | 4 | 2.48 | 0.058 | -4.771 | 0.0001 |
| | 5 | 2.47 | 0.054 | -4.763 | 0.0001 |
| | 6 | 2.48 | 0.122 | -4.971 | 0.0001 |
| | 7 | 2.49 | 0.003 | -4.986 | 0.0001 |
| | 8 | 2.49 | 0.043 | -4.714 | 0.0002 |
| | 9 | 2.48 | 0.026 | -4.891 | 0.0001 |
| | 10 | 2.45 | 0.127 | -4.483 | 0.0003 |
| | **Moyenne** | 2.47 ***deepseg = 2.01*** | 0.088 | -4.769 | 0.0002 |
| | Écart-type | 0.02 | 0.093 | 0.166 | 0.0001 |

On peut voir sur le Tableau 3.6 que la distance Hausdorff moyenne pour le réseau Deepseg est de 2.01, tandis que la distance Hausdorff moyenne pour nos réseaux est de 2.47. Les valeurs-p doivent encore un fois être interprétées avec prudence ici, puisque quatre distributions ne sont pas normales selon le test Agostino-Pearson. Néanmoins, compte tenu de la différence considérable entre les moyennes et des cinq valeurs-p valides, il est clair que le réseau Deepseg possède en moyenne des distances Hausdorff plus courtes, et donc qu'il est supérieur par rapport à cette métrique.

Le Tableau 3.7 résume les résultats présentés plus haut. Il est à noter que les moyennes arithmétiques des valeurs-p tiennent seulement en compte les itérations où les résultats suivent une distribution normale. De plus, ces moyennes-p ne devraient pas être interprétées comme des statistiques exactes, mais plutôt comme des heuristiques, tel que discuté à la section 2.7.3.

Tableau 3.7 Tableau récapitulatif des résultats pour l'étape axiale

| | **Moyenne** | **Deepseg** | **Valeur-p** |
|---|---|---|---|
| **Coefficient Dice** | 82.9 | **87.9** | 0.012 |
| **Coefficient Jaccard** | 80.0 | **85.4** | 0.009 |
| **Taux vrai positif** | 0.57 | **0.89** | 4.72E-14 |
| **Ligne centrale** | 0.61 | **0.30** | 0.001 |
| **Distance Hausdorff** | 2.47 | **2.01** | 2.00E-04 |

Il est clair en observant le tableau que le réseau Deepseg est unilatéralement supérieur à nos réseaux axiaux. Cependant, ceci ne nous concerne pas réellement, car les réseaux axiaux ne sont au fond qu'un produit intermédiaire dans le processus global de l'approche en deux étapes. Ce sont donc les résultats de l'étape sagittale à la prochaine section qui importent réellement.

La supériorité du réseau Deepseg peut être expliquée, au moins partiellement, par le fait que toutes les IRM utilisées pour entrainer nos réseaux sont des IRM sagittales. En effet, la basse résolution dans le plan axial de ces IRM ne permet pas de capturer des caractéristiques détaillées. Des réseaux entrainés uniquement avec ces images sont donc limités par leur

résolution. À l'opposé, bien que Deepseg ait aussi été entrainé avec des images axiales, la majorité de celles-ci provenaient d'IRM axiaux, ce qui explique sa performance supérieure.

Il faut cependant signaler le fait que Deepseg a été entrainé avec des données externes provenant d'autres sites cliniques. Il est donc en désavantage par rapport à nos réseaux, puisque les IRM de test proviennent du même site que celles utilisées pour entrainer nos réseaux. Ceci pourrait conférer un avantage considérable à nos réseaux, bien que l'ampleur de cet avantage soit difficile à évaluer sans données de sites externes aux deux réseaux.

### 3.1.2 Étape 2 : segmentation sur les coupes sagittales

Pour déterminer la période optimale d'entrainement, les dix réseaux de validation croisée sont entrainés pour une durée de 125 époques. Ce nombre d'époques correspond à un nombre d'époques suffisamment élevé pour que les courbes des pertes de validation des sous-ensembles s'aplanissent ou bien commencent à remonter, ce qui correspond généralement au moment où les réseaux commencent à faire du *overfitting*, surtout lorsque les pertes d'entrainement continuent de baisser. Ce phénomène d'aplanissement est bien visible sur la Figure 3.4, qui montre la moyenne des pertes de validation pour la totalité des sous-ensembles.

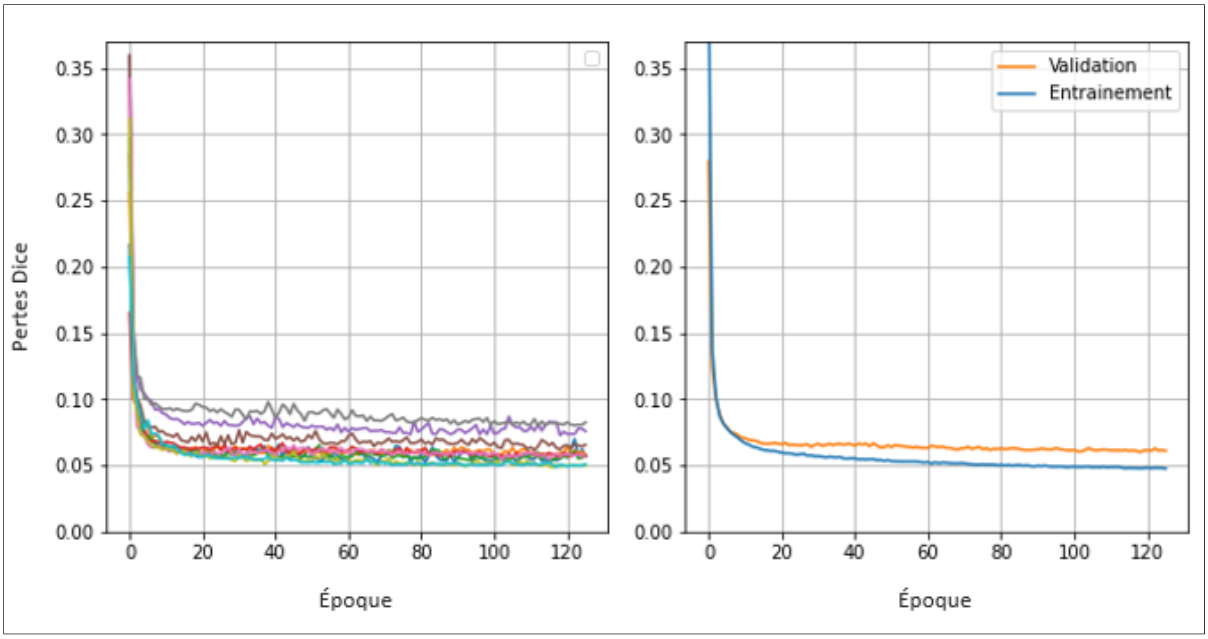



Figure 3.4 Courbes de pertes de validation pour chaque sous-ensemble (gauche) et moyenne des pertes de validation et d'entrainement pour tous les sous-ensembles (droite). La moyenne des pertes de validation atteint sa valeur minimale à l'époque 118.

**Coefficient Dice**

Le Tableau 3.8 affiche les pertes Dice de validation à l'époque finale de l'entrainement. L'entrainement de la deuxième étape étant effectué avec des images sagittales, ces valeurs peuvent être comparées directement avec celles au Tableau 3.9.

Tableau 3.8 Pertes et coefficients de validation Dice à l'époque 125

| Indice k | Pertes Dice | Coefficients Dice |
|---|---|---|
| 1 | 0.057 | 94.3 |
| 2 | 0.058 | 94.2 |
| 3 | 0.057 | 94.3 |
| 4 | 0.057 | 94.3 |
| 5 | 0.076 | 92.4 |
| 6 | 0.065 | 93.5 |
| 7 | 0.056 | 94.4 |
| 8 | 0.082 | 91.8 |
| 9 | 0.050 | 95.0 |
| 10 | 0.051 | 94.9 |
| **Moyenne** | 0.061 | 93.9 |
| Écart-type | 0.011 | 1.0 |

La moyenne des coefficients Dice sur les IRM de test est de 95.4, tandis que la moyenne des coefficients Dice de validation est de 93.9. Cette différence peut sembler curieuse au premier abord, puisque les dix réseaux sont identiques et que les données sont supposées avoir été sélectionnées aléatoirement à partir de la même population. Elle s'explique cependant par le fait que lors de l'évaluation sur les données de test, les segmentations subissent un post-traitement qui améliore leur performance face à certaines métriques, tel que discuté à la section précédente. Cette différence entre les coefficients Dice peut donc servir en quelque sorte comme mesure de l'amélioration produite par ce post-traitement.

Tableau 3.9 Moyenne des coefficients Dice sur les IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Coefficient Dice** | 1 | 95.0 | 0.077 | -5.204 | 5.05E-05 |
| | 2 | 95.2 | 0.156 | -5.816 | 1.33E-05 |
| | 3 | 95.1 | 0.199 | -5.791 | 1.40E-05 |
| | 4 | 95.1 | 0.167 | -5.803 | 1.37E-05 |
| | 5 | 95.4 | 0.145 | -6.829 | 1.62E-06 |
| | 6 | 95.5 | 0.148 | -6.839 | 1.58E-06 |
| | 7 | 95.7 | 0.174 | -7.005 | 1.14E-06 |
| | 8 | 95.7 | 0.162 | -7.057 | 1.02E-06 |
| | 9 | 95.3 | 0.137 | -6.145 | 6.61E-06 |
| | 10 | 94.8 | 0.066 | -5.269 | 4.37E-05 |
| | **Moyenne** | **95.3** *deepseg = 87.9* | 0.143 | -6.176 | 1.47E-05 |
| | Écart type | 0.3 | 0.042 | 0.708 | 1.80E-05 |

En observant le Tableau 3.9, on voit que la moyenne des coefficients Dice pour nos dix réseaux de validation croisée est de 95.3, tandis que la moyenne de Deepseg est de 87.9. Les valeurs-p pour cette différence se situent dans l'ordre de $1x10^{-5}$, indiquant que la différence est amplement significative. Il est donc permis de conclure que nos réseaux montrent une amélioration considérable des coefficients Dice par rapport à Deepseg.

**Coefficient Jaccard**

Le coefficient Jaccard étant apparenté au coefficient Dice, il est présenté afin que les deux coefficients, pris ensemble, se renforcent et se confirment. On s'attend donc ici à ce que les résultats de nos réseaux soient toujours supérieurs, ce qui est le clairement le cas.

Tableau 3.10 Moyennes des coefficients Jaccard sur les IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Coefficient Jaccard** | 1 | 93.0 | 0.188 | -5.710 | 1.67E-05 |
| | 2 | 93.2 | 0.237 | -6.155 | 6.48E-06 |
| | 3 | 93.0 | 0.252 | -6.163 | 6.36E-06 |
| | 4 | 92.9 | 0.250 | -6.145 | 6.61E-06 |
| | 5 | 93.3 | 0.193 | -7.066 | 1.01E-06 |
| | 6 | 93.4 | 0.251 | -7.008 | 1.13E-06 |
| | 7 | 93.5 | 0.209 | -7.136 | 8.77E-07 |
| | 8 | 93.6 | 0.228 | -7.161 | 8.34E-07 |
| | 9 | 93.3 | 0.183 | -6.654 | 2.30E-06 |
| | 10 | 92.8 | 0.130 | -5.575 | 2.24E-05 |
| | **Moyenne** | **93.2** *deepseg = 85.4* | 0.212 | -6.477 | 6.47E-06 |
| | Écart-type | 0.3 | 0.039 | 0.603 | 7.43E-06 |

**Taux de vrais positif**

Tel que discuté à l'ANNEXE III, taux de vrais positif est une métrique globale indiquant la proportion des pixels segmentés par le réseau qui sont aussi segmentés sur le masque. Comme on peut le voir sur le Tableau 3.11, l'ensemble de nos réseaux ont un taux supérieur à Deepseg, avec un taux moyen de 0.94, tandis que Deepseg a un taux moyen de 0.89. Cette différence est mise en évidence par les huit valeurs-p valides, qui sont dans l'ordre de $1x10^{-3}$, confirmant la supériorité de nos réseaux.

Tableau 3.11 Moyennes des taux de vrais positifs sur les IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Taux vrai positif** | 1 | 0.94 | 0.228 | -3.620 | 0.002 |
| | 2 | 0.94 | 0.064 | -3.476 | 0.003 |
| | 3 | 0.94 | 0.059 | -3.797 | 0.001 |
| | 4 | 0.94 | 0.119 | -3.364 | 0.003 |
| | 5 | 0.95 | 0.012 | -4.513 | 0.000 |
| | 6 | 0.95 | 0.046 | -3.806 | 0.001 |
| | 7 | 0.95 | 0.355 | -4.083 | 0.001 |
| | 8 | 0.95 | 0.263 | -4.121 | 0.001 |
| | 9 | 0.95 | 0.161 | -4.040 | 0.001 |
| | 10 | 0.94 | 0.065 | -3.571 | 0.002 |
| | **Moyenne** | **0.94** *deepseg = 0.89* | 0.137 | -3.839 | 0.002 |
| | Écart-type | 0.01 | 0.112 | 0.352 | 0.001 |

**Ligne centrale**

Tel que discuté à la section 2.1.3.1, la métrique de la ligne centrale en une métrique élaborée au cours de la présente étude afin de donner une mesure adaptée à la forme naturelle de la moelle épinière.

Tableau 3.12 Moyennes des résultats de la métrique Ligne centrale sur les IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Ligne centrale** | 1 | 0.19 | 0.303 | 1.760 | 0.0945 |
| | 2 | 0.21 | 0.000 | 1.241 | 0.2299 |
| | 3 | 0.23 | 0.000 | 0.961 | 0.3485 |
| | 4 | 0.21 | 0.724 | 1.546 | 0.1386 |
| | 5 | 0.21 | 0.787 | 1.403 | 0.1768 |
| | 6 | 0.20 | 0.614 | 1.429 | 0.1692 |
| | 7 | 0.20 | 0.239 | 1.516 | 0.1460 |
| | 8 | 0.23 | 0.000 | 0.898 | 0.3803 |
| | 9 | 0.19 | 0.396 | 1.837 | 0.0819 |
| | 10 | 0.20 | 0.000 | 1.494 | 0.1516 |
| | **Moyenne** | **0.21** *deepseg = 0.30* | 0.306 | 1.408 | 0.1345 |
| | Écart-type | 0.01 | 0.314 | 0.304 | 0.0387 |

On peut voir dans le Tableau 3.12 que nos réseaux affichent une distance moyenne de 0.21, tandis que Deepseg affiche une distance moyenne de 0.30. Nos réseaux sembleraient donc supérieurs. Cependant, l'ensemble des valeurs-p sont supérieures à 0.05, indiquant que les différences entre nos réseaux et Deepseg ne sont pas nécessairement significatives. Ceci pourrait être dû au fait que la distribution des résultats est plutôt large, comme on peut le voir en regardant le graphique « Ligne centrale » de la Figure 3.5. En effet, bien que cette distribution paraisse serrée au premier regard, en observant l'axe x du graphique, on s'aperçoit que les valeurs sont distribuées sur une plage assez large par rapport à l'écart observé entre nos deux résultats.

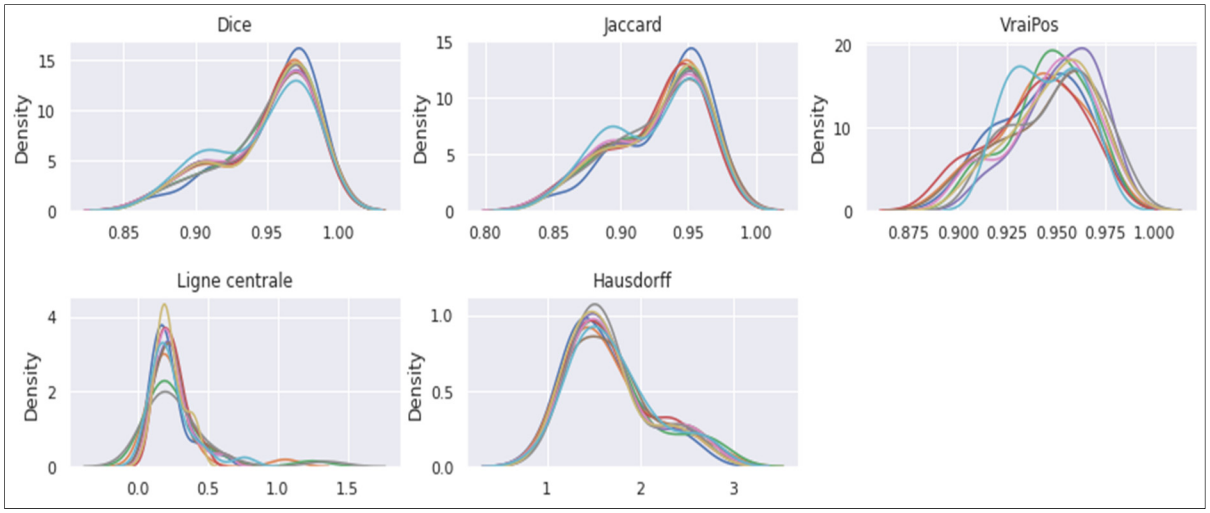



Figure 3.5 Densités des estimations par noyau des résultats pour les cinq métriques. Chaque courbe représente un des dix réseaux de validation croisée.

Le Tableau 3.12 indique aussi que quatre des dix valeurs-p ne sont pas valides. Ceci est probablement dû aux allongements vers la droite de certaines courbes que l'on peut observer sur le graphique mentionné précédemment. Les autres courbes semblent avoir une distribution normale et représentent probablement les itérations avec des valeurs-p valides. Suivant ces considérations, on ne peut pas conclure de façon significative que nos réseaux sont supérieurs à Deepseg en regard de cette métrique, bien qu'ils présentent une moyenne de distance environ 50% plus basse.

**Distance Hausdorff**

Tel que discuté à l'ANNEXE III, la distance Hausdorff est une métrique mesurant la distance maximale d'un ensemble de données au point le plus près d'un autre ensemble de données.

Tableau 3.13 Moyennes des distances Hausdorff sur les IRM de test pour les dix itérations de validation croisée

| | Indice k | Coefficient moyen | Test distribution | Valeur-t | Valeur-p |
|---|---|---|---|---|---|
| **Distance Hausdorff** | 1 | 1.49 | 0.078 | 5.929 | 1.05E-05 |
| | 2 | 1.50 | 0.290 | 5.856 | 1.22E-05 |
| | 3 | 1.53 | 0.031 | 5.594 | 2.15E-05 |
| | 4 | 1.51 | 0.349 | 5.549 | 2.37E-05 |
| | 5 | 1.51 | 0.038 | 5.567 | 2.28E-05 |
| | 6 | 1.52 | 0.122 | 5.160 | 5.57E-05 |
| | 7 | 1.51 | 0.116 | 5.532 | 2.46E-05 |
| | 8 | 1.50 | 0.077 | 5.379 | 3.43E-05 |
| | 9 | 1.49 | 0.038 | 5.421 | 3.13E-05 |
| | 10 | 1.54 | 0.242 | 4.928 | 9.34E-05 |
| | **Moyenne** | **1.51** *deepseg = 2.01* | 0.138 | 5.492 | 3.63E-05 |
| | Écart-type | 0.02 | 0.114 | 0.296 | 2.94E-05 |

On peut voir sur le Tableau 3.13 que la distance Hausdorff moyenne pour nos réseaux est de 1.51, tandis que la distance moyenne pour Deepseg est de 2.01. Les valeurs-p doivent être interprétées avec prudence ici encore, puisque trois distributions ne sont pas normales selon le test Agostino-Pearson. Cependant les sept valeurs-p valides sont dans l'ordre de $1x10^{-5}$. Il est donc raisonnable de conclure que la différence entre nos réseaux et Deepseg est significative.

Le Tableau 3.14 résume les résultats présentés plus haut. Il est à noter que les moyennes arithmétiques des valeurs-p tiennent seulement en compte les itérations où les résultats suivent une distribution normale, tel que discuté à la section 2.2.3.4. De plus, ces moyennes-p ne devraient pas être interprétées comme des statistiques exactes, mais plutôt comme des heuristiques, tel que discuté à la section 2.2.3.

Tableau 3.14 Récapitulatif des résultats pour l'étape sagittale

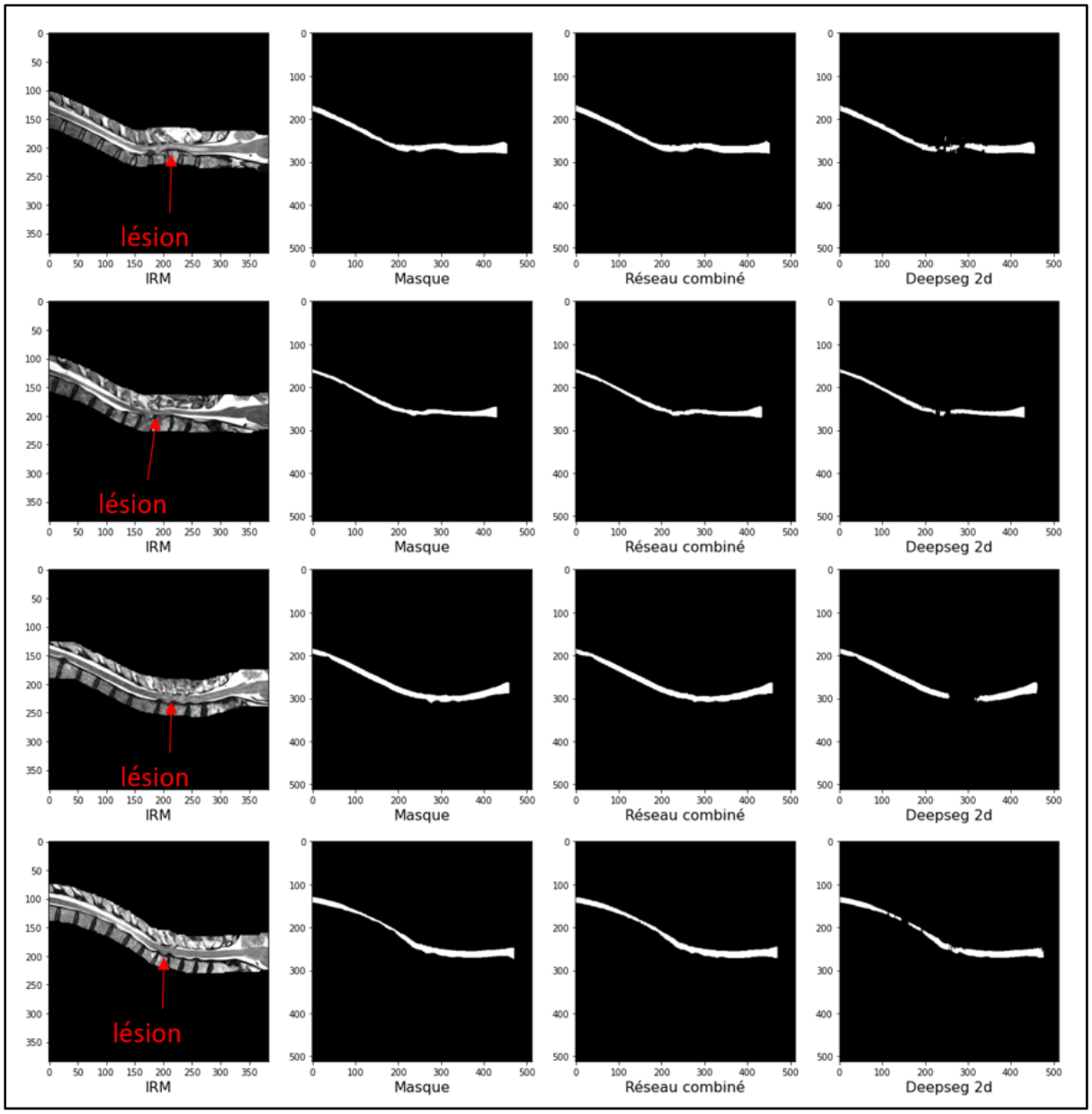
| | **Moyenne** | **Deepseg** | **Valeur-p** |
|---|---|---|---|
| **Coefficient Dice** | **95.3** | 87.9 | 1.47E-05 |
| **Coefficient Jaccard** | **93.2** | 85.4 | 6.47E-06 |
| **Taux vrai positif** | **0.94** | 0.89 | 0.002 |
| **Ligne centrale** | **0.21** | 0.30 | 0.1345 |
| **Distance Hausdorff** | **1.51** | 2.01 | 3.63E-05 |

Le Tableau 3.14 montre que les moyennes de nos dix réseaux sont supérieures à Deepseg pour l'ensemble des métriques. De plus, les valeurs-p sont significatives pour toutes les métriques hormis la ligne centrale. L'approche en deux étapes semblerait donc significativement supérieure à Deepseg. Cependant, il est important de nuancer ces résultats avec quelques points importants.

Premièrement, le réseau Deepseg est entrainé avec des données de sites externes, ce qui pourrait lui causer un désavantage considérable lors de l'évaluation sur nos données de test, car nos réseaux sont entrainés avec des IRM provenant du même site que les données de test. Il est cependant difficile de quantifier ce désavantage sans posséder d'IRM externes aux deux réseaux. Deuxièmement, il est important de considérer que le réseau Deepseg est conçu pour segmenter des IRM axiales ainsi que sagittales, tandis que nos réseaux sont conçus pour segmenter uniquement des IRM sagittales. Deepseg est donc plus souple que nos réseaux de ce point de vue. Cependant, il serait possible de créer une approche en deux étapes qui utiliserait les deux types d'IRM lors de l'entrainement et qui pourrait donc aussi segmenter les deux types. Ceci n'a pas été entrepris dans le cours de ce projet, car segmenter suffisamment d'IRM axiales aurait pris trop de temps. Il pourrait cependant être intéressant dans le futur d'entreprendre une telle démarche.

Pour ce qui est des protocoles d'acquisition, le réseau Deepseg et nos réseaux sont équivalents, c'est-à-dire que les deux sont conçus pour segmenter uniquement des IRM pondérées en T2 (les SCT possède cependant d'autres réseaux pour segmenter les IRM T1 et T2*). Deepseg et

nos réseaux sont aussi les deux capables de segmenter des IRM cervicaux et dorsaux sans discrimination et sans perte de précision, comme on peut le voir sur la Figure 3.6. Il est cependant à noter que les IRM lombaires peuvent poser problème lors de la détection de la ligne centrale par l'algorithme Optic C (voir l'ANNEXE I). Cependant, puisque Deepseg et nos réseaux utilisent tous deux cet algorithme, ce fait ne porte pas préjudice à la comparaison.



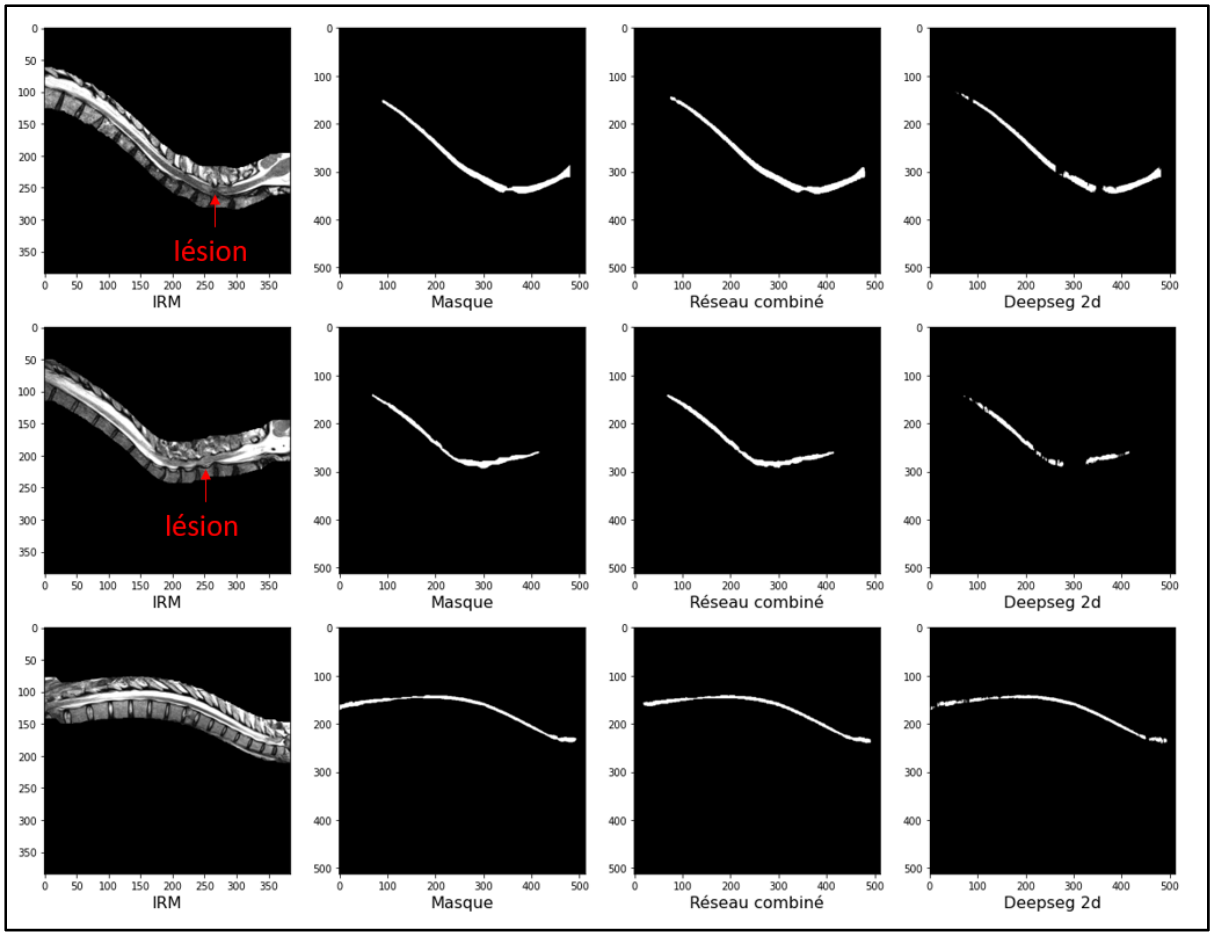



Figure 3.6 Comparaisons visuelles de notre approche avec Deepseg 2D

Comme on peut le voir sur la Figure 3.6, la majorité de nos segmentations apparaissent plus uniformes que celles de Deepseg, particulièrement dans les zones touchées par des lésions à la moelle épinière. Ces zones sont habituellement caractérisées par une décoloration de la moelle épinière, et les zones de lésions traumatiques perdent fréquemment leurs délimitations blanches formées par le liquide cérébro-spinal, les rendant encore plus difficiles à segmenter. La segmentation des lésions est donc la force principale de nos réseaux, rendue possible par le fait que la grande majorité des IRM utilisées proviennent de patients ayant subis des blessures traumatiques à la moelle épinière.

Une autre force de nos réseaux est l'habileté de segmenter les coupes sagittales qui sont en bordure de la moelle épinière. Sur ces coupes, la moelle n'apparait le plus souvent que

partiellement, comme on peut le voir sur la Figure 3.7. La forme caractéristique de la moelle épinière est donc perdue, ce qui rend plus difficile la segmentation. L'approche en deux étapes surmonte ceci en fournissant l'information du réseau axial au réseau sagittal, lui donnant ainsi plus d'informations pour segmenter les zones ambiguës.

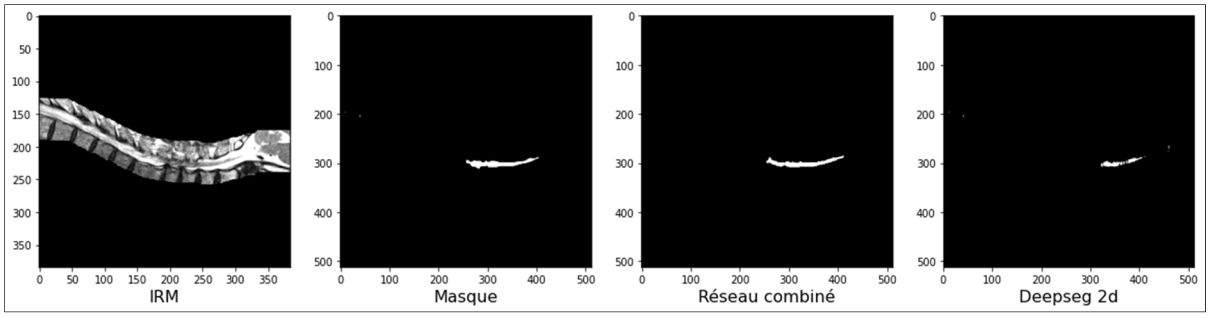



Figure 3.7 Exemple d'une coupe sagittale en bordure de la moelle épinière

Tel que mentionné à la section 2.1.1, les segmentations manuelles ont été effectuées à partir des segmentations de Deepseg afin de gagner du temps. Ceci pourrait favoriser Deepseg, car il y a de grandes sections des segmentations de Deepseg qui ne nécessitaient pas d'être retouchées. Dans l'étude (Gros et al., 2019), qui présente le réseau Deepseg 3D, les auteurs ont utilisé le même principe avec le réseau *propseg* pour produire leurs segmentations manuelles. Ils décrivent ceci comme un avantage pour propseg lors de la comparaison de leur réseau avec celui-ci.

#### 3.1.2.1 Comparaison de complexité

Pour ce qui est d'une comparaison de vitesse ou de complexité avec Deepseg 2D, l'unité de FLOPs se prête bien à la tâche (He et al., 2016; Hernandez et Brown, 2020). Un FLOPs représente une opération à virgule flottante lors d'une itération de propagation avant dans un réseau. Le module *ptflops* est conçu pour calculer les FLOPs d'un réseau donné. En utilisant ce module sur Deepseg et nos réseaux, on trouve que le réseau Deepseg 2D a un poids de 0.652

GFLOPs (milliard de FLOPs), tandis que notre réseau axial a un poids de 0.680 GFLOPs et notre réseau sagittal a un poids de 24.43 GFLOPs.

Ces chiffres représentent la quantité de FLOPs pour une seule propagation-avant. Afin d'arriver au nombre de FLOPs requis pour segmenter une IRM entière, il faut donc les multiplier par le nombre de coupes dans cette IRM. Donc, les IRM utilisées lors de cette étude ayant en moyenne 448 coupes axiales, la quantité de FLOPs requise pour la segmentation axiale est de 305 GFLOPs pour notre réseau et 292 GFLOPs pour Deepseg 2D. Lors de l'approche en deux étapes, la segmentation sagittale doit être comptabilisée aussi. Or, les IRM ont en moyenne 14 coupes sagittales, donc cela ajoute 342 GFLOPs au calcul. L'approche en deux étapes utilise finalement un total de 647 GFLOPs, tandis que Deepseg 2D n'utilise que 292 GFLOPs au total.

Une méthodologie alternative pour l'approche en deux étapes a cependant été développée afin de réduire la charge de calcul. Dans la cadre de cette méthode abrégée, les IRM sont rééchantillonnés de façon à réduire de moitié le nombre de coupes axiales préalablement à la segmentation. De cette façon, la charge du réseau axial est réduite de moitié. Suite à la segmentation axiale, les IRM sont rééchantillonnées à leur taille originale et plutôt que de segmenter l'ensemble des coupes sagittales, les coupes ayant moins de 20 pixels de segmentés par le réseau axial sont mises de côté comme étant vides. De cette façon, les nombre de coupes sagittales à segmenter passe d'une moyenne de 14 à environ 4.5. Suite à la segmentation sagittale, les coupes vides sont réintégrées aux IRM pour qu'elles reprennent leurs dimensions originales.

Avec cette façon de faire, les nombre de FLOPs requis pour segmenter une IRM avec l'approche en deux étapes passe de 647 GFLOPs à 262 GFLOPs, ce qui est inférieur au réseau Deepseg 2D, qui requiert 292 GFLOPs. De plus, la performance par rapport aux métriques est comparable à l'approche originale, comme on peut le voir sur le Tableau 3.15.

Tableau 3.15 Moyennes de la méthode abrégée et la méthode originale sur dix itérations de validation croisée

| | **Méthode abrégée** | **Méthode originale** |
|---|---|---|
| **Coefficient Dice** | 0.951 | **0.953** |
| **Coefficient Jaccard** | 0.928 | **0.932** |
| **Taux vrai positif** | 0.941 | **0.944** |
| **Ligne centrale** | 0.256 | **0.205** |
| **Distance Hausdorff** | 1.682 | **1.508** |

La méthode abrégée est un atout majeur pour l'approche en deux étapes, lui conférant non-seulement la supériorité quant à la qualité des segmentations par rapport à Deepseg 2D, mais aussi en regard de la rapidité d'exécution.

## 3.2 Approche semi-supervisée

Tous les réseaux sont initialement entrainés pour un total de 120 époques. Ceci correspond au moment où les courbes des pertes de validation s'aplanissent complètement alors que les pertes d'entrainement continuent de baisser. Suite à l'entrainement, la moyenne des pertes est calculée pour l'ensemble des itérations de validation croisée, comme on peut le voir sur la Figure 3.8. Ceci permet de bien observer le phénomène d'aplanissement des pertes de validation.

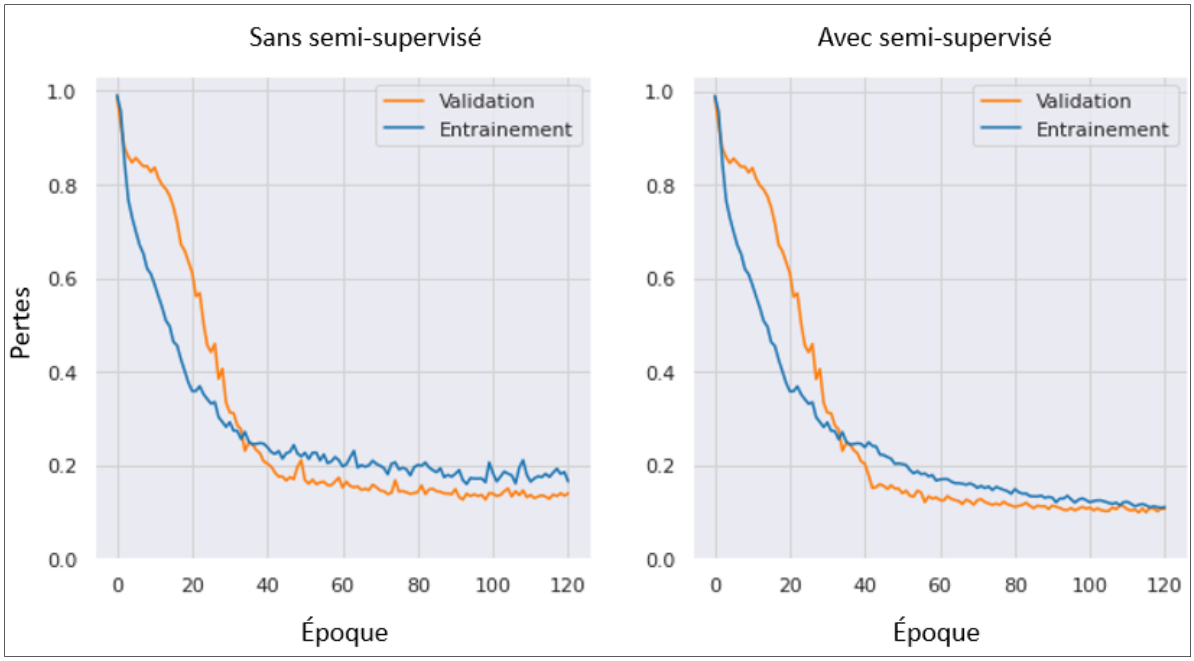



Figure 3.8 Pertes de validation moyennes pour l'approche avec 25% des données annotées. Puisque chacune des dix itérations est constituée de deux réseaux, ces courbes moyennes sont calculées à partir des données de 20 réseaux.

On peut observer sur la Figure 3.8 que les pertes de validation sont plus basses que les pertes d'entrainement, ce qui n'est pas commun. Cette situation est probablement dû au fait que les données d'entrainement sont ré-augmentées à chaque époque, fournissant un apport constant de nouvelles données et faisant ainsi en sorte que les réseaux aient plus de difficulté à généraliser les caractéristiques. Ceci est une bonne chose, car c'est un signe que l'augmentation fonctionne bien et aide à prévenir le *overfitting*, surtout vu le fait que les pertes de validation sont aussi basses.

Afin de déterminer quelle époque choisir pour les réseaux finaux, le minimum des courbes moyennes est retenu pour chaque approche. Ces valeurs sont visibles dans le Tableau 3.16.

Tableau 3.16 Époque avec la perte de validation moyenne minimale pour chaque approche

| | **Sans semi** | **Avec semi** |
|---|---|---|
| **25%** | 98 | 113 |
| **50%** | 115 | 110 |
| **75%** | 113 | 103 |
| **100%** | 107 | 119 |
| **Moyenne** | 109.7 | |

Ce sont donc les réseaux de l'époque 110 qui sont sélectionnés pour effectuer l'ensemble des tests présentés à la section suivante.

Tableau 3.17 Moyennes des coefficients Dice sur les IRM de test pour les dix itérations de validation croisée

| **Coefficient Dice** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Indice k** | **25%** | | **50%** | | **75%** | | **100%** | |
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| 1 | 83.0 | **91.3** | 90.6 | **92.0** | 90.5 | 90.3 | 91.9 | 92.2 |
| 2 | 87.8 | **89.8** | 91.0 | **93.8** | 91.1 | 90.9 | 89.5 | **91.8** |
| 3 | 85.5 | **91.3** | 88.9 | **91.6** | **91.4** | 90.0 | 91.5 | **92.1** |
| 4 | 82.6 | **92.0** | 90.5 | **92.2** | 90.5 | **92.5** | 93.0 | **93.7** |
| 5 | 87.2 | **89.5** | 89.7 | **91.5** | 88.5 | **91.7** | 91.0 | **92.2** |
| 6 | 83.8 | **91.4** | 88.7 | **89.3** | 83.4 | **92.1** | 91.5 | **93.4** |
| 7 | 85.3 | **90.1** | 87.0 | **91.0** | 86.0 | **94.1** | 90.8 | **92.7** |
| 8 | 85.7 | **90.3** | 88.5 | **92.3** | 88.8 | **92.4** | 90.1 | **93.9** |
| 9 | 84.8 | **88.9** | 85.8 | **93.0** | 91.1 | **92.2** | 91.5 | **92.1** |
| 10 | 88.1 | **91.1** | 88.4 | **90.8** | 90.9 | **93.1** | 89.1 | **89.8** |
| **Moyenne** | 85.4 | **90.6** | 88.9 | **91.8** | 89.2 | **91.9** | 91.0 | **92.4** |
| Écart-type | 1.9 | 1.0 | 1.6 | 1.2 | 2.6 | 1.3 | 1.2 | 1.2 |

Tableau 3.18 Moyennes des coefficients Jaccard sur les IRM de test pour les dix itérations de validation croisée

| Coefficient Jaccard | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Indice k** | **25%** | | **50%** | | **75%** | | **100%** | |
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| 1 | 79.5 | **88.3** | 87.6 | **89.5** | 87.1 | **87.6** | 89.2 | **89.7** |
| 2 | 84.3 | **86.7** | 87.9 | **91.2** | 88.0 | **88.1** | 86.2 | **89.3** |
| 3 | 82.2 | **88.3** | 85.7 | **88.8** | 88.3 | **87.4** | 88.8 | **89.6** |
| 4 | 79.1 | **88.9** | 87.5 | **89.6** | 87.4 | **89.8** | 90.1 | 91.2 |
| 5 | 83.9 | **86.6** | 86.6 | **88.8** | 85.1 | **89.0** | 88.2 | **89.8** |
| 6 | 80.6 | **88.4** | 85.6 | **86.4** | 80.3 | **89.4** | 88.7 | **91.0** |
| 7 | 81.9 | **87.1** | 83.9 | **88.2** | 82.8 | **91.7** | 87.8 | **90.2** |
| 8 | 82.3 | **87.2** | 85.5 | **89.6** | 85.6 | **89.8** | 87.0 | **91.4** |
| 9 | 81.3 | **85.7** | 82.4 | **90.2** | 88.1 | **89.6** | 88.7 | **89.7** |
| 10 | 84.6 | **88.0** | 85.2 | **87.9** | 87.9 | **90.4** | 86.0 | **87.0** |
| **Moyenne** | 82.0 | **87.5** | 85.8 | **89.0** | 86.1 | **89.3** | 88.0 | **89.9** |
| Écart-type | 1.9 | 1.0 | 1.7 | 1.3 | 2.7 | 1.3 | 1.3 | 1.3 |

Tableau 3.19 Moyennes des taux de vrai positif sur les IRM de test pour les dix itérations de validation croisée

| Taux vrai positif | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Indice k** | **25%** | | **50%** | | **75%** | | **100%** | |
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| 1 | 0.52 | **0.82** | 0.78 | **0.85** | **0.85** | 0.78 | 0.83 | **0.86** |
| 2 | 0.68 | **0.75** | 0.78 | **0.88** | 0.78 | **0.79** | 0.72 | **0.82** |
| 3 | 0.63 | **0.80** | 0.74 | **0.83** | **0.80** | 0.79 | 0.81 | **0.84** |
| 4 | 0.50 | **0.80** | 0.77 | **0.84** | 0.76 | **0.83** | 0.84 | **0.87** |
| 5 | 0.68 | **0.76** | 0.75 | **0.81** | 0.70 | **0.81** | 0.79 | **0.84** |
| 6 | 0.56 | **0.81** | 0.71 | **0.75** | 0.56 | **0.83** | 0.80 | **0.87** |
| 7 | 0.60 | **0.76** | 0.66 | **0.79** | 0.63 | **0.89** | 0.78 | **0.85** |
| 8 | 0.60 | **0.76** | 0.73 | **0.84** | 0.73 | **0.84** | 0.76 | **0.87** |
| 9 | 0.58 | **0.70** | 0.61 | **0.84** | 0.79 | **0.83** | 0.80 | **0.85** |
| 10 | 0.68 | **0.78** | 0.71 | **0.79** | 0.79 | **0.86** | 0.73 | **0.77** |
| **Moyenne** | 0.60 | **0.77** | 0.72 | **0.82** | 0.74 | **0.83** | 0.79 | **0.84** |
| Écart-type | 0.06 | 0.04 | 0.05 | 0.04 | 0.09 | 0.03 | 0.04 | 0.03 |

Tableau 3.20 Moyennes des lignes centrales sur les IRM de test pour les dix itérations de validation croisée

| Ligne centrale | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Indice k** | **25%** | | **50%** | | **75%** | | **100%** | |
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| 1 | 2.06 | **0.75** | 0.68 | **0.44** | 0.64 | **0.50** | 0.54 | **0.38** |
| 2 | 1.19 | **0.68** | 0.74 | **0.43** | 0.63 | **0.55** | 0.93 | **0.46** |
| 3 | 1.32 | **0.55** | 0.92 | **0.53** | 0.70 | **0.49** | 0.58 | **0.35** |
| 4 | 1.89 | **0.51** | 0.86 | **0.45** | 0.70 | **0.38** | 0.54 | **0.33** |
| 5 | 1.08 | **0.71** | 0.76 | **0.44** | 1.28 | **0.45** | 0.54 | **0.33** |
| 6 | 2.08 | **0.60** | 0.82 | **0.61** | 1.79 | **0.46** | 0.53 | **0.31** |
| 7 | 1.67 | **0.62** | 1.01 | **0.71** | 1.53 | **0.32** | 0.70 | **0.36** |
| 8 | 1.60 | **0.63** | 0.81 | **0.37** | 0.99 | **0.44** | 0.72 | **0.32** |
| 9 | 2.19 | **1.28** | 1.83 | **0.49** | 0.70 | **0.50** | 0.55 | **0.38** |
| 10 | 1.63 | **0.77** | 1.08 | **0.57** | 0.67 | **0.40** | 0.74 | **0.66** |
| **Moyenne** | 1.67 | **0.71** | 0.95 | **0.50** | 0.96 | **0.45** | 0.64 | **0.39** |
| Écart-type | 0.39 | 0.22 | 0.33 | 0.10 | 0.42 | 0.07 | 0.13 | 0.11 |

Tableau 3.21 Moyennes des distances Hausdorff sur les IRM de test pour les dix itérations de validation croisée

| Distance Hausdorff | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Indice k** | **25%** | | **50%** | | **75%** | | **100%** | |
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| 1 | 2.70 | **2.14** | 2.20 | **1.97** | 2.23 | **2.11** | 2.03 | **1.94** |
| 2 | 2.38 | **2.20** | 2.14 | **1.86** | 2.15 | **2.07** | 2.26 | **1.97** |
| 3 | 2.53 | **2.12** | 2.31 | **2.04** | **2.12** | 2.13 | 2.02 | **1.91** |
| 4 | 2.71 | **2.08** | 2.20 | **2.01** | 2.20 | **1.99** | 1.98 | **1.82** |
| 5 | 2.44 | **2.27** | 2.24 | **2.07** | 2.31 | **2.03** | 2.15 | **1.91** |
| 6 | 2.62 | **2.06** | 2.33 | **2.20** | 2.63 | **1.92** | 2.07 | **1.83** |
| 7 | 2.53 | **2.22** | 2.40 | **2.10** | 2.46 | **1.79** | 2.17 | **1.90** |
| 8 | 2.53 | **2.20** | 2.30 | **2.00** | 2.28 | **1.97** | 2.23 | **1.80** |
| 9 | 2.53 | **2.26** | 2.44 | **1.95** | 2.13 | **1.94** | 2.06 | **1.89** |
| 10 | 2.33 | **2.09** | 2.31 | **2.11** | 2.16 | **1.91** | 2.28 | **2.19** |
| **Moyenne** | 2.53 | **2.16** | 2.29 | **2.03** | 2.27 | **1.98** | 2.13 | **1.92** |
| Écart-type | 0.12 | 0.08 | 0.09 | 0.10 | 0.16 | 0.10 | 0.11 | 0.11 |

Comme on peut le voir dans les Tableaux 3.17 à 3.21, la quasi-totalité des dix itérations pour chaque métrique ont des coefficients supérieurs avec l'apprentissage semi-supervisé, et ce peu

importe le pourcentage de données annotées. De plus, l'écart-type des itérations semi-supervisées est inférieur pour l'ensemble des métriques et des approches hormis celle avec 100% des données annotées, ce qui indique une capacité de généralisation supérieure et une stabilité accrue pour les réseaux semi-supervisés.

Tableau 3.22 Tableau récapitulatif des résultats. Les valeurs représentent les moyennes des dix itérations de validation croisée sur les IRM de test

| | **25%** | | **50%** | | **75%** | | **100%** | |
|---|---|---|---|---|---|---|---|---|
| | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi | Sans semi | Avec semi |
| **Coefficient Dice** | 85.4 | **90.6** | 88.9 | **91.8** | 89.2 | **91.9** | 91.0 | **92.4** |
| **Coefficient Jaccard** | 82.0 | **87.5** | 85.8 | **89.0** | 86.1 | **89.3** | 88.0 | **89.9** |
| **Taux vrai positif** | 0.60 | **0.77** | 0.72 | **0.82** | 0.74 | **0.83** | 0.79 | **0.84** |
| **Ligne centrale** | 1.67 | **0.71** | 0.95 | **0.50** | 0.96 | **0.45** | 0.64 | **0.39** |
| **Distance Hausdorff** | 2.53 | **2.16** | 2.29 | **2.03** | 2.27 | **1.98** | 2.13 | **1.92** |

Le Tableau 3.22 et la Figure 3.9 montrent que les moyennes des quatre approches sont supérieures à leurs contreparties sans semi-supervision pour l'ensemble des métriques. La différence est particulièrement marquée pour l'approche avec 25% des données annotées, où les réseaux semi-supervisés obtiennent des coefficients près des valeurs obtenues par les réseaux supervisés avec 100% des données. Ceci illustre donc encore une fois la supériorité des réseaux semi-supervisés.

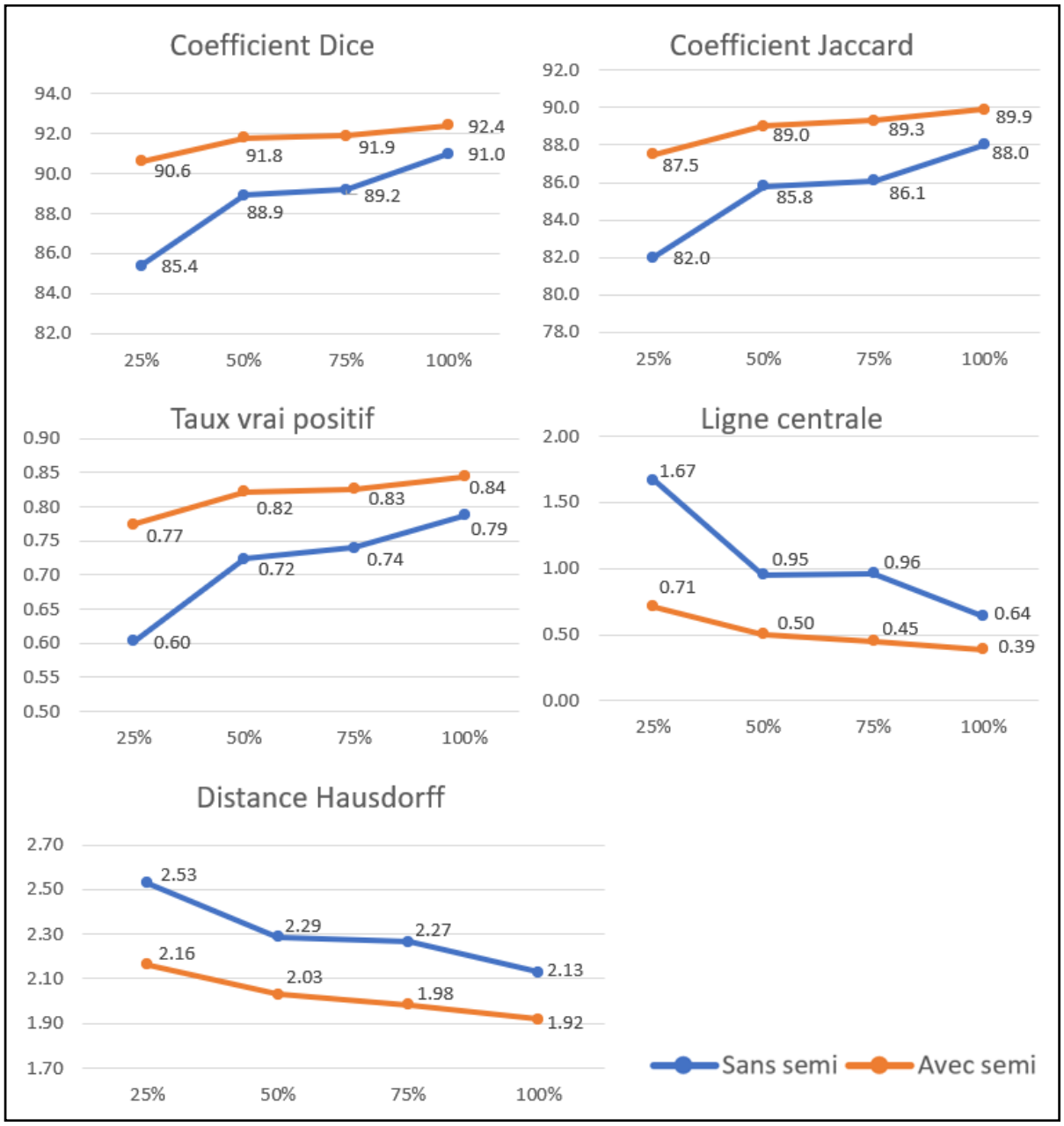



Figure 3.9 Moyennes des cinq métriques pour les quatre approches

On peut voir sur la Figure 3.9 que la différence entre les deux types de réseaux diminue au fur et à mesure qu'un pourcentage plus élevé de données sont annotées. Ceci correspond au scénario attendu, puisque le réseau semi-supervisé perd graduellement son avantage d'exploiter plus de données. Cependant, il est remarquable de voir que le réseau semi-supervisé demeure supérieur sur l'ensemble des métriques même avec 100% des données annotées. Ceci

pointe vers la possibilité que cette technique de semi-supervision puisse être utilisée non-seulement pour exploiter des données non-annotées, mais aussi pour améliorer des entrainements où l'intégralité des données sont annotées.

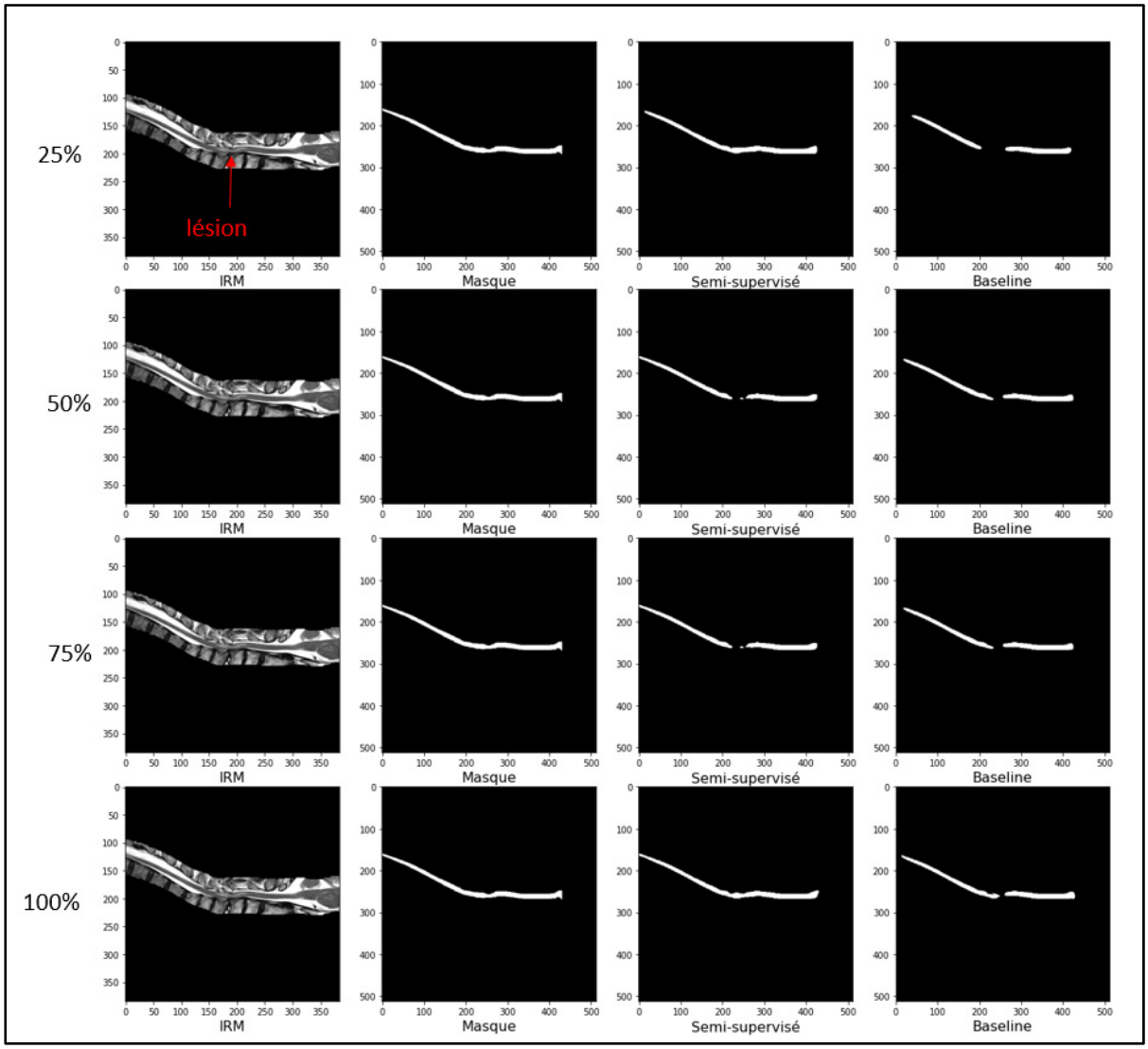



Figure 3.10 Segmentation de la même coupe sagittale avec différentes proportions de données annotées

La Figure 3.10 montre un exemple d'une coupe sagittale segmentée avec et sans semi-supervision. On voit que la différence entre le réseau semi-supervisé et celui uniquement supervisé (*baseline*) est que plus marquée pour l'approche utilisant 25% des données

supervisées que pour les autres approches. Ceci correspond au scénario attendu et concorde avec la Figure 3.9. On voit aussi que lors de l'approche avec 100% des données, le réseau semi-supervisé réussi à segmenter la lésion médullaire, tandis que le réseau *baseline* ne réussit pas tout à fait, ce qui vient illustrer l'amélioration produite par la technique semi-supervisée même avec des quantités de données annotées plus élevées.

# CONCLUSION

Le premier objectif de ce mémoire était de développer un outil capable de segmenter les moelles épinières atteintes de lésions traumatiques. Nos recherches ont mené à une nouvelle approche utilisant deux réseaux de type U-Net avec portes d'attention en cascade pour analyser les IRM dans le plan axial et sagittal. Cette approche s'est révélée être significativement supérieure à la meilleure approche du Spinal Cord Toolbox, Deepseg 2D, qui était prise comme état de l'art. La contribution principale de ce mémoire est donc l'élaboration de cette stratégie de réseaux combinés, qui est capable de segmenter des moelles épinières saines aussi bien que des instances atteintes de lésions maladives et traumatiques.

Le deuxième objectif de ce mémoire était d'élaborer une stratégie pour segmenter la moelle épinière qui bénéficierait d'un apport d'apprentissage semi-supervisé. Or, puisque nos résultats indiquent que les réseaux semi-supervisés développés dans ce projet sont supérieurs à leurs contreparties supervisés sur l'ensemble des métriques et pour l'ensemble des proportions de données annotées, il est permis de conclure que cet objectif a été atteint. Les réseaux semi-supervisés performent même de façon comparable avec 25% des données annotées que les réseau supervisés ayant 100% des données annotées. De plus, bien que notre approche semi-supervisée soit inspirée d'une autre approche (Chen et al., 2021), elle modifie en grande partie les éléments de cette implémentation et adapte avec succès la stratégie de base à un nouveau domaine, soit le domaine médical. Notre approche semi-supervisée est donc une contribution additionnelle notable de ce mémoire.

Les limitations de cette étude portent surtout sur les comparaisons des résultats de l'approche en deux étapes avec le réseau Deepseg 2D. En effet, bien que nos résultats obtenus sur les IRM de test soient considérablement supérieurs à ceux de Deepseg 2D, il est important de considérer que ces IRM des test proviennent du même site clinique que les IRM qui ont été utilisées pour entrainer nos réseaux, tandis que Deepseg 2D n'a jamais été exposé à des données de ce site. Nos réseaux possèdent donc un avantage dans ce sens. D'un autre côté, le réseau Deepseg 2D

a été utilisé comme point de départ lors des annotations manuelles afin de sauver du temps, ce qui lui confère à son tour un avantage, bien que probablement moins important.

Dans le cadre de ce projet, l'approche en deux étapes est limitée à l'analyse d'IRM sagittaux. Il serait cependant intéressant de développer une approche pour segmenter les IRM axiales avec la même technique en deux étapes. Dans ce cas, la première étape serait sagittale avec des coupes à résolution réduites, puis la deuxième étape serait axiale avec des coupes à haute résolution. Il est probable que deux paires de réseaux combinés axial/sagittal et sagittal/axial pris séparément soient plus performantes qu'une seule paire combinant les deux types d'IRM. De plus, puisqu'il est possible de détecter automatiquement le type d'IRM en utilisant leurs headers, il serait possible de créer un outil automatique qui acheminerait les IRM vers les réseaux appropriés lors de l'inférence. Une telle approche ne comporterait donc pas de désavantages envisageables par rapport à une approches mixte, outre une conception légèrement plus complexe, et offrirait probablement une performance supérieure vu la spécificité accrue des réseaux.

Une autre voie de travaux futurs intéressante serait de combiner l'approche en deux étapes avec l'approche semi-supervisée. Ceci n'a pas été réalisé lors de ce projet, car pour construire l'approche semi-supervisée, il convenait en premier lieu de travailler sur un problème simple et difficile afin de développer plus facilement les configurations et mettre en évidence les améliorations. Or, le problème de segmentation sagittale seule était bien adapté à la tâche. Cependant, cette approche étant maintenant bien développée, il serait intéressant de voir l'impact de l'apprentissage semi-supervisé sur l'approche en deux étapes.

# ANNEXE I

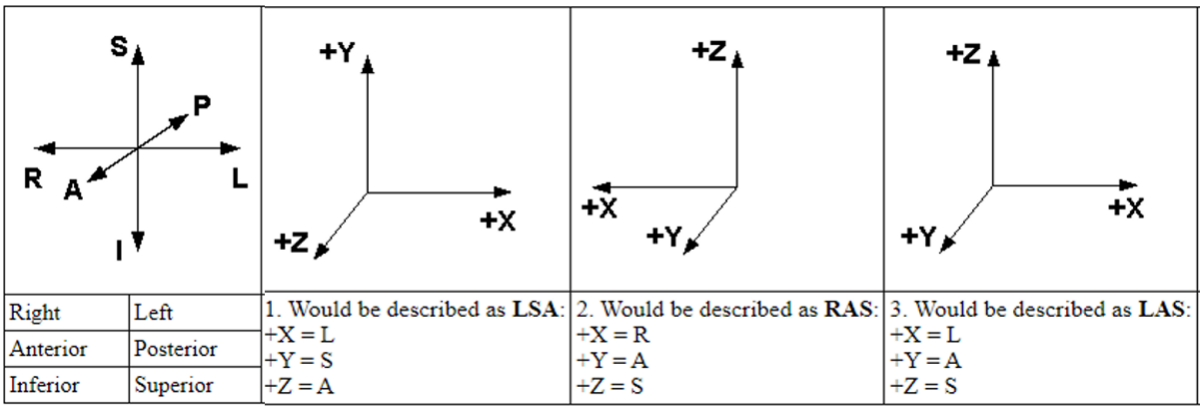
## Prétraitement des données

### Importation

Les données brutes parvenant de l'hôpital sont au départ en format NIfTI. Elles sont importées dans l'environnement Python en utilisant NiBabel, qui est une librairie spécialisée pour travailler avec les IRM en format NIfTI. Les IRM sont importées en tant qu'objets *image* de NiBabel. Ces objets NiBabel ont plusieurs propriétés utiles dont l'accès simplifié aux informations contenues dans les headers.

### Réorientation

La première étape dans la standardisation des IRM est de réorienter tous les IRM selon la même orientation, car certains IRM pourraient être sauvegardés sous différentes orientations. Dans notre système, les IRM sont donc systématiquement réorientées dans l'orientation RPI. Cette orientation se base sur le système d'axes décrit à la Figure-A I-1 . Contrairement à ce que l'on pourrait penser en regardant cette figure, l'orientation RPI ne suit pas les axes R, P et I, mais bien les axes L, A et S.




| | | | | |
|---|---|---|---|---|
| Right | Left | 1. Would be described as **LSA**:<br>+X = L<br>+Y = S<br>+Z = A | 2. Would be described as **RAS**:<br>+X = R<br>+Y = A<br>+Z = S | 3. Would be described as **LAS**:<br>+X = L<br>+Y = A<br>+Z = S |
| Anterior | Posterior | | | |
| Inferior | Superior | | | |

Figure-A I-1 Système d'axes pour l'orientation de voxels dans les IRM. Tirée de Graham Wideman, 2003

La réorientation est opérée en réemployant une partie du code du Spinal Cord Toolbox, qui est publiquement disponible sur GitHub. En bref, l'information sur l'orientation de l'IRM, qui est contenue dans l'objet *image* de NiBabel, est utilisée afin de permuter les axes de l'IRM selon la logique des axes de la Figure-A I-1.

L'orientation RPI est aussi l'orientation qui est utilisée par le SCT. Elle a été choisie pour faciliter l'utilisation de l'outil de recadrage Optic C décrit plus bas, et aussi vu le fait qu'aucune autre orientation ne présentait d'avantages envisageables. Il est à noter que la même réorientation est aussi appliquée au masque correspondant à chaque IRM, afin que les deux continuent à être alignés.

**Rééchantillonnage**

La deuxième étape du prétraitement suivant la réorientation est le rééchantillonnage des voxels composant les images. La résolution des coupes sagittales utilisées varie entre 384 × 384 et 512 × 512 pixels, tandis que la résolution axiale varie entre cette résolution sur un axe et une résolution déterminée par l'épaisseur des coupes sagittales et l'espace entre elles sur l'autre axe. L'épaisseur des coupes sagittales est de 3 mm pour l'ensemble des données, tandis que l'espace entre les coupes varie entre 3.3 et 3.6 millimètres.

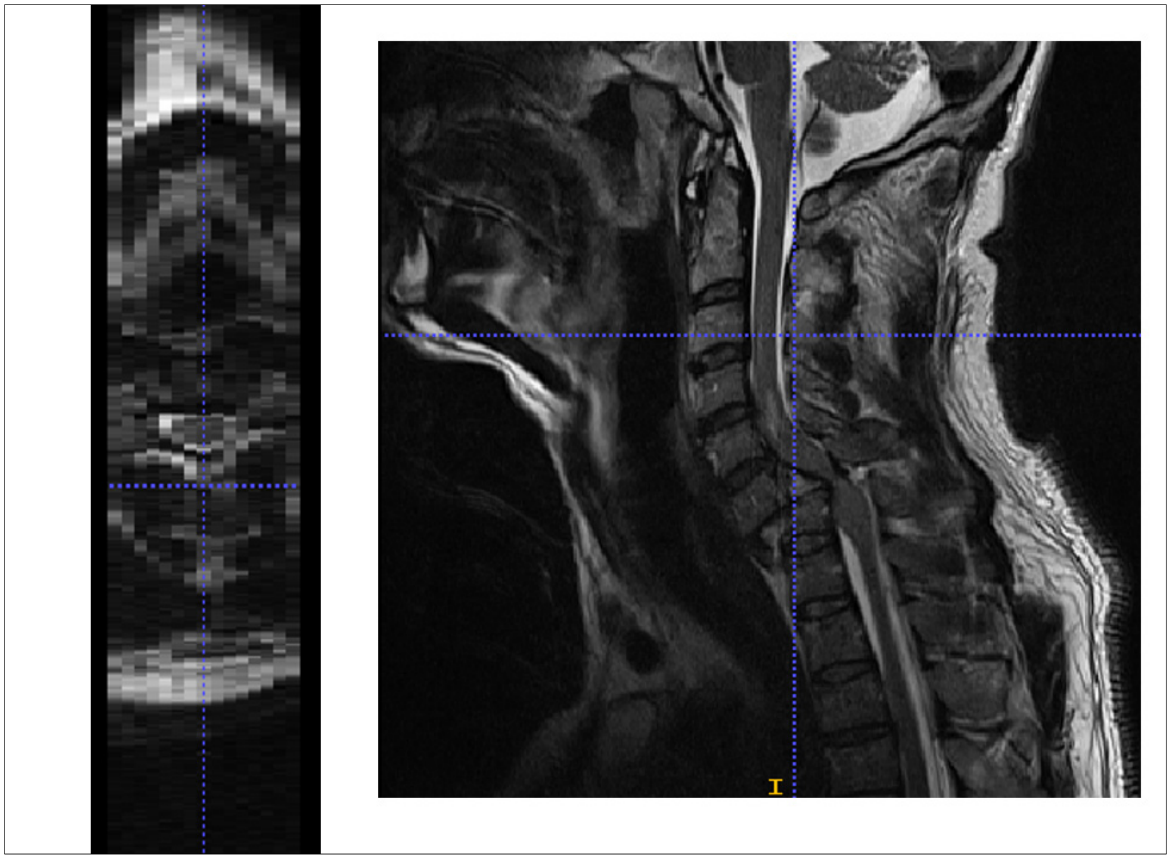


Figure-A I-2  Résolution axiale de 15 × 384 (gauche) et résolution sagittale de 384 × 384 (droite) pour une IRM sagittale avant le rééchantillonnage. Chaque tranche verticale sur la vue axiale (gauche) correspond à une coupe sagittale

L'étape de rééchantillonnage est nécessaire afin d'uniformiser les données par rapport au volume qu'elles représentent dans l'espace. En effet, les différentes IRM n'ont non-seulement pas toutes la même résolution en voxels, elles sont aussi différentes par rapport au volume qu'elles couvrent dans l'espace. Cette étape vise donc à rééchantillonner les IRM afin que chaque voxel représente le même volume. Cette condition est nécessaire afin que l'algorithme de recadrage à l'étape suivante produise des données uniformes, où la moelle épinière est les autres traits sont tous de la même grosseur par rapport à la dimension de l'image recadrée.

Afin d'arriver à un tel rééchantillonnage, il est nécessaire d'employer certaines informations complémentaires contenues dans les headers des fichiers NIfTI. Ces headers contiennent, parmi autres choses, les *zooms* des IRM. Ces zooms représentent la distance couverte par un

voxel dans chacun des trois axes spatiaux. Chaque IRM comporte donc trois zooms, correspondants aux trois dimensions de l'espace. À l'aide des zooms et des dimensions de l'IRM, il est possible de calculer les dimensions en voxels que chaque axe devrait avoir afin de représenter un volume donné dans l'espace. Ce volume est choisi en spécifiant dans ce calcul la distance en millimètres que devrait représenter un voxel dans chaque plan, comme on peut le voir sur la Figure-A I-3.

Dans cette étude, la valeur choisie pour cette distance est de 0.5 mm dans le plan axial. Cette résolution est bien adaptée au recadrage à 64 × 64 pixels qui aura lieu à l'étape suivante, car elle proportionne bien la moelle épinière sur des images de cette grandeur. De plus, cette résolution s'est montrée suffisante pour permettre aux caractéristiques clés des images d'être conservées et apprises par des réseaux convolutifs par la suite (Gros et al., 2019). Elle correspond aussi à la résolution utilisée par le Spinal Cord Toolbox pour la segmentation axiale.

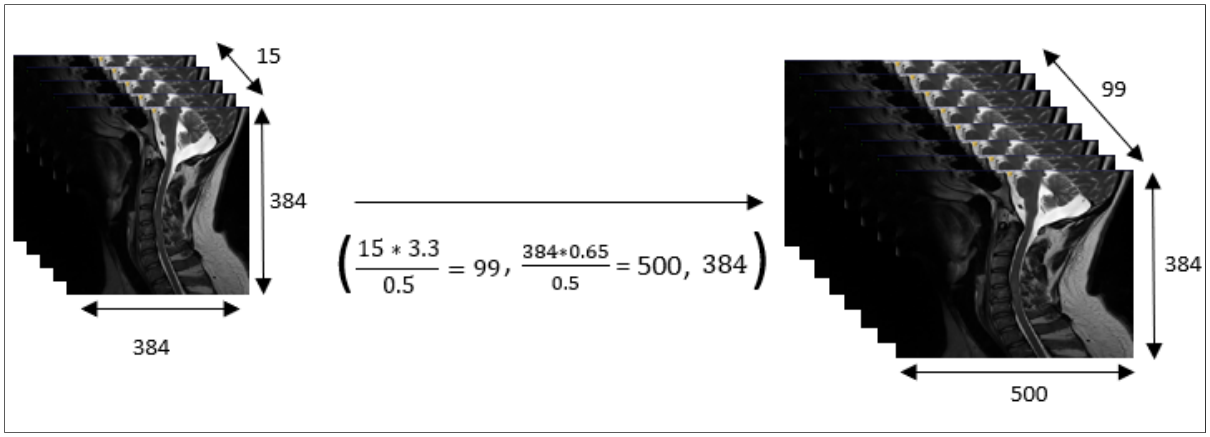



Figure-A I-3 Rééchantillonnage d'une IRM de dimension (15, 384, 384) à 0.5 mm dans le plan axial. Dans ce cas, la dimension des zooms en millimètres est de (3.3, 0.65, 0.65), tel qu'illustré. Il est à noter que la troisième dimension n'est pas modifiée puisqu'elle ne fait pas partie du plan axial. La transformation affine n'est pas illustrée

L'étape suivante dans la standardisation des IRM est une transformation affine. L'objectif de la transformation affine est en quelque sorte de pivoter et déplacer dans l'espace les voxels afin qu'ils se retrouvent alignés avec le cadre de référence de la machine à IRM. De cette façon, les

organes présentes sur les images se retrouvent toutes alignées de la même façon sur chaque IRM, peu importe l'angle du patient par rapport à la machine lors du scan. Cette transformation est effectuée en utilisant les données affines contenues dans les headers des IRM. Ces données, qui sont créées au moment du scan, décrivent la position des voxels par rapport à un espace de référence dont les coordonnées d'origine se trouvent à l'isocentre magnétique de la machine.

Le processus de transformation affine en tant que tel redimensionne aussi les IRM aux dimensions trouvées à l'étape précédente en effectuant une interpolation linéaire. Le tout est effectué à l'aide de la librairie NiBabel. Dans ce cas, la librairie NiBabel utilise le module *scipy.ndimage.affine_transform* comme base pour la transformation. Il est à noter que le même processus de rééchantillonnage est aussi appliqué au masque correspondant à chaque IRM, afin que les deux continuent d'être alignés.

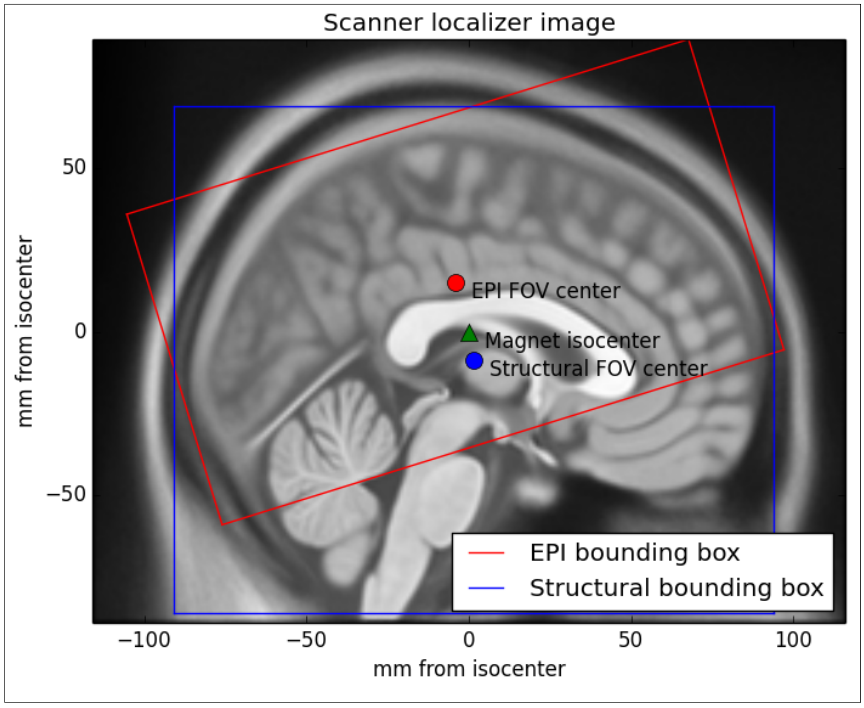



Figure-A I-4 L'objectif de la transformation affine est de réaligner le centre de l'IRM (rouge) avec le centre magnétique (vert) et le cadre de l'IRM avec le cadre de référence (bleu)
Tirée de Chris Markiewicz, 2022

**Recadrage**

Suite au rééchantillonnage, un algorithme est appliqué aux IRM afin d'isoler un certain volume autour de la colonne vertébrale. Cet algorithme, nommé Optic (Gros et al., 2018), se base sur des machines à vecteurs de support. Suite à son application, les coupes axiales sont recadrées à 64 × 64 pixels autour de la colonne vertébrale. Le recadrage est appliqué aux IRM ainsi qu'aux masques, afin que les images continuent de s'aligner lorsqu'on les superpose.

Cet algorithme facilite grandement le travail de segmentation qui aura lieu par la suite. En éliminant plus de 90% des pixels qui ne contiennent pas de moelle épinière, le réseau de neurones segmentant la moelle peut beaucoup mieux concentrer ses ressources sur les caractéristiques critiques à cette tâche plutôt que d'apprendre des caractéristiques superflues. De plus, cela évite au réseau de se tromper sur l'emplacement de la moelle épinière en réduisant l'espace de solutions possibles. Cet algorithme de recadrage est donc critique dans la préparation des données.

**Normalisation axiale**

L'étape finale du prétraitement avant que les coupes soient envoyées au réseau axial est la normalisation des intensités. Cette normalisation vise à augmenter le contraste de la moelle épinière avec les structures environnantes. Ceci est réalisé en limitant les valeurs contenues dans chaque coupe axiale à des valeurs limites. Il y a donc deux valeurs limites, une pour le bas de l'intensité et une pour le haut. La valeur pour la basse intensité est ajustée à la valeur du 2$^{e}$ percentile des pixels de l'image, tandis que la valeur pour la haute intensité est ajustée à la valeur du 98$^{e}$ percentile de l'image. L'effet de cette normalisation est que tous les pixels se trouvant au-delà de ces seuils sont ajustés aux valeurs limites. Il y a donc une perte des contrastes aux extrémités du spectre. Cependant, dans la pratique, cela a pour effet d'uniformiser les pixels de la moelle épinière et de mettre en évidence ses bordures, comme on peut le voir sur la Figure-A I-5.

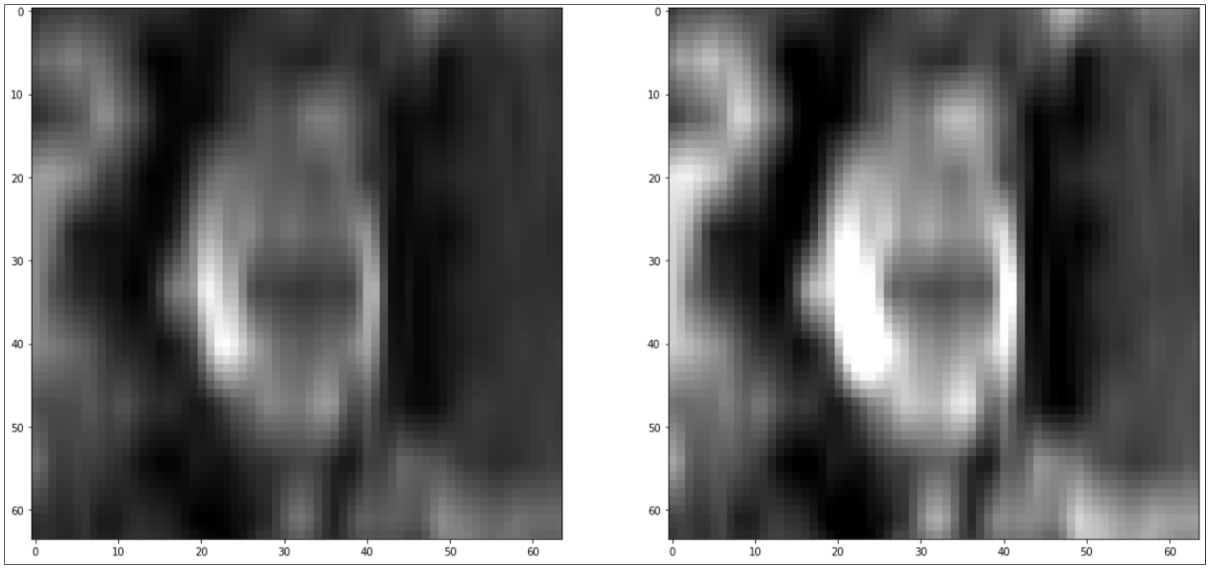


Figure-A I-5 Coupe axiale suite au rééchantillonnage et au recadrage autour de la colonne. L'image de droite représente la coupe suite à la normalisation des intensités

La gamme de valeurs pour les pixels des images est aussi réajustée lors de la normalisation. Préalablement à ce réajustement, la gamme de valeurs pour les pixels de chaque image n'est pas fixe. En effet, préalablement au prétraitement, elle varie généralement entre 0 et 1400, puis suite au recadrage, elle varie généralement dans une gamme plus restreinte, car une grande partie des valeurs extrêmes ont été éliminées. Ceci n'est pas nécessairement significatif pour la présentation des données à l'œil humain, car lors de ce processus, les contrastes sont automatiquement présentés sur une échelle adaptée à chaque gamme de valeurs. De cette façon, deux images ayant des gammes différentes mais une distribution identique sont généralement représentés identiquement.

Ceci n'est cependant pas le cas pour la façon dont les réseaux convolutifs interprètent les données. Pour ces réseaux, l'hétérogénéité des gammes de données représente un obstacle supplémentaire à la généralisation auquel ils doivent allouer des ressources. Il est donc utile lors du prétraitement de normaliser les gammes, car cette opération se fait facilement et facilite grandement le travail des réseaux. De plus, la normalisation permet d'accroitre la capacité des réseaux à généraliser sur de nouvelles données, car dans le cas où les données ne seraient pas normalisées, les réseaux pourraient avoir de la difficulté à effectuer des prédictions sur des

gammes de données auxquelles ils n'avaient pas accès durant l'entrainement. La normalisation est donc une étape cruciale dans la préparation des données, que ce soit lors de l'entrainement ou lors de l'inférence.

**Reconstruction sagittale**

Suite au recadrage, les données sont dédoublées en deux branches. Une branche subit la normalisation décrite à la section précédente pour être ensuite acheminée à l'entrée du réseau de segmentation axiale, tandis que l'autre branche subit autre série de transformations avant d'être acheminée à l'entrée du réseau de segmentation sagittale. La première de ces transformations suivant le recadrage est la reconstruction sagittale.

Suite au recadrage, les contours des images sont perdus. Or, étant donné que toutes les images sont recadrées de façon à ce que la moelle épinière soit centrée, l'information sur la position des coupes l'une par rapport à l'autre est détruite. Il est donc nécessaire lors du recadrage de conserver les informations sur les coordonnées originales de chaque coupe. Ainsi, il est possible par la suite de construire une représentation sagittale avec les images axiales recadrées.

Les images reconstruites suivent donc la forme sagittale de la moelle épinière et ont un arrière-plan noir, comme on peut le voir sur la figure 16. Le fait que l'arrière-plan soit uniformément noir aide grandement le réseau dans sa tâche de segmentation, car cet arrangement est une façon efficace de préserver la forme de la colonne vertébrale tout en éliminant les informations superflues. La préservation de la forme de la moelle épinière est un des facteurs clés du système développé ici, car c'est cet aspect qui permet au réseau d'apprendre les caractéristiques des liens entre les images axiales, ce qui est quelque chose qu'un réseau axial seul peine à accomplir.

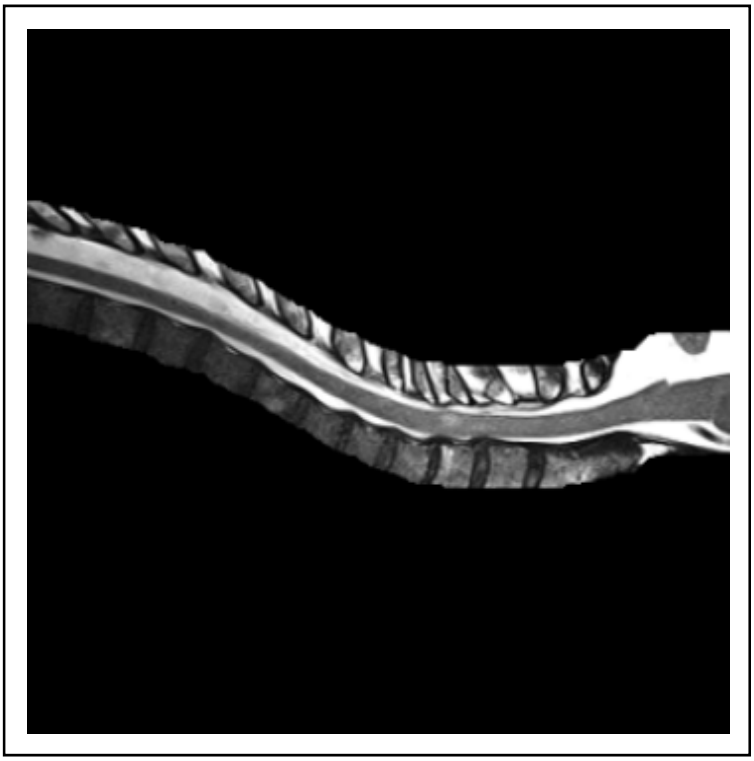


Figure-A I-6  Coupe sagittale reconstruite sur fond noir

**Rééchantillonnage sagittal**

Suivant la reconstruction, un rééchantillonnage sagittal est appliqué. À ce stade-ci, la reconstruction des coupes sagittales a ramené celles-ci à la résolution qu'elles avaient suite au premier rééchantillonnage. Or, cette résolution était de 5 mm dans le plan axial, ce qui n'implique aucune constance dans la résolution des images en termes de pixels, surtout dans le plan sagittal.

Le rééchantillonnage sagittal vise donc d'abord à redimensionner les coupes sagittales à une résolution standard afin que le réseau puisse fonctionner correctement. Puisque le premier rééchantillonnage avait tenu compte des différents paramètres reliés à la représentation dans l'espace et que l'on utilise ces images axiales rééchantillonnées pour construire les images sagittales, ces caractéristiques sont conservées ici. Il convient donc simplement de redimensionner les coupes sagittales à un certain nombre de pixels, établi à 384 × 384. Cette résolution correspond à la résolution originale minimale d'une IRM dans la base de données, et la majorité des IRM ont précisément cette résolution.

La structure U-Net s'est par le passé montré robuste à une grande variété de résolutions des données d'entrée. Il existe de nombreuses autres études présentant des résolutions d'entrée différentes. Dans le cadre de ce projet, la résolution choisie offre une altération minimale des grandeurs originales tout en préservant suffisamment de détails. Elle est donc un compromis entre la taille des images et la vitesse du réseau, car une résolution plus grande ralentirait certainement le réseau, mais ne serait pas nécessairement plus performante. Ce qui est plus, une résolution plus grande requiert plus de mémoire VRAM du GPU lors de l'entrainement, ce qui peut devenir problématique, surtout lorsque la taille de batch est élevée.

**Normalisation sagittale**

La normalisation des images sagittales se fait selon le même processus que la normalisation axiale, c'est-à-dire en ramenant d'abord les valeurs au-dessus du 98$^{e}$ percentile et en dessous du 2$^{e}$ percentile aux valeurs de ces percentiles respectivement, puis en appliquant une normalisation pour ramener toutes les valeurs entre 0 et 1. Les motifs derrière ce choix sont les mêmes qui ont été discutés lors de normalisation axiale. Cette étape représente l'étape finale du prétraitement avant que les données soient acheminées au réseau sagittal.

## ANNEXE II

## Apprentissage profond pour l'analyse d'image automatique

### Méthodologie et données

L'apprentissage profond et l'apprentissage machine en général visent à entrainer des réseaux capables de détecter des patrons dans les données d'entrée qu'on leur fourni et tirer des conclusions par rapport à ces patrons. La plupart des applications modernes d'apprentissage machine font partie de ce que l'on appelle l'apprentissage *supervisé*. Dans ce type d'apprentissage, les données d'entrée sont pré-catégorisées par les développeurs, et le réseau d'apprentissage est entrainé à partir de ces données à reconnaitre les catégories prédéfinies. Pour arriver à ceci, le réseau doit bâtir sa propre image ou représentation numérique de ce que représente chaque catégorie. Or, c'est lors de cette tâche que l'apprentissage profond se distingue de l'apprentissage machine plus superficiel (Lundervold et Lundervold, 2019). Dans ce dernier, on doit spécifier au réseau quoi regarder afin de bâtir la représentation, par exemple, la différence entre les contrastes d'une image ou la moyenne des pixels, tandis que dans l'apprentissage profond, c'est le réseau qui apprend seul quelles caractéristiques sont importantes. On appel cette fonctionnalité de l'apprentissage profond *feature learning*. Ceci est réalisé durant la *phase d'entrainement*, au cours de laquelle le réseau performe une série d'opérations destinées à ajuster les *poids* de chaque *neurone* qui fait partie de son *architecture*.

Pour aider le réseau à accomplir ceci, les développeurs séparent habituellement les données d'entrée en trois catégories, soit les données *d'entrainement,* de *validation* et de *test*. Les données d'entrainement servent à l'apprentissage proprement dit du réseau, avec lesquelles le réseau bâti son propre modèle de chaque catégorie. Les données de validation servent dans le processus d'entrainement dans la mesure où, à différents moments durant le processus de conception et d'entrainement d'un réseau, les concepteurs doivent avoir une idée de la précision du réseau afin d'ajuster les différents hyperparamètres. Ils se servent donc des données de validation afin d'obtenir une telle approximation. Les données de test sont réservées à la validation du réseau et ne sont utilisées qu'une fois que celui-ci est finalisé. Elles ne font donc pas partie du processus de construction du réseau à proprement parler. Elles sont

cependant nécessaires pour tirer des conclusions sur la validité du réseau. Ces données sont souvent mises de côté par les développeurs jusqu'à la toute fin afin de ne pas interférer avec leur objectivité. Elles proviennent aussi idéalement d'une ou plusieurs sources différentes des autres données afin de fournir l'évaluation la plus compréhensive possible des capacités réelles du réseau.

**Réseaux de neurones**

La grande majorité des réseaux d'apprentissage machine aujourd'hui utilisent une architecture basée sur les *réseaux de neurones*. Ces neurones sont arrangés en *couches* et sont interconnectées entres eux. Le nombre de couches et la complexité du réseau est ce qui sépare l'apprentissage profond de l'apprentissage machine. En effet, on dit en général d'un réseau qu'il est plus profond s'il a plus de couches. Le rôle d'un neurone est de prendre le signal d'entrée, lui appliquer certaines transformations, et puis relayer le signal à la prochaine couche.

Une des architectures de réseau classiques est le *feedforward neural network*. La plupart des architectures modernes d'apprentissage profond sont au moins partiellement inspirées de ce modèle, dont les bases ont été développées dans les années 1950 par Frank Rosenblatt (Rosenblatt F, 1958). Dans ce type de réseau, chaque neurone de chaque couche reçoit en entrée chaque signal de chaque neurone de la couche précédente, tel qu'illustré à la Figure-A II-1. Le neurone fait la sommation de tous ses signaux d'entrée et les multiplie par une valeur qu'on appelle un *poids*. Chaque neurone possède un poids qui est indépendant du poids des autres neurones. Ce sont ces poids qui sont ajustés lors de l'entrainement du réseau. Une fois cette multiplication effectuée, une constante appelée le *biais* est typiquement d'additionnée au résultat de la multiplication. Finalement, avant d'être acheminé au prochain neurone, le signal est soumis à une *fonction d'activation*. Cette fonction est typiquement une fonction non-linéaire comme la fonction *ReLU*, qui remet simplement les valeurs négatives des signaux à zéro et n'affecte pas les autres valeurs.

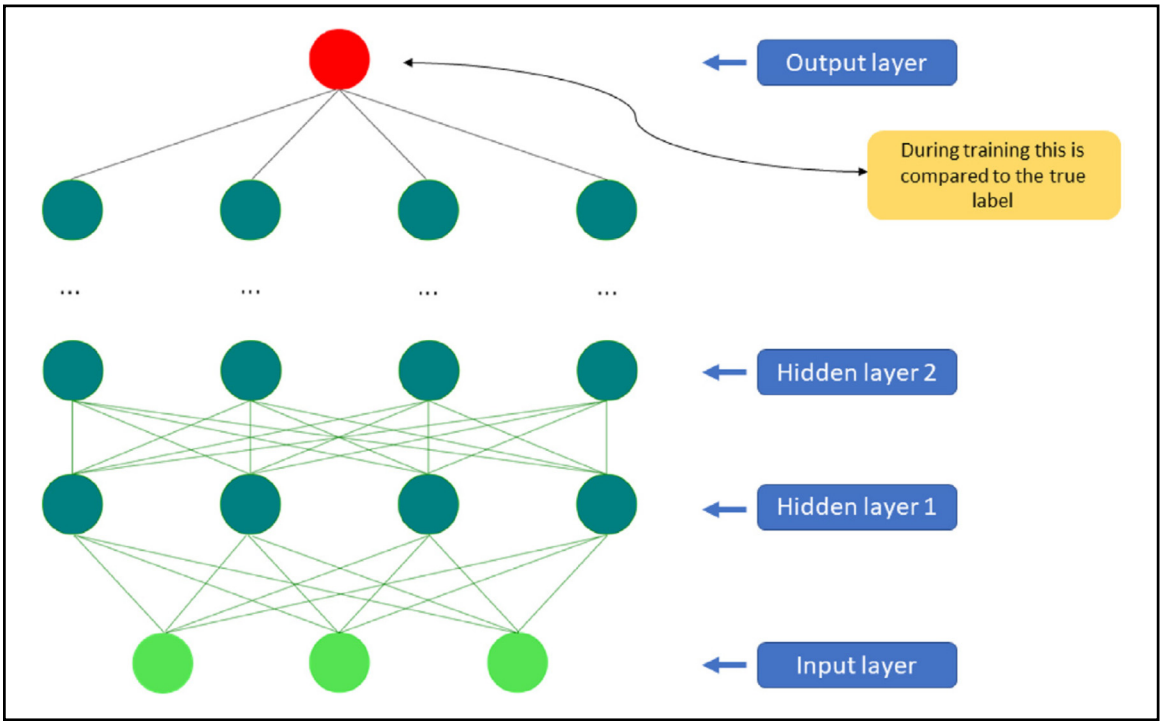



Figure-A II-1 Réseau de neurones feedforward
Tirée de Lundervold et Lundervold, 2019

Lors d'une itération d'entrainement typique, une *batch* de données est d'abord acheminée à la *couche d'entrée*. La taille de cette batch est un paramètre ajustable du réseau appelé *batch size*. Ces données sont ensuite propagées à travers toutes les *couches cachées* du réseau suivant le processus décrit au paragraphe précédent, jusqu'à leur arrivée à la *couche de sortie.* Cette propagation à travers le réseau est appelée la *propagation avant*. Une fois les signaux arrivés à la couche de sortie, leurs valeurs sont comparées aux valeurs de référence à l'aide d'une *fonction objective* ou *fonction de perte*. La dérivée de cette fonction par rapport au signal de sortie s'appelle le *gradient*. Ce gradient est ensuite propagé à l'inverse à travers le réseau dans un processus qu'on appelle la *propagation arrière*. Lors de cette propagation, le gradient de la fonction objective est recalculé par rapport au poids de chaque neurone afin d'ajuster ces poids pour minimiser l'erreur à la sortie du réseau. Les poids des neurones sont ajustés avec des incréments plus ou moins grands dépendant du *taux d'apprentissage* et de l'*optimisateur* choisis par les concepteurs du réseau.

Une fois qu'une batch a terminé sa propagation avant et que la propagation arrière résultante est aussi terminée, une autre batch est envoyée et le cycle se répète. Une fois que toutes les données ont été envoyées, on dit qu'une *époque* est terminée. La durée totale de l'entrainement est donc mesurée en époques. Le nombre d'époques utilisé pour l'entrainement d'un réseau est crucial, car un surentrainement peut mener à ce qu'on appel du *overfitting*, où le réseau apprend trop en détail les traits particuliers des données d'entrainement et de validation, résultant en un réseau qui est très performant pour ces données, mais peu performant sur des nouvelles données lorsque l'entrainement est terminé. Et à l'inverse, si le réseau n'est pas entrainé suffisamment, sa performance ne sera pas aussi optimale qu'elle pourrait l'être, résultant en ce qu'on appelle du *underfitting*.

**Réseaux convolutifs**

La plupart des applications d'apprentissage profond en imagerie médicale utilisent des réseaux de neurones convolutifs (Lundervold et Lundervold, 2019). Bien que d'autres architectures comme les réseaux *feedforward* discutés plus haut puissent aussi apprendre les caractéristiques clés des images, les réseaux convolutifs sont habituellement plus performant dans le domaine de la vision par ordinateur. Les réseaux convolutifs (CNN) n'ont pas une connexion entre chaque neurone et ceux de la couche précédente comme les réseaux feedforward. Au lieu, ces réseaux sont interconnectés de façon à minimiser les connexions entre les couches de neurones tout en optimisant pour préserver les liens supportant les relations spatiales dans les données (Lundervold et Lundervold, 2019). Cette organisation de l'architecture du réseau est construite en prenant en compte la structure des images et la façon dont les caractéristiques clés de celles-ci sont habituellement bâties. Cette optimisation des connexions ainsi que plusieurs autres mécanismes font en sorte que les CNN peuvent avoir beaucoup plus de couches que les réseaux classiques, ce qui augmente drastiquement leur performance face à plusieurs tâches.

Dans un CNN, les données présentées à la couche d'entrée sont organisées en tenseurs, qui sont essentiellement des tableaux de données pouvant avoir plusieurs dimensions, cependant, tous les éléments d'une dimension doivent être de longueur égale, et un tenseur ne peut contenir qu'un seul type de données. De plus, les tenseurs sont immuables, voulant dire qu'on ne peut

pas les modifier sans créer une copie. Les tenseurs sont surtout importants du point de vue programmation, car les librairies et frameworks contiennent plusieurs opérations et règlements prédéfinis par rapport à ceux-ci, simplifiant ainsi le processus de développement par rapport à l'utilisation de tableaux traditionnels. Ils sont aussi importants au niveau *hardware*, car leur structure de bas niveau leur permet d'être exécutés et gardés en mémoire dans des processeurs graphiques (GPU), lesquels sont utilisés pour entrainer les réseaux de neurones.

**Couches de base**

Les tenseurs d'entrée sont présentés à la couche d'entrée, qui est habituellement une couche convolutive. Dans une couche convolutive, les données d'entrée sont d'abord convoluées avec ce qu'on appelle les *filtres*. Les filtres sont des tenseurs de données dont les valeurs servent la fonction de *poids* dans le réseau. Ce sont donc ces valeurs qui sont ajustées durant l'entrainement du réseau. Le processus de convolution entre les tenseurs d'entrée et les filtres prend la forme d'une multiplication matricielle, tel qu'illustré à la Figure-A II-2. La dimension du tenseur résultant dépend de la *stride*, qui est le pas avec lequel la section du tenseur d'entrée qui est multipliée se déplace. Donc, plus petite est la stride, plus grand sera le tenseur résultant de la convolution. Un autre paramètre de la convolution est le *padding*. Le padding est le nombre de couches de zéros entourant le tenseur d'entrée. Il sert à augmenter la quantité d'information retenue des rebords du tenseur, qui serait autrement moins grande que la quantité retenue du milieu. Il sert aussi à augmenter la dimension du tenseur résultant de la convolution, car autrement, celui-ci deviendrait de plus en plus petit à chaque couche convolutive du réseau.

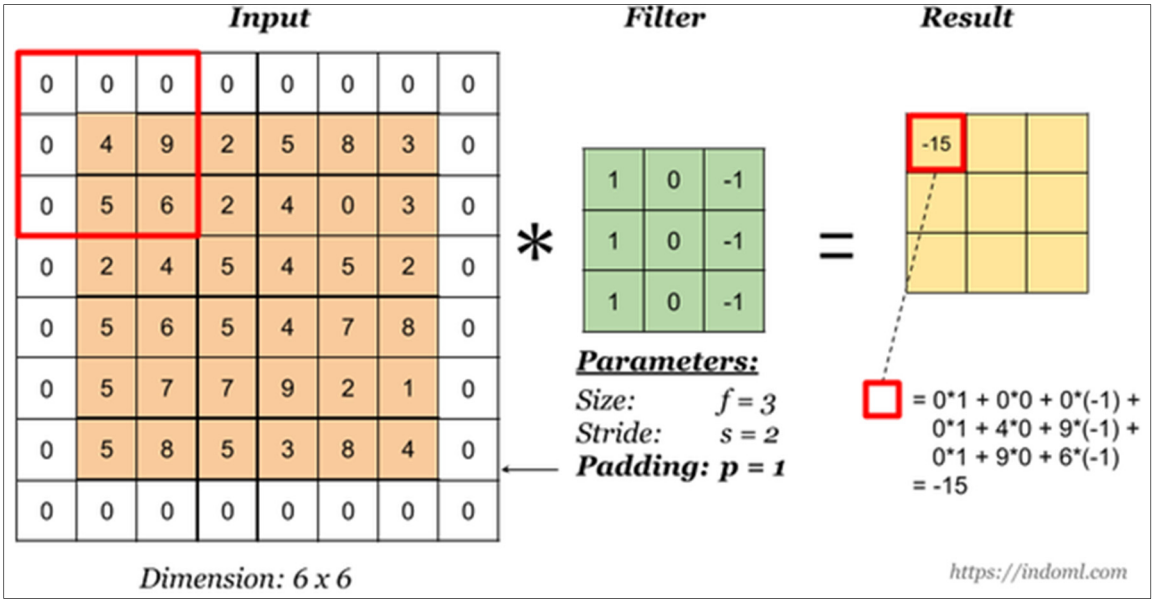



Figure-A II-2 Opération de base d'une convolution dans un CNN
Tirée de Indoml, 2021

Suite à la convolution en tant que telle, une activation est habituellement appliquée au résultat de la convolution. Comme discuté plus haut, cette activation prend la forme d'une fonction non-linéaire telle que la fonction ReLU. Ensuite, un biais est souvent ajouté au tenseur avant de l'envoyer vers la prochaine couche. Le processus sommaire d'une couche convolutive peut être visualisé à la figure x. Le ou les tenseurs résultant d'une couche convolutive s'appellent des *feature maps*. Cependant, il faut faire attention car on peut aussi utiliser ce terme pour désigner le produit de la convolution avant l'activation. Dans ce cas, l'activation est souvent référencée comme une couche indépendante de la couche convolutive.

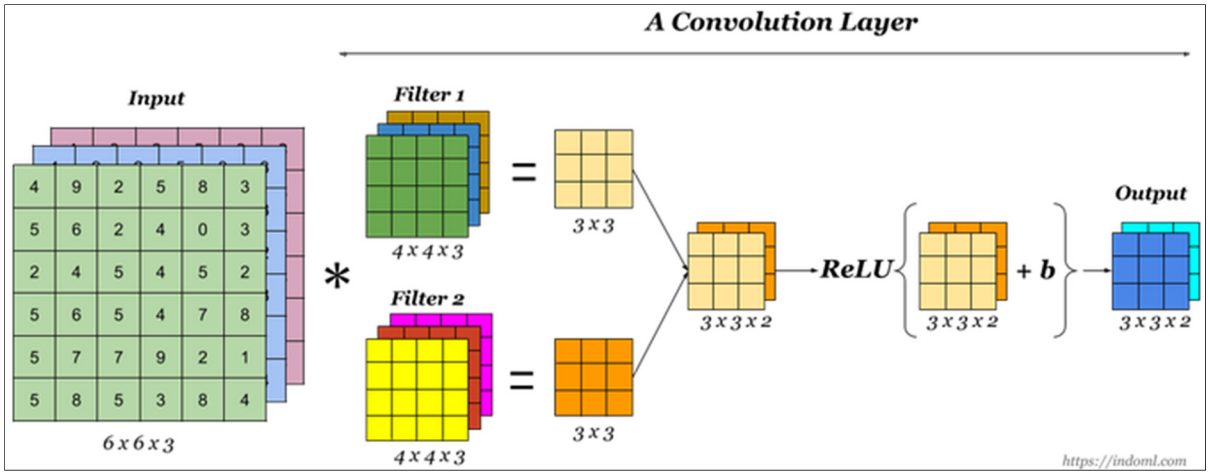



Figure-A II-3 Couche convolutive. Tirée de Indoml, 2021

On retrouve périodiquement entre les couches convolutives des couches de *pooling*. Ces couches réduisent la taille des tenseurs d'entrée en produisant un seul chiffre pour une région donnée de ces tenseurs. La technique utilisée le plus souvent est de simplement prendre le chiffre le plus élevé de la région du tenseur inspectée. On peut aussi faire la moyenne des chiffres dans cette région. Ces processus peuvent être visualisés à la Figure-A II-4. En plus de réduire la taille des données, le pooling sert à réduire les calculs et peut mettre en valeur certaines caractéristiques détectées par le réseau. Certains réseaux utilisent une *stride* plus élevée lors des convolutions afin de réduire la taille des données au lieu d'utiliser le pooling.

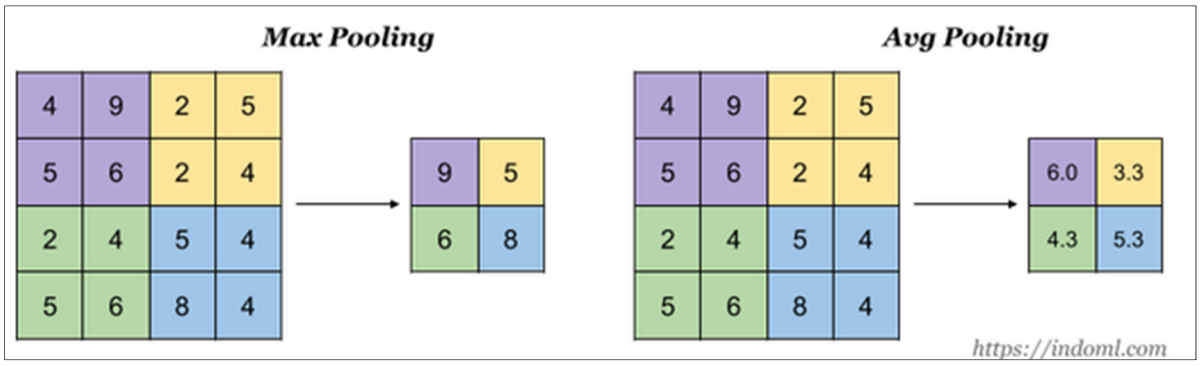



Figure-A II-4 Visualisation des opération max pooling et average pooling
Tirée de Indoml, 2021

Donc, les réseaux convolutifs classiques sont constitués d'une alternance de couches convolutives et de couches de pooling. Ils se terminent habituellement par une ou plusieurs couches qui sont pleinement connectées entres elles comme dans les réseaux *feedforward*.

**Stratégies additionnelles**

Les réseaux convolutifs se sont beaucoup développés dans les dernières années, et il existe maintenant plusieurs techniques qui viennent suppléer l'architecture de base des CNN. On retrouve parmi celles-ci la *dropout regularization* (Srivastava et al., 2014). Cette technique consiste à n'utiliser qu'une partie des neurones du réseau pour chaque époque d'entrainement. Chaque neurone se voit donc attribué un statut actif ou inactif pour chaque époque d'entrainement. Cette attribution se fait selon le coefficient de dropout. Par exemple, si ce coefficient est égal à 0.5, la moitié des neurones seront aléatoirement désactivées à chaque époque. Il est à noter que ce tirage aléatoire est repris à chaque époque, de façon à ce que le réseau soit différent à chacune de celles-ci. Cette stratégie a pour effet de réduire le *overfitting* en réduisant la dépendance entre les neurones. Elle est aujourd'hui très utilisée et apporte normalement un gain important dans la performance des réseaux (Lundervold et Lundervold, 2019).

Une autre technique qui est très répandue aujourd'hui est la *batch normalization* (Ioff et Szegedy, 2015). Dans cette technique, certaines entrées de certaines couches du modèle subissent un traitement mathématique visant à standardiser les données. Cette standardisation se fait habituellement en soustrayant la moyenne et en divisant par l'écart type pour chaque batch de données (Lundervold et Lundervold, 2019). Elle est le plus souvent performée après l'activation de la couche précédente, mais peut aussi être performée avant dans les cas où la fonction d'activation choisie retourne des résultats trop loin d'une distribution normale (Ioff et Szegedy, 2015). Cette technique aide à entrainer les réseaux en réduisant le *internal covariate shift*, qui est le terme pour exprimer un phénomène perturbateur qui a lieu durant la phase de propagation arrière où l'ajustement des poids est opéré. Cet effet résulte du fait que les poids sont ajustés en ordre inverse du réseau, de la sortie vers l'entrée, ce qui implique que lorsque l'ajustement d'un poids est réalisé, cet ajustement tient pour acquis que tous les poids qui le

précèdent dans le réseau ne changeront pas. Mais dans la pratique, ces poids changent, vu que l'ajustement a lieu en sens inverse. Or, le désaccord qui résulte de cet arrangement s'appelle le *internal covariate shift*. Donc, le principe de la batch normalization est que, en standardisant périodiquement les données, la dépendance entre les poids des différentes couches se trouve réduite, ce qui améliore la vitesse d'entrainement et réduit la sensibilité du réseau à l'initialisation des poids (Ioff et Szegedy, 2015). De plus, cette technique peut dans certains cas annuler le besoin d'utiliser la *dropout regularization* (Ioff et Szegedy, 2015).

**Optimisateur**

La grande majorité des publications récentes en apprentissage profond utilisent un optimisateur de type *descente de gradient.* Ce type d'optimisateur est basé sur des gradients. Chaque poids ou paramètre dans un réseau a un gradient, qui correspond à sa dérivée partielle par rapport au résultat de la fonction de perte. Lors de la descente de gradient, on cherche à minimiser la fonction de perte, donc si un paramètre a un gradient négatif, on veut augmenter la valeur de ce paramètre, et s'il a un gradient positif, on veut réduire sa valeur. Ceci est le principe de base d'un optimisateur à descente de gradient. Au fil du temps, plusieurs variations ont évolué afin d'optimiser la performance et la précision de la descente de gradient.

**Descente de gradient stochastique**

Une des premières méthodes développées pour optimiser le processus de descende de gradient est la méthode stochastique avec momentum. Dans cette méthode, qui est encore très utilisée aujourd'hui, les données pour l'entrainement sont mélangées à chaque époque et puis séparées en mini-batch. Les éléments de chaque mini-batch sont ensuite propagés vers l'avant à travers le réseau, et puis, une fois que tous les éléments d'une batch ont été propagés, leur résultat moyen est calculé et propagé vers l'arrière à travers une descente de gradient.

Dans cette méthode, le momentum agit de façon semblable au momentum dans la normalisation de batch, c'est-à-dire qu'il détermine à quel point la batch précédente a un impact sur la batch actuelle. Les formules pour la mise à jour des paramètres lors de la descente de gradient stochastique avec momentum peuvent être écrites sous la forme suivante :

$$v_{act} = \mu * v_{prec} + g_{act} \quad (3.1)$$

$$p_{act} = p_{prec} - lr * v_{act} \quad (3.2)$$

$p = paramètre$
$\mu = momentum$
$g = gradient$
$v = vélocité$
$lr = taux\ d'apprentissage$

Il est à noter que, bien que le principe demeure semblable, certains détails peuvent changer dans l'implémentation des différents frameworks. Les équations plus haut sont tirées du framework PyTorch, qui est le framework utilisé dans la présente étude. On voit par ces équations que le momentum détermine quelle proportion de la vélocité précédente est additionnée au gradient actuel, qui détermine à son tour quelle valeur soustraire au paramètre en question. On peut aussi observer que le momentum aura pour effet d'augmenter le changement appliqué au paramètre si la vélocité précédente est du même signe que le gradient actuel, ou bien de réduire ce changement si la vélocité précédente est du signe contraire au gradient actuel. Le momentum a donc pour effet de réduire les oscillations lors de l'entrainement ainsi que d'augmenter la vitesse de convergence vers un optimum local.

**Adaptation du domaine**

L'adaptation du domaine est une sous-branche de l'apprentissage par transfert. L'apprentissage par transfert englobe plusieurs types de divergences entre les domaines et tâches sources et cibles. Par exemple, l'apprentissage par transfert inductif vise à adapter un réseau dont les données sources (données pour l'entrainement initial) sont très semblables aux données cibles (données pour le nouvel entrainement), mais qu'on veut adapter pour performer une nouvelle tâche. De son côté, l'adaptation du domaine représente le cas de l'apprentissage par transfert où l'on veut adapter un réseau avec des données cibles différentes des données sources, mais pour accomplir la même tâche que le réseau initial (Wang et Deng, 2018). Il est à noter que certains auteurs présentent une autre définition de l'adaptation du domaine, qui sera discuté à la section suivante.

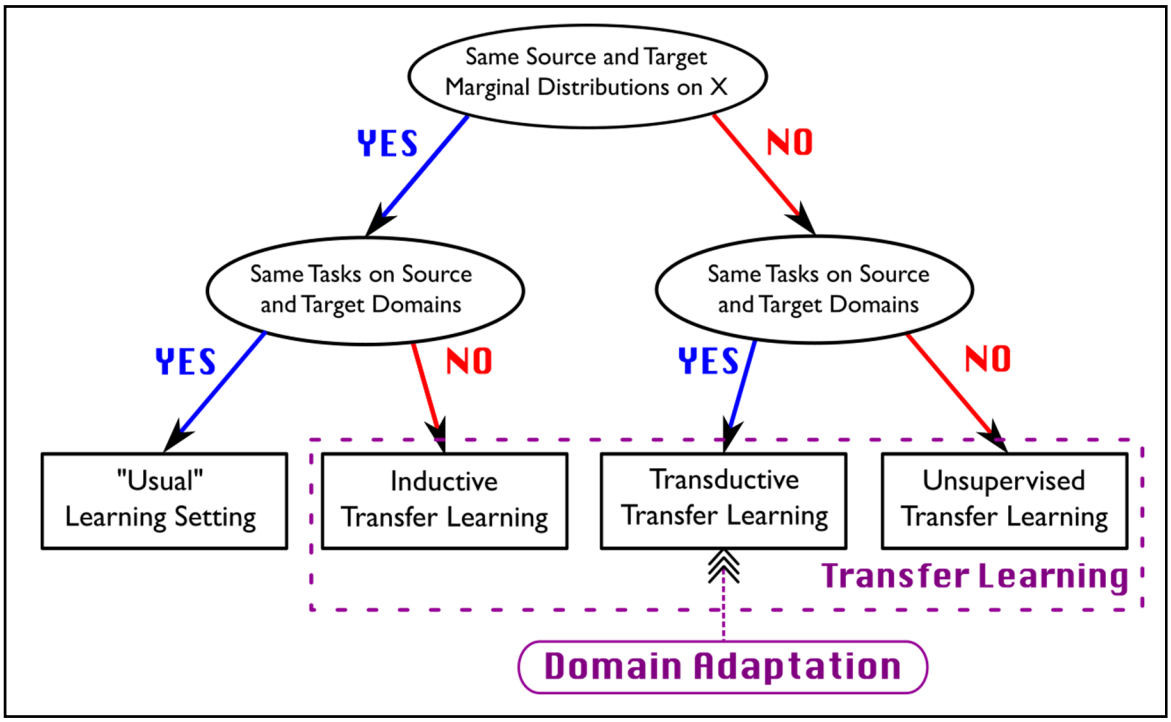



Figure-A II-5 Différents types d'apprentissage par transfert
Tirée de Emilie Morvant, 2017

Il existe plusieurs techniques d'adaptation du domaine en apprentissage profond. Certaines techniques sont dites supervisées, indiquant que l'adaptation se fait à l'aide de données cibles annotées. D'autres techniques sont dites semi-supervisées, indiquant qu'elles sont réalisées avec un mélange de données cibles annotées et non-annotées, et d'autres techniques sont non-supervisées, voulant dire qu'elles sont réalisées seulement avec des données cibles non-annotées.

Certaines approches visent à apporter des changements ciblés dans l'architecture d'un réseau afin d'adapter celui-ci à de nouvelles données. On retrouve parmi celles-ci des approches qui sont basées sur une normalization de batch adaptative (Li et al., 2016), des approches qui sont basées sur deux réseaux dont les poids sont reliés mais non partagés (Rozantsev et al., 2019), ainsi que des approches basées sur un dropout conçu spécifiquement pour adapter un réseau à d'autres domaines (Xiao et al., 2016). D'autres approches se basent sur une adaptation statistique afin d'aligner la distribution du domaine source et du domaine cible. Les méthodes les plus utilisées pour accomplir ceci sont le *maximum mean discrepancy* (Yan et al., 2017),

l'alignement corrélatif (Sun et al., 2016), et la divergence Kullback-Leibler (Zhuang et al., 2015), ainsi que d'autres méthodes (Wang et Deng, 2018). Il existe aussi des approches d'adaptation du domaine se basant sur un rapprochement des domaines en utilisant leur propriétés géométriques (Chopra et al., 2013). Parmi les approches citées ici, la majorité appartiennent à la catégorie des approches non-supervisées.

La majorité des approches pour l'adaptation d'un domaine sont cependant supervisées. Ces techniques ont comme désavantage de requérir l'annotation manuelle des données, cependant, elles sont très efficaces et fonctionnent presque toujours avec succès (Wang et Deng, 2018). Les méthodes basées sur ce type d'approche suivent habituellement un processus de base semblable qui est souvent appelé *fine-tuning*. Dans ce processus, un réseau accomplissant le même type de tâche que l'on veut accomplir (ex. catégorisation, segmentation, etc.) est d'abord choisi. Idéalement, ce réseau a été entrainé avec une grande base de données, car ceci fait en sorte que les caractéristiques apprises par le réseau sont plus générales et peuvent donc s'adapter plus efficacement à un nouveau domaine. À l'opposé, si la base de données utilisée pour entrainer le réseau original était trop petite, les caractéristiques apprises seront trop particulières à l'échantillon original pour être généralisables à un nouvel échantillon.

Ayant choisi un réseau de base adéquat, la première étape est de remplacer la couche de sortie de ce réseau avec une ou plusieurs couches adaptées au nouveau domaine afin d'avoir le bon format de résultats en sortie. Ensuite, on gèle les poids d'un certain nombre de couches à partir de l'entrée du réseau de façon à ce qu'ils ne soient pas modifiés lors de l'entrainement à venir. Le nombre de couches à geler est variable et dépend, entre autres, de la similarité des domaines source et cible et de la taille des bases de données source et cible (Chu et al., 2016). L'acte de geler les premières couches du réseau est motivé par le fait que les réseaux en général ont tendance à apprendre des caractéristiques de plus en plus spécifiques à leur domaine dans les couchent qui s'approchent de la sortie. Les caractéristiques les plus générales se trouvent donc dans les premières couches, et ce sont ces caractéristiques que l'on peut étendre à d'autres domaines. Voilà pourquoi on gèle ces couches et pourquoi le nombre de couches à geler est relié à la similarité des domaines source et cible. Les couches à geler sont aussi déterminées

par la quantité de données cibles annotées qui est disponible pour le réentrainement. Dans le cas où cette quantité n'est pas très grande, on voudra geler plus de couches pour éviter de faire du *overfitting*. L'étape finale du fine-tuning consiste à réentraîner le réseau modifié avec la base de données cible en n'ajustant que les poids des couches que l'on a choisi de ne pas geler.

L'étude (Yosinki et al., 2014) peut aider à comprendre ce qui se passe lors du processus de fine-tuning. Dans cette étude, les chercheurs ont entrainé deux réseaux aux architectures identiques en séparant la base de données ImageNet en deux. Ils ont nommé ces réseaux A et B respectivement. Les chercheurs font l'expérience de sectionner les réseaux à différentes couches et réinitialiser les poids des couches sectionnées à des valeurs aléatoires. Puis, ils réentraînent les réseaux modifiés comme pour faire du fine-tuning. Comme on peut le voir sur la Figure-A II-6, les résultats obtenus pour les réseaux réentraînés en ne gelant pas les poids des couches non-réinitialisées sont significativement supérieurs.

Cette figure nous donne aussi d'autres informations sur ce qui se passe lors du réentrainement des réseaux. On peut voir sur la courbe (2) que lorsque le réseau B est sectionné à partir de la couche trois ou quatre et réentraîné avec les mêmes données B en gardant les couches préliminaires gelées, sa performance va en diminuant. Selon les auteurs de l'article, cette diminution est due au fait qu'il existait entre les couches du réseau original des relations complexes que le nouveau réseau est incapable de reproduire, même si on l'entraine avec les mêmes données. Ils nomment ces relations des *co-adapted features*. On peut observer sur la courbe (3) que lorsque les couches préliminaires ne sont pas gelées mais réentraînées elles aussi, cette perte de performance est évitée et on retrouve les niveaux originaux. On peut aussi observer sur la courbe (4) une diminution de performance supplémentaire. Cette diminution va en augmentant à mesure que le réseau est sectionné à une couche plus éloignée de l'entrée. Selon les auteurs, ceci est dû au fait que l'on conserve des couches qui reconnaissent des caractéristiques de plus en plus spécifiques et qui viennent donc nuire à la reconnaissance de la nouvelle base de données. Il est intéressant de noter que sur la courbe (5), lorsque les couches non-sectionnées ne sont pas gelées comme sur la courbe (4), les deux facteurs qui causaient

les pertes de performance sont évités et le fait de garder plus de couches du réseau original semble augmenter la performance.

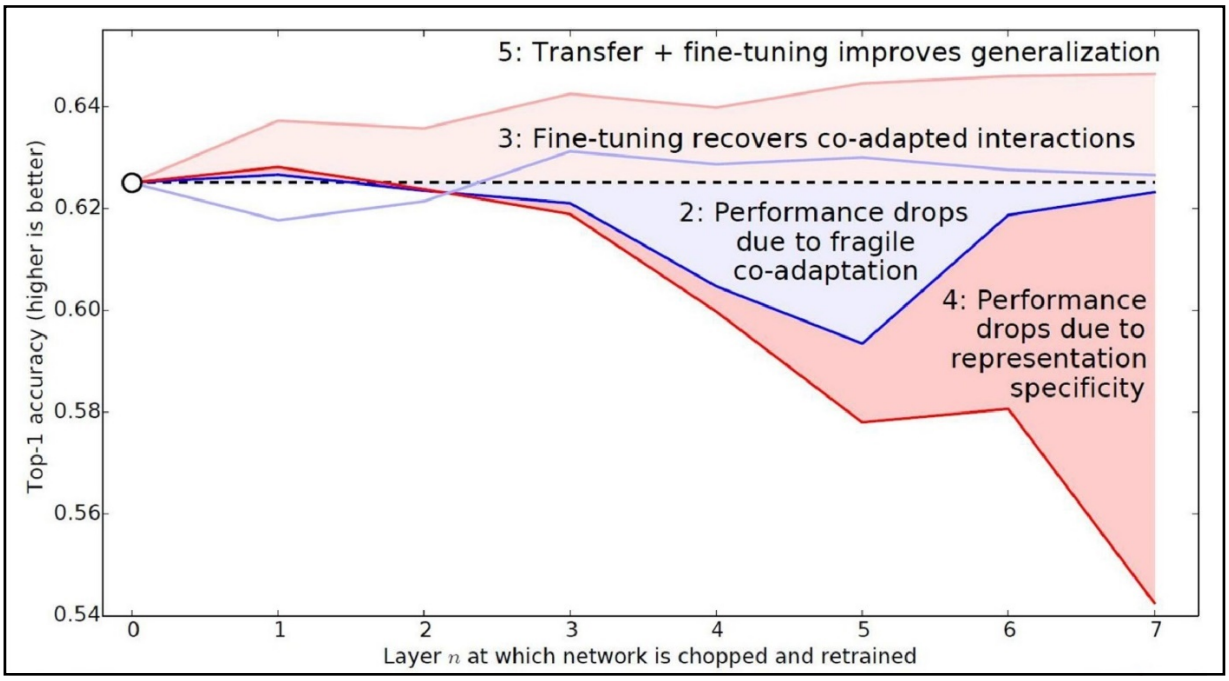



Figure-A II-6 La ligne pointillée montre la ligne contrôle, où le réseau B est entrainé avec la base de données B. (2) BnB: montre la performance du réseau B lorsqu'il sectionné à une couche *n* et réentraîné avec la même base de données B en gardant les couches d'entrée gelées. (3) BnB+: montre la performance du réseau B lorsqu'il subit le même traitement que BnB mais que les couches d'entrée ne sont pas gelées mais entrainées aussi. (4) AnB : montre la performance du réseau A lorsqu'il sectionné à une couche *n* et réentraîné avec la base de données B en gardant les couches d'entrée gelées. (5) AnB+ : montre la performance du réseau A lorsqu'il subit le même traitement que AnB mais que les couches d'entrée ne sont pas gelées mais entrainées aussi avec la base de données B
Tirée de Yosinki et al., 2014

## ANNEXE III

### Métriques pour la validation des approches

Une des métriques préexistantes utilisée est le coefficient Dice, qui a développé dans le années 1940 pour estimer la similarité entre deux échantillons. Ce coefficient est bien adapté pour les cas où les classes d'une image sont débalancées (Sudre et al., 2017; Drozdzal et al., 2018; Perone et al., 2018), ce qui est fréquemment le cas pour les images médicales. Le volume de la moelle épinière sur nos IRM étant relativement petit par rapport au volume total des IRM, nos images ne font pas exception à cette tendance. Le coefficient Dice utilisé lors de cette étude peut être formulé de la façon suivante :

$$C_{dice} = \frac{2 * \sum_{i=0}^{L \times H} s_i * m_i}{\sum_{i=0}^{L \times H} s_i + m_i} \tag{3.3}$$

Où $s$ correspond à la segmentation, $m$ correspond au masque, et $L$ et $H$ correspondent à la longueur et la hauteur en pixels d'une coupe IRM.

Une autre métrique utilisée est le coefficient Jaccard, aussi connu comme *intersection over union* (IoU). Ce coefficient est semblable au coefficient Dice puisqu'il est lui aussi basé sur une mesure du chevauchement entre la segmentation et le masque. Il peut s'écrire de la façon suivante :

$$C_{jac} = \frac{\sum_{i=0}^{L \times H} (s_i * m_i)}{\sum_{i=0}^{L \times H} (s_i + m_i) - \sum_{i=0}^{L \times H} (s_i * m_i)} \tag{3.4}$$

À l'aide de cette équation et de l'équation du coefficient Dice, on peut déduire que le coefficient Jaccard pour une coupe deux-dimensionnelle donnée aura toujours une valeur inférieure au coefficient Dice, mais supérieure à la moitié de celui-ci :

$$\frac{C_{dice}}{2} \leq C_{jac} \leq C_{dice} \tag{3.5}$$

Le coefficient Jaccard est donc corrélé positivement avec le coefficient Dice, ce qui fait en sorte que, lors de la comparaison de deux segmentations à un masque, les deux coefficients sont toujours en accord quant à la meilleure segmentation. Cependant, puisque le coefficient Jaccard punit plus sévèrement les mauvaises segmentations, il peut se trouver que, lorsqu'on calcule la moyenne des deux coefficients sur plusieurs images, le coefficient Jaccard et le coefficient Dice ne soient pas en accord. Donc, puisque nos résultats sont calculés en faisant la moyenne de plusieurs images, il demeure pertinent d'utiliser les deux métriques en concert, puisqu'elles ne rendront pas nécessairement des résultats homogènes.

La troisième métrique utilisée est le taux de vrais positifs (TPR). Cette métrique statistique est communément utilisée pour mesurer le degré de précision des segmentations (Lemay et al., 2021; Prados et al., 2017). Elle peut s'écrire sous la forme :

$$TPR = \frac{TP}{TP + FP} \tag{3.6}$$

Où $TP$ représente le nombre de pixels dans la segmentation qui sont des vrais positifs, tandis que $FP$ représente le nombre de pixels qui sont des faux positifs. La métrique finale indique donc la proportion des pixels segmentés par le réseau qui sont aussi segmentés sur le masque.

La métrique préexistante finale utilisée est la distance Hausdorff, qui est une métrique mesurant la distance euclidienne maximale entre n'importe quel point d'un ensemble et le point le plus près de celui-ci dans un autre ensemble (Rote, 1991; Prados et al., 2017). Lors du calcul de cette métrique, les deux ensembles sont évalués tour-à-tour par rapport à l'autre ensemble et seulement la plus élevée des deux valeurs maximales trouvées est conservée. La distance Hausdorff est fréquemment utilisée dans la littérature pour évaluer la qualité des segmentations (Prados et al., 2017;  Taha et Hanbury, 2015; Paugam et al., 2019).

Parmi les quatre métriques choisies, on retrouve donc deux métriques de chevauchement, une métrique statistique et une métrique de distance, ce qui couvre bien les trois types de métrique existant pour la segmentation.

# ANNEXE IV

## Comparaison de Deepseg 2D et Deepseg 3D

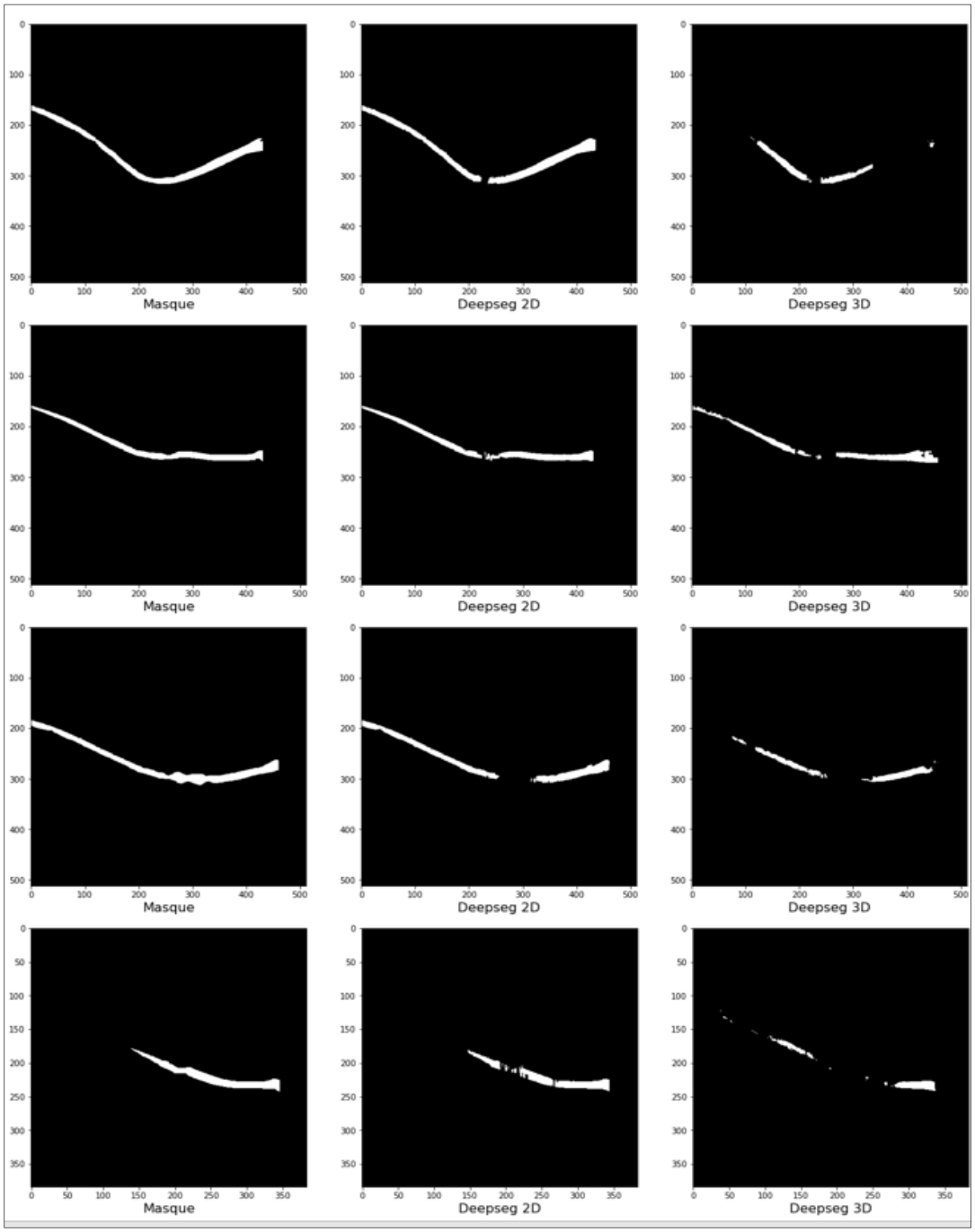



Figure-A IV-1 Comparaison des segmentations manuelles (gauche) avec les segmentations de Deepseg 2D (milieu) et Deepseg 3D (droite)

Le Tableau-A IV-1 montre les moyennes obtenues par les deux réseaux pour les cinq métriques qui sont présentées à l'ANNEXE III. Le réseau Deepseg 2D est hautement supérieur au réseau 3D sur l'ensemble des métriques. Il est donc évident que le réseau 2D est plus performant, du moins sur les IRM utilisées lors de cette étude. Voilà pourquoi c'est ce réseau qui a été sélectionné pour effectuer les comparaisons avec les réseaux élaborés au cours de ce projet.

Tableau-A IV-1 Moyennes obtenues sur les 20 IRM de test

| | **Deepseg 2D** | **Deepseg 3D** |
|---|---|---|
| **Coefficient Dice** | **0.879** | 0.717 |
| **Coefficient Jaccard** | **0.854** | 0.686 |
| **Taux vrai positif** | **0.886** | 0.477 |
| **Ligne centrale** | **0.303** | 2.178 |
| **Distance Hausdorff** | **2.010** | 3.298 |

# ANNEXE V

## Analyse de l'architecture U-Net

L'architecture du réseau convolutif utilisé pour analyser les coupes axiales et sagittales est basée sur l'architecture U-Net (Ronneberger et al., 2015). Ce type d'architecture est caractérisée par l'utilisation d'un chemin contractif à l'entrée du réseau, suivi d'un chemin expansif pour redimensionner les données à leur taille originale. Les deux chemins sont bâtis en couches, et la sortie de chaque couche contractive est concaténée à la couche correspondante du chemin expansif à travers des *skips connections*. Les couches contractives apprennent des caractéristiques plus générales que les couches expansives, mais de plus en plus spécifiques au fur et à mesure qu'elles deviennent plus profondes. Ainsi, le fait de concaténer leurs sorties au couches expansives combine les informations contextuelles des couches contractives aux informations plus spécifiques des couches expansives, permettant ainsi au réseau d'être guidé par les connaissances contextuelles afin pour diriger ses recherches de caractéristiques plus spécifiques (Ronneberger et al., 2015).

Les réseaux utilisés dans cette étude conservent donc ces principes architecturaux, mais modifient d'autres détails du réseau U-Net original.

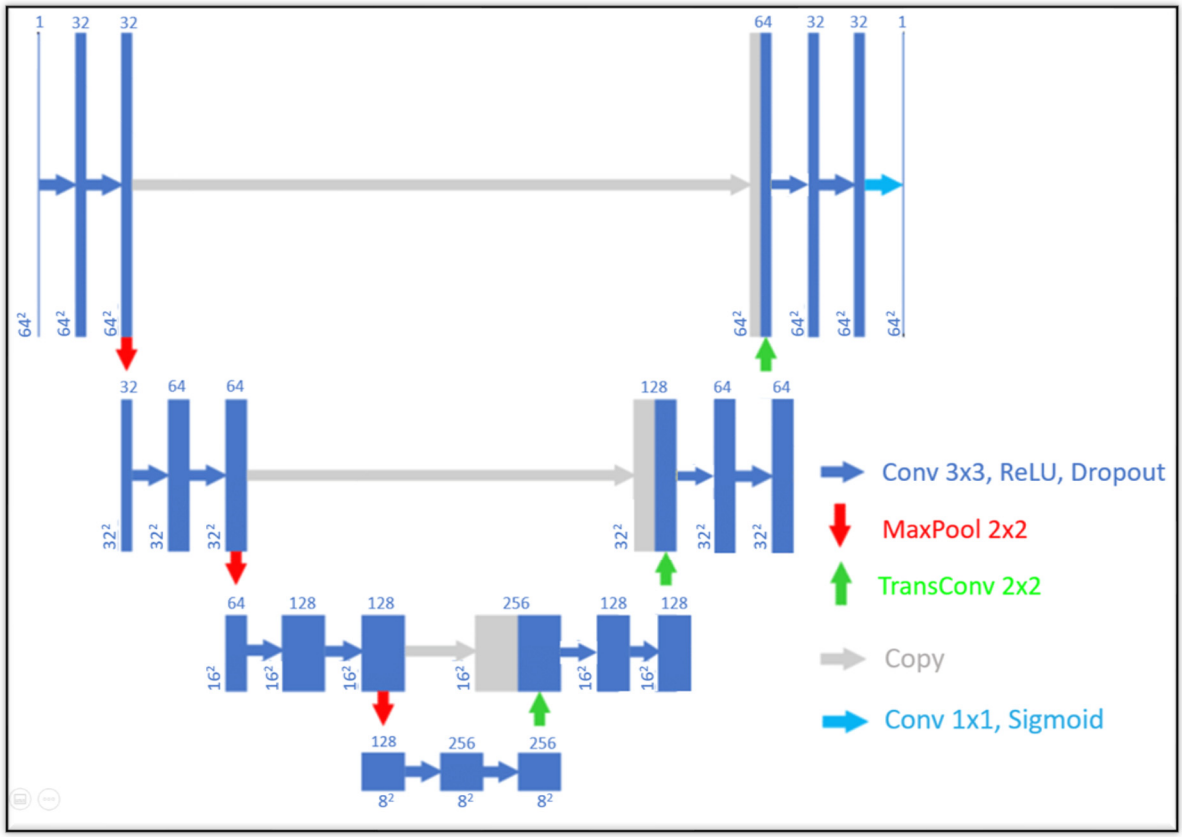



Figure-A V-1 Exemple d'un réseau U-Net modifié. Ce réseau comporte trois couches contractives et trois couches expansives

**Canaux**

Le nombre de canaux correspond au nombre de coupes deux dimensionnelles produites par les opérations d'un réseau à partir d'une coupe d'entrée. Dans le cas du réseau U-Net, les canaux augmentent au fur et à mesure que le réseau devient plus profond. Le nombre de canaux et leur taille respectives reflète la propension du réseau à étendre les données pour tenter d'en tirer davantage d'informations. C'est le nombre de canaux qui détermine le nombre de filtres convolutifs impliqués dans une convolution, et donc qui détermine aussi le nombre total de poids dans le réseau. Plus spécifiquement, le nombre de filtres requis pour une convolution donnée correspond au nombre de canaux à l'entrée de cette convolution, multiplié par le nombre de canaux à sa sortie. On comprend que, plus un réseau aura de canaux, plus il aura de poids à entrainer. Un nombre plus élevé de canaux ralenti donc non-seulement la vitesse d'un réseau, il rend aussi plus complexe l'ajustement optimal des poids lors de la propagation

arrière. La taille des canaux est un autre facteur qui influe sur la vitesse du réseau. Ce facteur n'influence pas le nombre de poids dans le système et n'a donc pas d'impact direct lors de la propagation arrière, cependant, il influe grandement sur le nombre de calculs requis lors de la propagation avant.

Rajouter des canaux à un réseau ou bien les agrandir est une façon simple d'ajouter de la complexité. Cependant, pour que le réseau en bénéficie, cette complexité doit être bien gérée, car un apport trop grand peut aussi bien être nuisible. Le fait d'ajouter des canaux à un réseau fonctionnel ou de les agrandir a fréquemment pour effet simplement d'ajouter une charge supplémentaire sans ajouter à la performance. L'ajustement du nombre de canaux requiert donc une stratégie. L'objectif est d'en inclure suffisamment pour optimiser la performance du réseau sans pour autant le ralentir inutilement ou nuire à sa précision.

La façon dont l'architecture U-Net est structurée facilite l'ajustement du nombre de canaux dans le réseau. En effet, vu son principe structural de proportionnalité, où les canaux sont multipliés par un coefficient à chaque couche du chemin contractif et réduites selon ce coefficient à chaque couche du chemin expansif, il suffit d'ajuster ce coefficient et le nombre de canaux produits lors de la première convolution pour régler tous les canaux du réseau. Par exemple, le fait de régler la première convolution pour produire 32 canaux et de régler le coefficient de proportionnalité à deux pour un réseau à quatre couches ferait en sorte que les couches du chemin contractif auraient respectivement 32, 64, 128 et 256 canaux.

De façon semblable au nombre de canaux, l'ajustement de la taille des canaux est lui aussi effectué en tenant compte de deux variables. La première variable correspond aux dimensions des images d'entrée, et la deuxième correspond au ratio qui règle le sous-échantillonnage lors des opérations de *max pooling* du chemin contractif ainsi que les convolutions vers le haut du chemin expansif. C'est ce ratio qui contrôle l'évolution de la taille des canaux à l'intérieur du réseau. Ce ratio est habituellement réglé pour réduire de moitié la taille des canaux à chaque couche du chemin contractif et la redoubler à chaque couche du chemin expansif afin de la ramener à sa valeur d'origine.

Ce ratio peut être contrôlé par les opérations de *max pooling* du chemin contractif. Ainsi, une *stride* et un *kernel* de 2 × 2 pour une opération de max pooling correspondent à un ratio de 0.5. Ceci correspond à la valeur la plus fréquemment utilisée dans la littérature ainsi que la valeur utilisée dans le réseau U-Net original. Des tests empiriques préliminaires indiquent que des valeurs plus élevées que 3 × 3 pour le kernel et la stride des opérations max pooling ne produisent pas de bons résultats. De plus, des valeurs non-symétriques telles que 2 × 3 ou 3 × 2 ne semblent pas produire de bons résultats.

**Profondeur**

Puisque la structure U-Net est bâtie en couches successives, il est possible de d'ajouter ou d'enlever des couches facilement. L'ajout et le retrait de couches influence évidemment la vitesse et la complexité du réseau. Comme discuté à la section précédente, la complexité peut aussi bien améliorer la performance du réseau que lui nuire, dépendant de comment elle est organisée. Le réseau U-Net original comprend cinq couches, cependant, beaucoup d'approches effectuées par la suite utilisent quatre couches ou moins (McCoy et al., 2019; Gros et al, 2019; Paugam et al., 2019; Siddique et al., 2021).

Selon (Paugam et al., 2019), l'inutilité potentielle des couches plus profondes est dû à la détérioration excessive de leur résolution. En effet, la structure U-Net est basée sur cette détérioration, qui permet de capturer des caractéristiques plus vagues ou globales des images dans les couches plus profondes, cependant, lorsque cette détérioration devient trop grande, cela peut nuire au réseau. Le nombre de couches optimal est donc non-seulement déterminé par la résolution et le contenu des données, il est aussi déterminé par le taux de détérioration, qu'on pourrait équivaloir aux opérations max pooling décrites à la section précédente.

**Blocs convolutifs**

Les convolutions font partie de blocs convolutifs, qui sont les composants de base du réseau U-Net. Ces blocs sont composés d'une convolution à deux dimensions, suivie d'une normalisation de batch optionnelle, une couche ReLU et puis finalement une couche dropout.

Le réseau U-Net original utilise des convolutions 3 × 3 non-paddées pour l'ensemble du réseau. Ceci a pour effet de réduire la taille des canaux de 2 × 2 pixels à chaque convolution. Les canaux deviennent donc de plus en plus petits au fur et à mesure qu'on approche de la sortie du réseau. De plus, lorsque le réseau doit concaténer les *skips connections*, cela nécessite un recadrage des données des couches contractives pour réduire leur taille aux dimensions des couches expansives. La plupart des réseaux plus récents utilisent donc un padding de 1 lors des convolutions 3 × 3, ce qui préserve les dimensions des canaux et facilite les changements subséquents possibles à l'architecture. Cette façon de procéder pourrait aussi avoir l'avantage de préserver mieux l'information en bordure des images.

**Normalisation de batch**

Le réseau U-Net original n'utilise pas de normalisation de batch. Cependant, plusieurs études par la suite ont trouvées qu'une structure U-Net peut bénéficier d'une telle normalisation (McCoy et al., 2019; Gros et al, 2019; Paugam et al., 2019). Certaines études placent la normalisation en amont la couche ReLU, tandis que d'autres la placent en aval. Il ne semble pas avoir de consensus clair sur l'utilisation systématique de la normalisation de batch dans une structure U-Net, ni sur la position optimale de cette couche si elle est utilisée.

Les couches de normalisation de batch ont comme paramètre principal le *momentum*. Celui-ci détermine l'importance accordée à la moyenne mobile de la mini-batch précédente dans le calcul de la moyenne mobile actuelle. Plus il est élevé, plus d'importance sera accordée à la moyenne précédente dans le calcul actuel, selon la formule suivante :

$$\mu = a * \mu_{prec} + (1 - a) * \mu_{act} \tag{3.7}$$

$\mu = moyenne\ mobile$ à $appliquer\ actuellement$ 3.1
$a = momentum$
$\mu_{prec} = moyenne\ mobile\ appliquée\ précédemment$
$\mu_{act} = moyenne\ mobile\ de\ la\ batch\ actuelle$

Cette formule est la même pour la variance utilisée, si l'on remplace les termes de moyennes mobiles pour des termes de variance. Il est généralement recommandé d'appliquer une valeur

assez élevée pour le momentum, soit entre 0.9 et 0.999. La valeur par défaut pour la grande majorité des framework d'apprentissage profond est de 0.9. La valeur du momentum peut être réduite si la taille de batch lors de l'entrainement est très élevée, étant donné que dans ces cas les valeurs pour la moyenne et la variance de batch s'approchent des valeurs de la population entière.

**Couches dropout**

Le réseau U-Net original n'utilise pas de couches dropout. De nombreuses études publiées par la suite (McCoy et al., 2019; Gros et al, 2019; Paugam et al., 2019) ont cependant trouvées qu'une architecture de type U-Net peut bénéficier de telles couches. Ces études placent habituellement les couches dropout à la suite des couches ReLU dans l'ensemble de leur réseau, cependant, certaines études placent seulement ces couches à certains endroits plus spécifiques dans leur réseau (McCoy et al., 2019).

L'étude (Li et al., 2019) argumente que les couches dropout sont fondamentalement incompatibles avec les couches de normalisation de batch lorsqu'elles sont placées en amont de ces dernières dans un réseau. Les auteurs mettent en cause un phénomène qu'ils appellent le *variance shift*, où la variance mobile utilisée lors du calcul de normalisation de batch est déréglée par le fait que certains poids soient désactivés aléatoirement dans les couches dropout. Cependant, plusieurs études, dont (Gros et al., 2019) utilisent les deux types de normalisation entremêlées avec succès.

**Convolutions transposées**

Les auteurs de la publication sur le réseau U-Net original ne spécifient pas le type exact de convolution utilisée afin d'agrandir la taille des canaux dans le chemin expansif. Cependant, la majorité des publications subséquentes sur l'architecture de type U-Net utilisent des convolutions transposées.

Une convolution transposée fonctionne en quelque sorte comme l'inverse d'une convolution régulière. Dans une convolution régulière, chaque élément du kernel convolutif est multiplié

avec un élément différent de l'entrée, puis les résultats de ces multiplications sont additionnés pour chaque déplacement du kernel et forment un élément de la sortie. Dans une convolution transposée, chaque élément du kernel est multiplié avec un seul élément de l'entrée, et chacune de ces multiplications est additionnée à un élément de la sortie différent. La Figure-A V-2 illustre bien le principe. On voit qu'un élément de l'entrée est multiplié avec chacun des neuf éléments du kernel pour former neuf éléments de la sortie. Au fur et à mesure que le kernel se déplace, les éléments dans la sortie s'accumulent par addition si la stride est plus petite que le kernel.

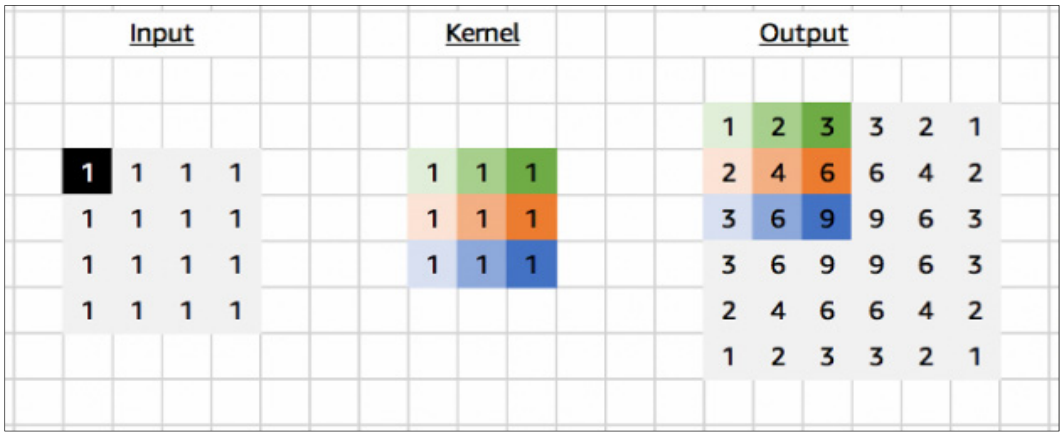



Figure-A V-2 Convolution transposée avec un kernel de 3 x 3, une stride de 1, une entrée 4 x 4 et une sortie 6 x 6
Tirée de Thom Lane, 2018

Les réseaux entrainés lors de la présente étude utilisent tous ce type de déconvolution. Ce choix est motivé par la littérature aussi bien que par des tests empiriques préliminaires comparant les résultats de convolutions transposées avec ceux d'interpolations bilinéaires et d'opérations de *max unpooling*.

La taille des kernels lors des convolutions transposées est déterminée par la dimension des opérations de max pooling dans le chemin contractif, de façon à ce que les canaux retrouvent la même taille qu'ils avaient dans la couche correspondante du chemin contractif. Ceci est

rendu possible par le fait que, donné le même kernel et la même stride, l'opération de convolution transposée produit toujours les dimensions inverses d'une opération de pooling.

**Portes d'attention**

Plusieurs études ont récemment utilisé des structures appelées *portes d'attention* en combinaison avec des réseaux de type U-Net (Oktay et al., 2018; Schlemper et al., 2019). Ces portes d'attention sont placées entre chacune des couches contractives et expansives. Elles prennent comme signaux d'entrée les données à être concaténées venant des couches contractives ainsi que les données de la couche expansive précédente. Leur rôle est d'identifier les régions clés des données qui contiennent les structures à segmenter et transférer cette information en aval. Elles viennent donc en quelque sorte aider à transférer les caractéristiques plus globales mais moins précises apprises par les couches plus profondes aux couches moins profondes contenant des cartes caractéristiques plus précises.

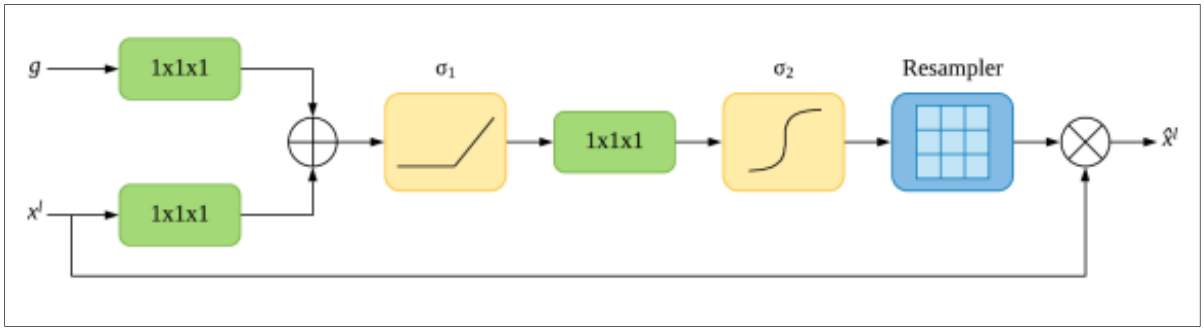



Figure-A V-3 Diagramme d'une porte d'attention. Le signal *g* correspond aux donnée arrivant des couches contractives. Le signal *x* correspond aux données arrivant de la couche expansive précédente
Tirée de Siddique et al., 2021

Comme on peut le voir sur la Figure-A V-3, le signal provenant des couches contractives, qui est sensé contenir l'information sur les régions clés des images, est additionné au signal des couches expansives, qui contient de l'information plus détaillée mais moins globale. Les convolutions 1 × 1 servent de transformations linéaires. Suite à l'addition des signaux, une fonction ReLU est appliquée, suivie d'une autre transformation linéaire, puis une couche

sigmoïde et une interpolation bilinéaire pour redimensionner les données. Le signal résultant est finalement multiplié avec le signal de la couche expansive précédente.

## BIBLIOGRAPHIE


Ahuja, C. S., Wilson, J. R., Nori, S., Kotter, M. R. N., Druschel, C., Curt, A., & Fehlings, M. G. (2017). Traumatic spinal cord injury. *Nature Reviews Disease Primers, 3*(1), 17018. doi:10.1038/nrdp.2017.18

Al-Habib, A. F., Attabib, N., Ball, J., Bajammal, S., Casha, S., & Hurlbert, R. J. (2009). Clinical Predictors of Recovery after Blunt Spinal Cord Trauma: Systematic Review. *Journal of Neurotrauma, 28*(8), 1431-1443. doi:10.1089/neu.2009.1157

Bouckaert, R. R. (2003). *Choosing between two learning algorithms based on calibrated tests*. Paper presented at the Proceedings of the Twentieth International Conference on International Conference on Machine Learning, Washington, DC, USA.

Chen, X., Yuan, Y., Zeng, G., & Wang, J. (2021). Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2613-2622.

Chopra, S., Balakrishnan, S., & Gopalan, R. (2013). *DLID: Deep Learning for Domain Adaptation by Interpolating between Domains*.

Chu, B., Madhavan, V., Beijbom, O., Hoffman, J., & Darrell, T. (2016). Best Practices for Fine-Tuning Visual Classifiers to New Domains. In (pp. 435-442).

De Leener, B., Kadoury, S., & Cohen-Adad, J. (2014). Robust, accurate and fast automatic segmentation of the spinal cord. *Neuroimage, 98*, 528-536. doi:10.1016/j.neuroimage.2014.04.051

De Leener, B., Lévy, S., Dupont, S. M., Fonov, V. S., Stikov, N., Louis Collins, D., . . . Cohen-Adad, J. (2017). SCT: Spinal Cord Toolbox, an open-source software for processing spinal cord MRI data. *Neuroimage, 145*(Pt A), 24-43. doi:10.1016/j.neuroimage.2016.10.009

De Leener, B., Taso, M., Cohen-Adad, J., & Callot, V. (2016). Segmentation of the human spinal cord. *Magnetic Resonance Materials in Physics, Biology and Medicine, 29*(2), 125-153. doi:10.1007/s10334-015-0507-2

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation, 10*(7), 1895-1923. doi:10.1162/089976698300017197

Dixon, A. (2019). Cervical flexion teardrop fracture with cord haemorrhage. Retrieved from https://radiopaedia.org/cases/cervical-flexion-teardrop-fracture-with-cord-haemorrhage?lang=gb

Drozdzal, M., Chartrand, G., Vorontsov, E., Shakeri, M., Di Jorio, L., Tang, A., . . . Kadoury, S. (2018). Learning normalized inputs for iterative estimation in medical image segmentation. *Medical Image Analysis, 44*, 1-13. doi:https://doi.org/10.1016/j.media.2017.11.005

Gros, C., De Leener, B., Badji, A., Maranzano, J., Eden, D., Dupont, S. M., . . . Cohen-Adad, J. (2019). Automatic segmentation of the spinal cord and intramedullary multiple sclerosis lesions with convolutional neural networks. *Neuroimage, 184*, 901-915. doi:10.1016/j.neuroimage.2018.09.081

Gros, C., De Leener, B., Dupont, S. M., Martin, A. R., Fehlings, M. G., Bakshi, R., . . . Sdika, M. (2018). Automatic spinal cord localization, robust to MRI contrasts using global curve optimization. *Med Image Anal, 44*, 215-227. doi:10.1016/j.media.2017.12.001

Gupta, R., Mittal, P., Sandhu, P., Saggar, K., & Gupta, K. (2014). Correlation of qualitative and quantitative MRI parameters with neurological status: a prospective study on patients with spinal trauma. *Journal of clinical and diagnostic research : JCDR, 8*(11), RC13-RC17. doi:10.7860/JCDR/2014/9471.5181

He, K., Zhang, X., Ren, S., & Sun, J. (2016, 27-30 June 2016). *Deep Residual Learning for Image Recognition.* Paper presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hernandez, D., & Brown, T. (2020). *Measuring the Algorithmic Efficiency of Neural Networks*.

Indoml. (2021). Student Notes: Convolutional Neural Networks (CNN) Introduction. Retrieved from https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: accelerating deep network training by reducing internal covariate shift*. Paper presented at the Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, Lille, France.

Kingma, D. P., & Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. http://arxiv.org/abs/1412.6980

Kost, J., & McDermott, M. (2002). Combining dependent P-values. *Statistics & Probability Letters, 60*, 183-190. doi:10.1016/S0167-7152(02)00310-3

Lane, T. (2018). Transposed Convolution Explained with... MS Excel! Retrieved from https://medium.com/apache-mxnet/transposed-convolutions-explained-with-ms-excel-52d13030c7e8

Lemay, A., Gros, C., Zhuo, Z., Zhang, J., Duan, Y., Cohen-Adad, J., & Liu, Y. (2021). Automatic multiclass intramedullary spinal cord tumor segmentation on MRI with deep learning. *NeuroImage: Clinical, 31*, 102766. doi:https://doi.org/10.1016/j.nicl.2021.102766

Li, X., Chen, S., Hu, X., & Yang, J. (2019, 15-20 June 2019). *Understanding the Disharmony Between Dropout and Batch Normalization by Variance Shift.* Paper presented at the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Li, Y., Wang, N., Shi, J., Liu, J., & Hou, X. (2016). Revisiting Batch Normalization For Practical Domain Adaptation. *Pattern Recognition, 80*. doi:10.1016/j.patcog.2018.03.005

Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics, 6*(1), 1-29. doi:10.1080/23270012.2019.1570365

Lundervold, A. S., & Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik, 29*(2), 102-127. doi:https://doi.org/10.1016/j.zemedi.2018.11.002

Magu, S., Singh, D., Yadav, R. K., & Bala, M. (2015). Evaluation of Traumatic Spine by Magnetic Resonance Imaging and Correlation with Neurological Recovery. *Asian spine journal, 9*(5), 748-756. doi:10.4184/asj.2015.9.5.748

Markiewicz, C. (2002). Coordinate systems and affines. Retrieved from https://nipy.org/nibabel/coordinate_systems.html

Martineau, J., Goulet, J., Richard-Denis, A., & Mac-Thiong, J.-M. (2019). The relevance of MRI for predicting neurological recovery following cervical traumatic spinal cord injury. *Spinal Cord, 57*(10), 866-873. doi:10.1038/s41393-019-0295-z

McCoy, D. B., Dupont, S. M., Gros, C., Cohen-Adad, J., Huie, R. J., Ferguson, A., . . . Talbott, J. F. (2019). Convolutional Neural Network-Based Automated Segmentation of the Spinal Cord and Contusion Injury: Deep Learning Biomarker Correlates of Motor Impairment in Acute Spinal Cord Injury. *AJNR Am J Neuroradiol, 40*(4), 737-744. doi:10.3174/ajnr.A6020

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation*.

Miyanji, F., Furlan, J. C., Aarabi, B., Arnold, P. M., & Fehlings, M. G. (2007). Acute cervical traumatic spinal cord injury: MR imaging findings correlated with neurologic outcome--prospective study with 100 consecutive patients. *Radiology, 243*(3), 820-827. doi:10.1148/radiol.2433060583

Morvant, E. (2017). Transfer Learning and Domain Adaptation. Retrieved from https://en.wikipedia.org/wiki/Domain_adaptation#/media/File:Transfer_learning_and_domain_adaptation.png

Oktay, O., Schlemper, J., Folgoc, L., Lee, M., Heinrich, M., Misawa, K., . . . Rueckert, D. (2018). Attention U-Net: Learning Where to Look for the Pancreas.

Paugam, F., Lefeuvre, J., Perone, C. S., Gros, C., Reich, D. S., Sati, P., & Cohen-Adad, J. (2019). Open-source pipeline for multi-class segmentation of the spinal cord with deep learning. *Magn Reson Imaging, 64*, 21-27. doi:10.1016/j.mri.2019.04.009

Pereira, S., Pinto, A., Alves, V., & Silva, C. A. (2016). Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images. *IEEE Transactions on Medical Imaging, 35*(5), 1240-1251. doi:10.1109/TMI.2016.2538465

Perone, C. S., Calabrese, E., & Cohen-Adad, J. (2018). Spinal cord gray matter segmentation using deep dilated convolutions. *Scientific Reports, 8*(1), 5966. doi:10.1038/s41598-018-24304-3

Poole, W., Gibbs, D. L., Shmulevich, I., Bernard, B., & Knijnenburg, T. A. (2016). Combining dependent P-values with an empirical adaptation of Brown's method. *Bioinformatics, 32*(17), i430-i436. doi:10.1093/bioinformatics/btw438

Prados, F., Ashburner, J., Blaiotta, C., Brosch, T., Carballido-Gamio, J., Cardoso, M. J., . . . Cohen-Adad, J. (2017). Spinal cord grey matter segmentation challenge. *Neuroimage, 152*, 312-329. doi:https://doi.org/10.1016/j.neuroimage.2017.03.010

Qi, G. J., & Luo, J. (2022). Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(4), 2168-2187. doi:10.1109/TPAMI.2020.3031898

Ronneberger, O., Fischer, P., & Brox, T. (2015, 2015//). *U-Net: Convolutional Networks for Biomedical Image Segmentation.* Paper presented at the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Cham.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review, 65 6*, 386-408.

Rote, G. (1991). Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters, 38*(3), 123-127. doi:https://doi.org/10.1016/0020-0190(91)90233-8

Rozantsev, A., Salzmann, M., & Fua, P. (2016). Beyond Sharing Weights for Deep Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PP*. doi:10.1109/TPAMI.2018.2814042

Salzberg, S. L. (1997). On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery, 1*(3), 317-328. doi:10.1023/A:1009752403260

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention Gated Networks: Learning to Leverage Salient Regions in Medical Images. *Medical Image Analysis, 53*. doi:10.1016/j.media.2019.01.012

Shah, M., Xiao, Y., Subbanna, N., Francis, S., Arnold, D. L., Collins, D. L., & Arbel, T. (2011). Evaluating intensity normalization on MRIs of human brain with multiple sclerosis. *Med Image Anal, 15*(2), 267-282. doi:10.1016/j.media.2010.12.003

Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access, 9*, 82031-82057. doi:10.1109/ACCESS.2021.3086020

Skeers, P., Battistuzzo, C. R., Clark, J. M., Bernard, S., Freeman, B. J. C., & Batchelor, P. E. (2018). Acute Thoracolumbar Spinal Cord Injury: Relationship of Cord Compression to Neurological Outcome. *J Bone Joint Surg Am, 100*(4), 305-315. doi:10.2106/jbjs.16.00995

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res., 15*(1), 1929–1958.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017, 2017//). *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations.* Paper presented at the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Cham.

Sun, B., Feng, J., & Saenko, K. (2016). *Return of frustratingly easy domain adaptation*. Paper presented at the Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona.

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging, 15*(1), 29. doi:10.1186/s12880-015-0068-x

Vovk, V. (2012). Combining p-values via averaging.

Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing, 312*, 135-153. doi:https://doi.org/10.1016/j.neucom.2018.05.083

Wideman, G. (2003). Orientation and Voxel-Order Terminology. Retrieved from http://www.grahamwideman.com/gw/brain/orientation/orientterms.htm

Wilson Daniel, J. (2019). The harmonic mean p-value for combining dependent tests. *Proceedings of the National Academy of Sciences, 116*(4), 1195-1200. doi:10.1073/pnas.1814092116

Xiao, T., Li, H., Ouyang, W., & Wang, X. (2016). Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1249-1258.

Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., & Zuo, W. (2017). Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems (NIPS), 27*.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. J. (2019). CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6022-6031.

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2015). *Supervised Representation Learning: Transfer Learning with Deep Autoencoders.* Paper presented at the IJCAI.