A priori et apprentissage profond pour la segmentation en imagerie cérébrale

Pierre-Antoine Ganaye

N°d'ordre NNT : 2019LYSEI100

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé (CREATIS)

Ecole Doctorale N° 160
(Electronique, Electrotechnique, Automatique)

Spécialité/ discipline de doctorat :
Traitement du Signal et de l'Image

Soutenue publiquement le 26/11/2019, par :
Pierre-Antoine Ganaye

A priori et Apprentissage Profond pour la Segmentation en Imagerie Cérébrale

---

Devant le jury composé de :

Petitjean, Caroline	Maître de conférences HDR	Université de Rouen	Rapporteur
Thome, Nicolas	Professeur des Universités	CNAM	Rapporteur
Garcia, Christophe	Professeur des Universités	INSA Lyon	Examinateur
Jodoin, Pierre-Marc	Professeur des Universités	Université de Sherbrooke	Examinateur
Sdika, Michaël	Ingénieur de Recherche	CNRS	Invité
Benoit-Cattin, Hugues	Professeur des Universités	INSA Lyon	Directeur de thèse

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage **secretariat@edchimie-lyon.fr** INSA : R. GOURDON	M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec. : M.C. HAVGOUDOUKIAN **ecole-doctorale.eea@ec-lyon.fr**	M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 **gerard.scorletti@ec-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : H. CHARLES **secretariat.e2m2@univ-lyon1.fr**	M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://www.ediss-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE **secretariat.ediss@univ-lyon1.fr**	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 **emmanuelle.canet@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 Fax : 04.72.43.16.87 **infomaths@univ-lyon1.fr**	M. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 **zamboni@maths.univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Marion COMBE Tél : 04.72.43.71.70 Fax : 04.72.43.87.12 Bât. Direction **ed.materiaux@insa-lyon.fr**	M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 **jean-yves.buffiere@insa-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Marion COMBE Tél : 04.72.43.71.70 Fax : 04.72.43.87.12 Bât. Direction **mega@insa-lyon.fr**	M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **http://ed483.univ-lyon2.fr** Sec. : Viviane POLSINELLI Brigitte DUBOIS INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 **viviane.polsinelli@univ-lyon2.fr**	M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Résumé

L'imagerie médicale est un domaine vaste guidé par les avancées en instrumentation, en techniques d'acquisition et en traitement d'images. Les progrès réalisés dans ces grandes disciplines concourent tous à l'amélioration de la compréhension de phénomènes physiologiques comme pathologiques.

En parallèle, l'accès à des bases de données d'imagerie plus large, associé au développement de la puissance de calcul, a favorisé le développement de méthodologies par apprentissage machine pour le traitement automatique des images dont les approches basées sur des réseaux de neurones profonds. Parmi les applications où les réseaux de neurones profonds apportent des solutions, on trouve la segmentation d'images qui consiste à localiser et délimiter dans une image les régions avec des propriétés spécifiques qui seront associées à une même structure. Malgré de nombreux travaux récents en segmentation d'images par réseaux de neurones, l'apprentissage des paramètres d'un réseau de neurones reste guidé par des mesures de performances quantitatives n'incluant pas la connaissance de haut niveau de l'anatomie.

L'objectif de cette thèse est de développer des méthodes permettant d'intégrer des a priori dans des réseaux de neurones profonds, en ciblant la segmentation de structures cérébrales en imagerie IRM. Notre première contribution propose une stratégie d'intégration de la position spatiale du patch à classifier, pour améliorer le pouvoir discriminant du modèle de segmentation. Ce premier travail corrige considérablement les erreurs de segmentation étant très éloignées de la réalité anatomique, en améliorant également la qualité globale des résultats. Notre deuxième contribution est ciblée sur une méthodologie pour contraindre les relations d'adjacence entre les structures anatomiques, et ce directement lors de l'apprentissage des paramètres du réseau, dans le but de renforcer le réalisme des segmentations produites. Nos expériences permettent de conclure que la contrainte proposée corrige les adjacences non-admises, améliorant ainsi la consistance anatomique des segmentations produites par le réseau de neurones.

Remerciements

À mes parents qui m'ont soutenu aveuglément dans la poursuite de mes études.

À Clémence pour avoir été présente pour le meilleur et pour le pire, en m'apportant ta joie de vivre au quotidien.

À Hugues pour m'avoir accordé sa confiance, partagé son expérience et pour être toujours là pour éclaircir mes pensées.

À Michaël sans qui mon avenir aurait été tout autre, merci pour la confiance que tu m'as accordée il a plusieurs années et pour le partage du flot ininterrompu de ton savoir.

À mes amis, Benjamin, Côme, Chet, Flo, Guillaume, Lény, Max, Méloée, Thibault, Valentine de continuer à perfectionner mon éducation sociale.

À mes mentors qui ont donné du sens à mes études universitaires, Jairo Cugliari et Julien Ah-pine.

À mes collègues de laboratoire, Fei, Noélie, Sarah, Thu, Hussein, Anchen pour avoir survécu à mon humour.

À mes responsables du département Télécom de l'INSA, Stéphane Frénot et François Lesueur, pour toute la confiance que vous m'avez apportée dans ma tâche d'enseignement ces trois dernières années.

À tous les personnels de CREATIS qui font vivre le laboratoire au quotidien.

Un grand merci à Caroline Petitjean et Nicolas Thome pour avoir accepté de participer à l'évaluation de mon travail de thèse, c'est un honneur de pouvoir bénéficier de votre expérience et de votre temps en tant que rapporteur. Je tiens également à remercier Christophe Garcia et Pierre-Marc Jodoin de me faire partager leur expertise en tant que examinateur.

À travers le doctorat, nous poursuivons la recherche d'innovations scientifiques, j'y ai personnellement vécu une riche aventure humaine, grâce à vous.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

L'imagerie médicale est un domaine vaste guidé par les avancées en instrumentation, en techniques d'acquisition, en reconstruction d'image ou encore en traitement du signal. Les progrès réalisés dans l'une ou l'autre de ces disciplines concourent tous à l'amélioration de la recherche ou de la prise en charge clinique. Par exemple, l'évolution des techniques d'acquisition en imagerie médicale permet une meilleure visualisation de l'anatomie et des mécanismes fonctionnels ainsi qu'une amélioration de la précision des marqueurs d'imagerie. En parallèle, le développement et l'accès à des bases de données d'imagerie de plus en plus grandes et parfois accompagnées d'annotations fournies par des experts médicaux, a favorisé le développement d'outils applicatifs pour la classification d'images et la segmentation de structures anatomiques. Le développement de méthodologies pour le traitement automatique des images a bénéficié de l'accès à ces données, en particulier pour les approches basées sur des réseaux de neurones profonds. Malgré toutes ces avancées, l'apprentissage des paramètres d'un réseau de neurones est guidé par des mesures de performances qui peuvent conduire à des incohérences anatomiques, dans un domaine où chaque décision doit être rationnelle. Les connaissances médicales développées depuis des décennies, en anatomie humaine, ne sont à l'heure actuelle pas incorporées lors de la prise de décision d'un réseau de neurones, pourtant chaque résultat devrait être réaliste du point de vue anatomique avant d'être précis.

Au cours de cette thèse, nous avons développé des méthodes innovantes pour intégrer des *a priori* dans des réseaux de neurones profonds, pour la segmentation de structures cérébrales en IRM. Ces connaissances obtenues à partir des données peuvent être introduites sous la forme d'entrées supplémentaires du modèle, ou de contraintes exercées sur les paramètres du modèles lors de l'apprentissage.

Ce document est organisé en trois parties et 17 chapitres. La première partie est consacrée à la présentation de l'état de l'art en segmentation d'images médicales. Les deux autres parties présentent nos principales contributions.

Dans la partie II, nous étudions une stratégie d'intégration de la position du patch à classifier, pour améliorer le pouvoir discriminant du modèle de segmentation cérébrale.

Dans la partie III, nous proposons une méthodologie pour renforcer un *a priori* anatomique lors de l'apprentissage des paramètres du réseau, dans le but de d'augmenter le réalisme des segmentations produites.

I État de l'art

Chapitre 1

Introduction

Dans cette première partie, nous explorons l'utilisation de la segmentation en imagerie cérébrale, en listant les principales approches utilisées au cours des vingt dernières années, jusqu'au réseaux de neurones. Après avoir expliqué les spécificités de l'imagerie médicale, nous faisons une revue de la segmentation d'images par réseaux de neurones profonds.

Les approches principales pour la segmentation d'images médicales sont introduites dans le chapitre 2, où après avoir donné quelques exemples d'utilisations cliniques, nous décrivons les méthodologies reposant sur le recalage et l'apprentissage automatique. Dans ce même chapitre, nous faisons un état des lieux des contraintes liées à la mise en place d'un modèle de segmentation, du point de vue des données, du type d'imagerie et des ressources de calcul.

L'apprentissage profond de réseaux de neurones en segmentation d'images étant l'approche centrale de ce manuscrit, les concepts phares comme l'optimisation des paramètres, les couches utilisées, les architectures ou encore les fonctions de coût, sont présentés dans le chapitre 3, en préambule des deux parties suivantes qui présentent nos contributions.

Chapitre 2

Segmentation en imagerie médicale

La segmentation d'images consiste à créer une partition de l'image en régions, les pixels d'une région ayant des propriétés communes qui les différencient des pixels des autres régions. On affecte alors une étiquette pour identifier les pixels d'une même région. Dans certaines applications, ces régions sont associées à des objets ou encore à des structures anatomiques. Ce processus de délimitation peut être effectué manuellement, cependant en fonction du nombre de régions à délimiter, de la taille de l'image ou encore du type d'image, la complexité de l'annotation peut augmenter au point de nécessiter un expert entraîné pendant plusieurs années d'étude et de pratique, comme c'est le cas pour un radiologue. En imagerie médicale, l'accès à un praticien expérimenté est souvent limité par le temps dont il dispose et par les ressources financières à disposition pour le rémunérer. D'autres facteurs externes comme la fatigue visuelle ou la répétitivité de la tâche, peuvent par la suite influencer la qualité de la segmentation et éventuellement le diagnostic ou la prise en charge thérapeutique.

Dans ce chapitre, nous présentons dans un premier temps dans la section 2.1 des applications répandues de la segmentation en imagerie médicale, puis les approches majeures explorées jusqu'à présent pour automatiser cette tâche (section 2.2). Nous clôturons le chapitre par la section 2.3, dans laquelle nous abordons les défis de l'imagerie cérébrale qui ont guidé les développements méthodologiques de cette thèse.

2.1 Pourquoi étudier le pathologique, l'anatomique ?

Dans les travaux présentés dans ce manuscrit, nous étudions exclusivement des méthodes de segmentations automatiques pour délimiter des structures anatomiques du corps humain, en particulier le cerveau. Ces outils sont intégrés dans de nombreux protocoles médicaux, allant de l'aide au diagnostic à la préparation d'une séance de radiothérapie. Nous

détaillons dans cette section certaines des utilisations de la segmentation automatique, pour l'imagerie cérébrale.

2.1.1 Imagerie cérébrale

Le cerveau est le centre de contrôle du corps humain, il gère le fonctionnement des organes vitaux, la cognition et le contrôle des muscles. Même si la compréhension de son fonctionnement reste encore partielle, le développement de l'imagerie cérébrale a permis de grandes avancées en neuro-sciences et en médecine pour caractériser des pathologies. Nous détaillons ici, quelques une des utilisations de l'imagerie cérébrale.

Connectivité fonctionnelle Les structures du cerveau interagissant entre elles lors du processus de cognition, la compréhension de ces mécanismes est possible grâce à l'Imagerie par Résonance Magnétique (IRM) fonctionnelle, où l'on étudie l'activation de régions au cours d'un stimuli [Kwong et al., 1992, Yu-Feng et al., 2007]. En mesurant la concentration en oxygène, on peut par exemple observer le besoin en énergie des tissus, dont on peut déduire des cartes d'activation du cerveau, qui mesurent l'activité de zones du cerveau en fonction d'actions conditionnées. Finalement, l'association de ces cartes d'activations à des régions anatomiques est possible par la segmentation d'une IRM anatomique acquise sur le même patient et recalée sur les cartes d'activation. Les méthodes de segmentation pour la connectivité cérébrale se basent sur des atlas (section 2.2.1) et aussi sur des approches d'apprentissage automatique tels que les réseaux de neurones (section 2.2.6.3).

Étude longitudinale Le suivi longitudinal consiste à étudier l'évolution d'un phénomène physiologique (ex : vieillissement) ou d'une pathologie (ex : sclérose en plaques (SEP), alzheimer) au cours du temps. Son rôle est d'étudier des caractéristiques visuelles comme la morphologie (atrophie, hypertrophie) de structures anatomiques, afin d'éventuellement adapter le traitement mis en place. La comparaison région par région à plusieurs instants temporels requière de quantifier les informations morphologiques pour toutes les séries d'images acquises. Cette tâche mobilise un radiologue pour annoter des quantités importantes d'images, dont la difficulté peut varier en fonction de la pathologie. Par exemple, dans le cas de la SEP, qui est une maladie neurodégénérative qui affecte les gaines de myéline autour des nerfs du cerveau et de la moelle épinière, les médecins effectuent un suivi longitudinal [Fisher et al., 2008, Simon et al., 1999] du patient pour observer l'apparition ou la diminution de lésions révélées sur les IRM. Le recalage est parfois empêchée par la présence de lésions, qui peuvent rendre des méthodes de recalage instables (ex : recalage déformable : section 2.2.2.2). Toutefois pour les pathologies neurologiques ne déformant pas les tissus, si on a segmenté les lésions, un pré-traitement [Sdika and Pelletier, 2009] permet de se ramener au cas d'un sujet sain.

Électroencéphalographie (EEG) L'EEG est un examen indolore et non-invasif pour mesurer l'activité électrique du cerveau à partir d'un ensemble d'électrodes placées sur le cuir chevelu. Il en résulte un électroencéphalogramme qui quantifie l'activité neurophysiologique du cerveau au cours du temps, ce qui peut aider au diagnostic clinique ou à comprendre le fonctionnement du cerveau par les neurosciences (source Wikipedia).

Dans le but de modéliser de façon précise la propagation d'un courant dans le cerveau, une segmentation préalable des structures cérébrales permet d'affiner les paramètres de propagation des modèles numériques sous-jacent. Dans ce contexte, une IRM du patient est acquise pour obtenir une segmentation de la boite crânienne, du liquide cérébro-spinal et de la matière blanche avec comme finalité de paramétrer les méthodes d'éléments finis [Cook and Koles, 2006, Wolters et al., 2006, Ferree et al., 2000] pour modéliser la propagation du courant. Ces informations sont en effet nécessaires à la création d'un modèle précis du flux de courant pour le ciblage et la reconstruction de la source de courant. Lors de la segmentation des images de patients, des outils tel que [Huang et al., 2013] ont été proposés pour simplifier le pipeline de traitement. Ils peuvent être basés sur le recalage (sections 2.2.2.1 et 2.2.2.2) ou l'apprentissage automatique de descripteurs visuels. En utilisation clinique, l'EEG est utilisé par exemple pour localiser le foyer épileptique à l'origine d'une crise [Fuchs et al., 2007], en vu de préparer son ablation.

2.2 Approches de segmentation anatomique pour l'imagerie cérébrale

Pour répondre au besoin d'outils de segmentation automatique pour l'imagerie cérébrale, plusieurs familles de méthodes de segmentation ont été développées ces dernières années, nous en explorons les principales dans cette section. Les approches qui reposent sur l'utilisation d'atlas anatomiques (sections 2.2.1), telles que recalage d'image (section 2.2.2) sont apparues dans les années 1990, avec la segmentation par atlas (section 2.2.3) et multi-atlas (section 2.2.4). Puis l'apparition de méthodes de clustering et d'apprentissage supervisé (section 2.2.6) ont permis le perfectionnement des méthodes multi-atlas et l'apparition des approches par fusion non-locale (section 2.2.5).

2.2.1 Atlas

Le terme atlas (figure 2.1) est utilisé dans le domaine médical pour nommer la paire constituée d'une image acquise (IRM, TDM, US, ...) ou moyennée et de sa carte annotée des structures. Cette dernière capture les propriétés anatomiques ainsi que les relations entre les régions annotées. La segmentation basée sur un atlas est une des méthodes les plus répandues en imagerie médicale. Dans le cadre de l'apprentissage supervisé, un atlas est la vérité terrain que l'on cherche à reproduire, pour la segmentation basée atlas (section

2.2.3) c'est l'image de référence que l'on déforme par recalage pour la faire correspondre à l'image à segmenter.

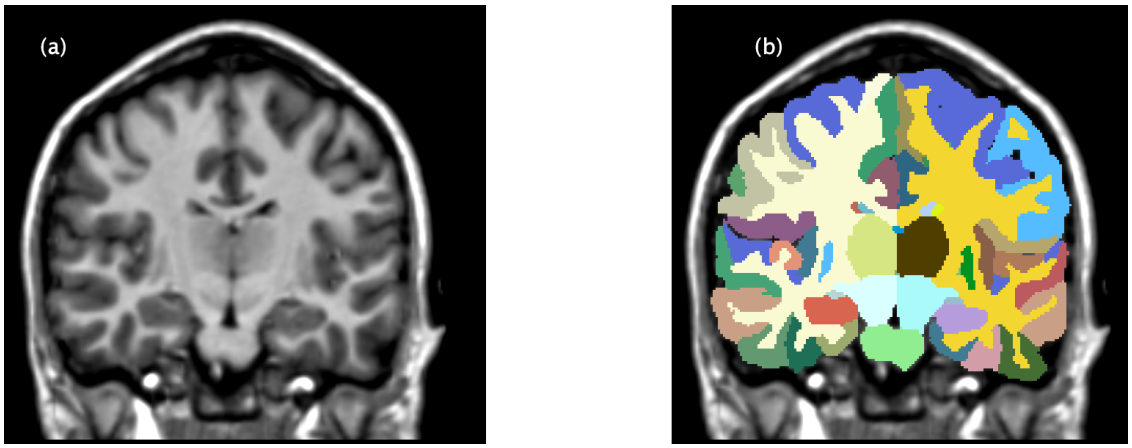


FIGURE 2.1 – Exemple d'une coupe cérébrale annotée en IRM issue d'un atlas. À gauche l'image acquise en IRM (a), à droite la combinaison de l'image et de la carte des structures anatomiques formant l'atlas (b).

Dans la suite de ce manuscrit, nous décrivons un atlas comme une paire (I, S) avec I l'image en niveau de gris et S la carte des structures anatomiques correspondante. Les atlas peuvent être construits à partir de plusieurs sujets choisis pour représenter la tendance moyenne, cela ne capture cependant pas la variabilité anatomique spécifique au sujet. Une base d'atlas est composée de plusieurs patients dans le but de saisir une diversité de la population plus large. Un atlas est normalement composé d'une image unique segmentée, mais cette dernière peut aussi être enrichie d'annotations issues d'images acquises avec des modalités ou des contrastes différents pour mettre en valeur des structures anatomiques d'intérêts, comme le montre la figure 2.2.

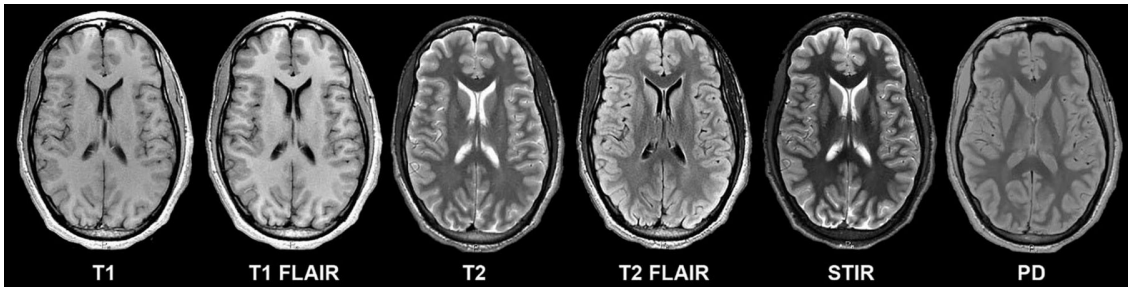


FIGURE 2.2 – Exemple de coupes cérébrales en IRM acquises avec différentes séquences, permettant d'observer certaines régions anatomiques avec un meilleur contraste. Source [Tanenbaum et al., 2017].

2.2.2 Recalage d'image

Le recalage est utile pour comparer des images issues de sources différentes (multimodale), permettant de mettre en commun des données hétérogènes (ex : TDM, IRM) ou encore pour l'interpolation inter-coupe et l'estimation de mouvement. Le recalage produit

une correspondance d'un espace de coordonnées Ω_F vers un autre espace de référence Ω_M, donnant la possibilité de passer d'un espace source à un espace cible.

Le recalage d'image est la recherche d'une transformation géométrique $\mathbf{T} : \Omega_F \rightarrow \Omega_M$, permettant de faire correspondre I_F une image fixe et I_M une image déformable (aussi appelée moving image) dans un espace de référence Ω_F, où x est une position dans Ω_F. Pour maximiser la similarité ou minimiser l'erreur entre les deux images $I_F(x)$ et $I_M(T(x))$, on dispose d'une fonction de transformation $\mathbf{T}$ qui décrit comment l'image déformable est réalignée sur l'image fixe. On utilise une mesure de similarité $C(I_F, I_M \circ T)$ pour mesurer la qualité du recalage, qui est la fonction de coût à maximiser à l'aide d'une méthode d'optimisation telle que l'algorithme du gradient. Les critères de similarité peuvent quantifier des informations morphologiques ou simplement basées sur l'intensité. Dans ce dernier cas on peut utiliser des mesures d'erreurs comme la somme des différences au carré ou des mesures issues de la théorie de l'information.

Les techniques de recalage sont souvent identifiées en fonction du caractère de la transformation : linéaire (rigide et affine) et non-linéaire.

2.2.2.1 Recalage linéaire

Les méthodes de recalage linéaire sont basées sur une transformation linéaire, qui représente un changement global de l'image et sont donc appliquées sur toute l'image. On définit une transformation linéaire $\mathbf{T}$ tel que $\mathbf{T}(x) = \mathbf{A}\mathbf{x} + b$ avec $\mathbf{A} \in \mathbb{R}^3 \times 3$ $b \in \mathbb{R}^3$. Le recalage affine et rigide sont considérés comme des méthodes linéaires par une majorité des auteurs.

Transformation rigide La transformation rigide applique une rotation et une translation avec 6 paramètres, 3 pour la rotation et 3 pour la translation.

Transformation affine La transformation affine étend l'approche rigide en incluant la mise à échelle et la transvection, portant le nombre de paramètres à 12. Cette transformation préserve les lignes parallèles mais pas les angles et distances. Avec $\mathbf{A}$ la matrice de transformation contenant 9 paramètres (rotation, mise à l'échelle, transvection) et $\mathbf{b}$ le vecteur de translation, on peut décrire la transformation d'un point $\mathbf{x}$ tel que $\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$.

2.2.2.2 Recalage non-linéaire

À l'inverse du recalage linéaire qui applique une transformation globale sur toute l'image, le recalage non-linéaire (élastique, déformable, non-rigide) fournit des transformations locales, qui apportent un meilleur alignement des régions. Le coût algorithmique est évidemment plus élevé du fait de la modélisation de plusieurs déformations locales, mais dans certaines applications médicales ce coût est justifié par la capacité de prise en compte

de la variabilité anatomique, qui ne se résume pas à des transformations affines. Parmi les méthodes non-linéaires, on retrouve des modélisations basées sur plusieurs concepts, tels que les fluides, les fonctions de base [Friston et al., 1995], les splines [Szeliski and Coughlan, 1997] et d'autres encore [Sotiras et al., 2013]. Les contraintes les plus répandues pour cette famille de transformation sont les suivantes : symétrique, inversible, de type difféomorphisme.

2.2.3 Mono-atlas

La segmentation basée atlas consiste à estimer la transformation $\mathbf{T}$ par recalage, pour ensuite aligner la segmentation manuelle de l'atlas S_M avec l'image du patient I_F. Comme on recherche à aligner de façon précise I_M sur I_F, il est d'usage de choisir une fonction T non-linéaire, qui s'adapte mieux aux déformations locales et ainsi à la variabilité du corps humain. Une fois les paramètres de la transformation déterminés, on propage les étiquettes de l'atlas de référence S vers l'image non-annotée avec $S(T(x))$, pour tous les pixels x (voir figure 2.3). Cette méthode est appelée propagation d'étiquettes ou encore segmentation par recalage, elle peut être généralisée à l'utilisation de plusieurs atlas.

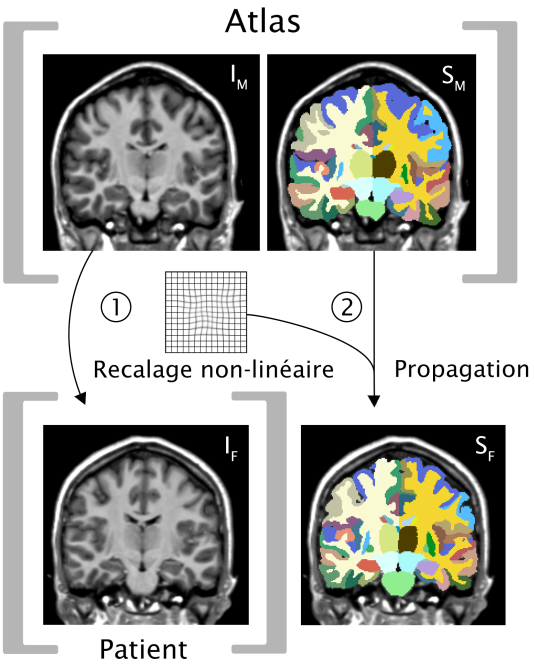


FIGURE 2.3 – La segmentation par atlas se base sur une déformation T pour propager les étiquettes de la carte S_M de l'atlas vers l'image d'un nouveau patient I_F.

2.2.4 Multi-atlas

La qualité de la segmentation donnée par le recalage basé sur un unique atlas dépend fortement de la performance de l'étape de recalage, le risque d'obtenir des erreurs augmente d'autant plus lorsque les images à recaler ne sont pas du même patient. Les erreurs

de segmentation étant courantes dans la majorité des applications, la robustesse de la segmentation basée sur un seul atlas (mono-atlas) est limitée par la méthode de recalage. Afin de répondre à ce problème, des travaux ont proposé d'exploiter simultanément plusieurs atlas (multi-atlas).

Dans l'approche de segmentation multi-atlas de [Wang and Yushkevich, 2013, Sdika, 2010], chaque image I_{M_i} issue de l'atlas $(I, S)_i$ d'une base comprenant n sujets, est recalée vers le patient de référence I_F. Les cartes d'étiquettes S_i résultant de la propagation $S_i(T_i(x))$ sont ensuite fusionnées (somme pondérée, vote à la majorité, voir revue dans [Artaechevarria et al., 2009]) pour produire une segmentation finale. Il est d'usage d'utiliser uniquement les atlas possédant une mesure de similarité forte avec l'image à recaler. La méthodologie multi-atlas apporte une meilleure robustesse aux erreurs grâce à la fusion des résultats de chaque recalage, mais aussi de par la variabilité anatomique plus large, captée par les n atlas de la base. Il existe plusieurs variations de la segmentation multi-atlas, qui s'articulent autour de la sélection des atlas et de la méthode de fusion des cartes d'annotations.

Sélection d'atlas Il est souhaitable d'exploiter dans le processus de recalage uniquement les atlas qui présentent des similarités avec l'image de référence, ce qui a tendance à améliorer la qualité du résultat. La majorité des méthodes de sélection [Aljabar et al., 2009, Rohlfing et al., 2004, Wu et al., 2007] utilise les mêmes métriques de similarité que pour le recalage (somme des moindres carrés, information mutuelle). Ces mesures sont appliquées après l'étape de recalage sur les images déformées, en comparant les différences pixel à pixel, pour finalement retenir pour la fusion des cartes les atlas partageant le plus de similarités.

Fusion des cartes d'annotations La fusion des cartes d'annotations obtenues est un domaine qui a reçu beaucoup de contributions avec le développement de l'apprentissage automatique. Les contributions sont divisées en deux familles : les approches globales qui assigne une pondération à l'échelle de l'image, puis les approches locales [Wang et al., 2012, Asman and Landman, 2013] où des poids sont fixés respectivement à des régions spatiales [Bai et al., 2013, Aljabar et al., 2007, Wang and Yushkevich, 2013, Sdika, 2010, Sdika, 2015]. Le choix de la pondération ou du nombre de vote, a également été étudié du point de vue de la classification automatique. Par exemple dans [Sdika, 2015] une approche mono-atlas propose d'associer une image d'intensité avec une image de classifieur, où pour chaque voxel x, un modèle paramétrique est entraîné sur les intensités du voisinage local à x issues de plusieurs atlas, pour corriger les possibles erreurs de recalage. En intégrant des fonctions de décision, on obtient une pondération locale plus fine et ainsi un résultat de meilleure pertinence que les approches globales.

Bien que la qualité de la carte de délimitation des régions soit améliorée par rapport au recalage mono-atlas, la complexité de l'approche multi-atlas se trouve multipliée par le nombre d'atlas, du fait qu'il faille recaler chacun des atlas sur l'image à segmenter. Toutefois, les approches locales de fusion ont inspiré le développement des méthodes basées par patch présentées dans la section suivante.

2.2.5 Fusion non-locale

Pour la plupart des méthodes de segmentation multi-atlas, l'étape de recalage est présentée comme le maillon faible du pipeline en raison de l'influence que peut avoir un mauvais alignement des images sur le résultat final. Cette difficulté est accentuée par la complexité anatomique de certaines régions comme le cortex, qui présente parfois des marqueurs visuels (intensité, texture) peu discriminants.

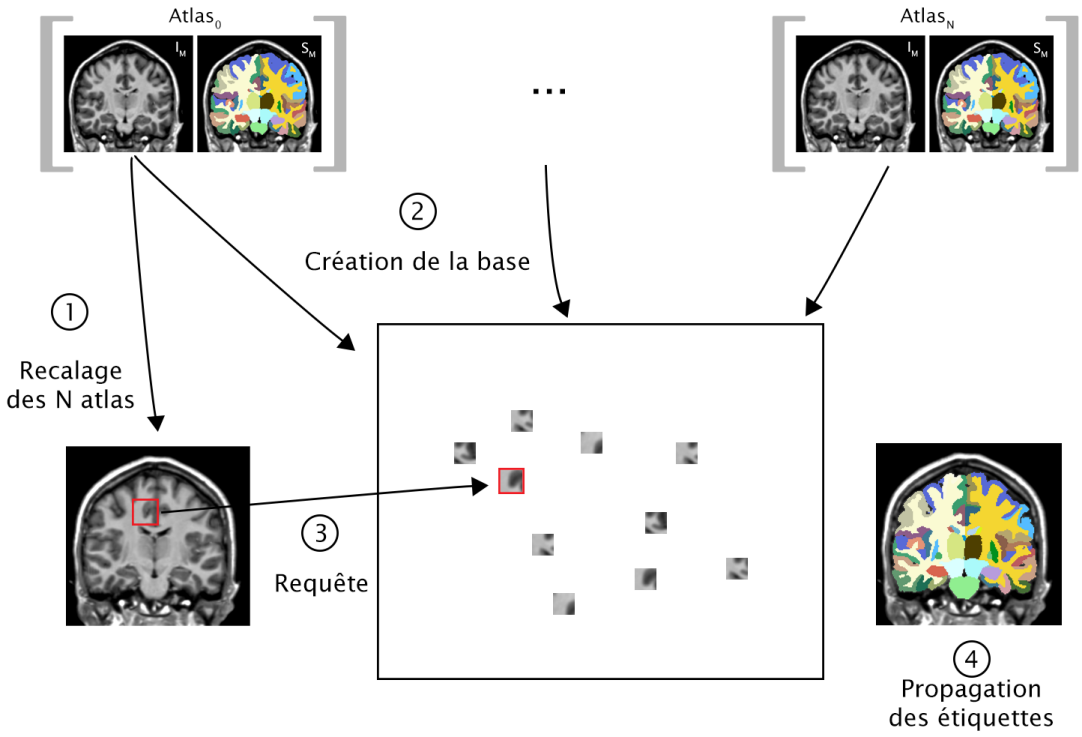


FIGURE 2.4 – La segmentation par fusion non-locale diminue les incertitudes liées à l'étape de recalage, en déterminant pour un patch centré en x, le patch $\hat{x}$ voisin de x et partageant le plus de similarité visuelle avec x, puis en propageant son étiquette. $\hat{x}$ est extrait des atlas après recalage affine ou déformable de ces derniers.

Un patch est une petite partie d'une image, généralement rectangulaire, utilisée en segmentation en tant que représentation locale du pixel central du patch. Les approches basées par fusion non-locale sont moins sensibles aux problèmes de recalage car ces derniers ne réalisent pas une mise en correspondance directe entre x et $\mathbf{T}(x)$. Plus précisément, pour chaque voxel x de l'image non-annotée I_F, on compare le patch centré en x avec l'ensemble des patchs contenus dans le voisinage de x des atlas I_{M_i}, pour attribuer l'étiquette du patch

le plus similaire (voir figure 2.4). Pour chaque voxel x de l'image à segmenter, le patch est une représentation brute du voisinage entourant x.

Bien que le patch soit de plus faible dimension que l'image d'origine, son utilisation simplifie le processus de recalage en diminuant les contraintes d'alignement des images, pouvant même obtenir des résultats de l'état de l'art, simplement avec une transformation linéaire afin d'orienter les images dans le même espace. Dans [Wang and Yushkevich, 2013, Rousseau et al., 2011, Coupé et al., 2011], les auteurs ont montré que l'utilisation de l'approche par patch pour la propagation des étiquettes apporte une qualité de segmentation similaire au recalage déformable, avec des temps de calcul qui sont réduits si l'alignement est fait avec une transformation linéaire.
Dans la section suivante, nous voyons comment les méthodes d'apprentissage supervisé permettent de répondre aux problématiques de la segmentation cérébrale, en s'appuyant ou pas sur l'utilisation du patch comme représentation globale d'un pixel.

2.2.6 Apprentissage automatique

L'apprentissage automatique est un vaste domaine à l'intersection de la statistique, des probabilités et de l'informatique, qui regroupe l'apprentissage supervisé et non-supervisé.

L'apprentissage supervisé inclus toutes les méthodes de prédiction dont l'utilisation nécessite des données annotées. On note $f(\mathbf{x};\theta) = \hat{y}$, un modèle prédictif (aussi appelé classifieur) qui prend en entrée un vecteur descripteur $\mathbf{x}$ dont on estime la classe ou étiquette $\hat{y}$. Les modèles supervisés sont généralement paramétrés par un ensemble de poids θ, optimisés lors de la phase d'apprentissage sur une base annotée d'exemples $(\mathbf{x}, y)$. Les classifieurs les plus utilisés en segmentation d'images médicales sont les séparateurs à vastes marges (SVM), les arbres de décision et les réseaux de neurones. Nous faisons dans cette section un bref focus sur les SVM (section 2.2.6.1), K plus proche voisin (section 2.2.6.2) et réseaux de neurones (section 2.2.6.3).

L'apprentissage non-supervisé regroupe par définition toutes les méthodes pour lesquelles on n'utilise pas d'annotation et dont la finalité est de trouver des similitudes dans les données (clustering) ou de réduire le nombre de dimension de l'espace de représentation. En segmentation d'images médicales, parmi les méthodes les plus connues, on compte les K plus proche voisin, l'analyse en composantes principales (ACP) et les approches dites Mean Shift [Comaniciu and Meer, 2002].

L'utilisation de modèles supervisés pour la segmentation cérébrale complète l'approche par fusion non-locale, en introduisant un apprentissage des paramètres se basant sur des descripteurs visuels [Lowe, 2004, Bay et al., 2006] des patchs. En plus d'améliorer la précision du recalage déformable, dans [Wang and Yushkevich, 2013, Sdika, 2015] la combinaison de plusieurs classifieurs corrige les erreurs d'alignement, sans avoir à éliminer l'atlas

du processus de propagation des étiquettes, préservant donc une meilleure variabilité des données.

La segmentation par patch a également l'avantage d'augmenter le nombre d'images à comparer, ce qui combiné à la faible dimension de l'espace de représentation du patch, apporte de meilleures garanties de généralisation (section 3.6), préférées en apprentissage automatique. En effet, le fléau de la dimension [Friedman, 1997] qui est un théorème classique en apprentissage automatique, précise que lorsque la dimensionnalité de l'espace de représentation augmente, la distance séparant les points augmente. C'est un problème dans le cadre de l'apprentissage supervisé, car pour que le classifieur capture toutes les typologies de données dans cet espace, la quantité d'exemples nécessaire doit augmenter au fil de la progression de la dimensionnalité. Ainsi, dans un espace de représentation plus compact (celui du patch), le nombre d'échantillons nécessaires à l'apprentissage d'un modèle généralisable est plus faible que pour un espace de plus forte dimension (celui de l'image d'origine). En général, il est préférable de réduire le nombre de dimension des images d'entrée du modèle, à l'aide de méthodes spécifiques comme la factorisation de matrice ou par des descripteurs visuels résumant des caractéristiques comme [Lowe, 2004, Bay et al., 2006].

Les méthodes de classification automatiques sont des solutions parfaitement adaptées à la segmentation par patch, notamment pour corriger les erreurs de recalage de l'étape de fusion des étiquettes. Des travaux [Wang and Yushkevich, 2013, Sdika, 2015, Rousseau et al., 2011, Powell et al., 2008, Vrooman et al., 2007], ont par ailleurs proposé de combiner le recalage (déformable ou linéaire) à l'utilisation d'un modèle paramétrique de type classifieur, tel qu'un séparateur à vaste marge (SVM) dans [Sdika, 2015, Powell et al., 2008], un réseau de neurones dans [Powell et al., 2008] ou encore un approche par plus proche voisin [Vrooman et al., 2007, Anbeek et al., 2013].

2.2.6.1 SVM

Pour la segmentation des structures sous-corticales, dans [Powell et al., 2008] les auteurs choisissent de comparer l'apport de l'apprentissage supervisé par rapport à la segmentation basée sur un atlas. Pour cela, toutes les images sont recalées par un modèle déformable multi-échelles, à la suite de quoi des descripteurs d'intensité, de position et probabiliste sont extraits. Ces derniers sont utilisés comme entrées de deux classifieurs supervisés : un réseau de neurones à trois couches (section 2.2.6.3) et un séparateur à vaste marge [Cortes and Vapnik, 1995]. Les résultats montrent que la qualité de segmentation des images est meilleure avec le réseau de neurones, suivi par le SVM et enfin l'approche par atlas. Ce travail démontre l'intérêt d'utiliser des modèles prédictifs pour prendre en compte des descripteurs liés à la forme ou à la position des structures, que la classique propagation d'étiquette ne peut réaliser.

2.2.6.2 K plus proche voisin

Dans [Vrooman et al., 2007, Anbeek et al., 2013], les auteurs choisissent de combiner au recalage un classifieur par plus proches voisins exploitant le patch comme représentation visuelle des pixels, ainsi que la position du patch dans [Anbeek et al., 2013]. Dans ces deux approches, de même que pour [Powell et al., 2008], la qualité de segmentation et la robustesse sont améliorées en combinant un classifieur au recalage déformable, au point que les délimitations produites pour le liquide cérébro-spinal, la matière blanche et la matière grise, sont équivalentes à celles données manuellement par des experts.

Bien que l'utilisation de méthodes supervisées ait amélioré les approches classiques de segmentation par atlas, nous verrons dans la section suivante comment les avancées de l'apprentissage profond ont fait évoluer encore plus loin la segmentation cérébrale.

2.2.6.3 Réseau de neurones

Ces dernières années, des avancées impressionnantes dans la communauté de l'apprentissage profond, ont repoussé les limites dans la majorité des challenges de reconnaissance visuelle. Bien que nous présentons ces méthodes en détails dans le chapitre 3.1, les principaux travaux de segmentation par réseaux de neurones convolutifs (RNC) sont introduits dès maintenant car ils sont dans la continuité de l'approche par patch.

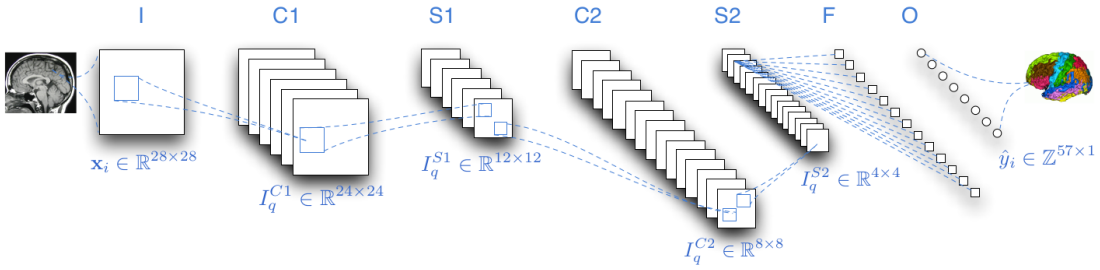


Fig. 1. **Deep learning approach to automate brain image parcellation using a convolutional network model.** *From left to right, the deep architecture consists of several layers starting with the input layer (I). In an alternating manner the CN consists of a hierarchical architecture of convolutional (C1, C2) and subsampling (S1, S2) layers followed by a full-connection layer (F), and finally the output layer (O).*

FIGURE 2.5 – Architecture du réseau de segmentation proposé dans [Lee et al., 2011], inspiré de [LeCun and Bengio, 1998], où chaque patch 2D extrait du volume est encodé par une suite de convolutions, pour donner en sortie la région d'appartenance la plus probable.

Les réseaux de neurones convolutifs (RNCs) sont une forme de réseaux de neurones spécialisés pour les tâches de vision par ordinateur, qui se distingue par l'utilisation de couches de convolution et de sous-échantillonage pour apprendre des représentations visuelles efficaces. L'utilisation du patch comme d'une description visuelle du voxel central est une configuration classique pour l'apprentissage d'un réseau de neurones convolutif, comme ce que proposent les auteurs dans [Lee et al., 2011, de Brebisson and Montana, 2015, Moeskops et al., 2016]. Dans [Lee et al., 2011], les auteurs exploitent une architecture de réseau à deux couches de convolution, similaire à celle proposée dans [LeCun and

Bengio, 1998], un des travaux de référence dans le domaine des RNCs. L'idée de choisir un réseau de neurones comme classifieur principal a été proposée quelques années plus tard dans [de Brebisson and Montana, 2015, Moeskops et al., 2016], avec des adaptations spécifiques à l'imagerie cérébrale.
Dans [de Brebisson and Montana, 2015], pour la segmentation des structures corticales et sous-corticales, les auteurs combinent des descripteurs visuels (patchs orthogonaux avec et sans sur-échantillonnage, patch 3D) avec des mesures de distances. Pour chaque source de données, un bloc encode l'information, puis tous les vecteurs caractéristiques des blocs sont fusionnés dans des couches totalement connectées avant de produire les cartes de probabilité de chaque région. [Moeskops et al., 2016] suggèrent que l'exploitation des patchs à plusieurs résolutions réduit les erreurs de segmentation. Pour cela, un réseau à trois blocs (un par résolution de patch) encode les données de résolutions croissantes, puis fusionne les descripteurs avant la couche de sortie. Les travaux [de Brebisson and Montana, 2015] orientent globalement les choix d'architecture pour mieux prendre en compte le contexte entourant le patch central, soit avec les coupes orthogonales, soit dans un voisinage plus large autour du patch.

L'utilisation des patchs comme représentation du voxel a toutefois ses limites dans le cadre des RNC. Par exemple, pour segmenter l'image d'un nouveau patient, tous les patchs de l'image sont extraits et traités pour obtenir la région la plus probable. La complexité temporelle de l'algorithme par patch se trouve limitée par le nombre de données à traiter, bien qu'il soit en théorie possible de paralléliser le traitement sur plusieurs ordinateurs, en raison de l'indépendance des résultats inter-patch. D'autres solutions plus efficaces basées sur les RNCs ont émergé récemment. Dans [Roy et al., 2017], les auteurs proposent d'adapter une nouvelle approche de segmentation bout-à-bout, où l'image en entrée du RNC est entièrement annotée (pixel par pixel) en sortie, ce qui réduit fortement les temps d'apprentissage et d'inférence, en permettant de la même façon l'utilisation de fonctions de coût spécifiques pour mesurer la similarité des régions, comme étudié dans [Sudre et al., 2017].

Dans la section suivante, nous identifions certaines spécificités propres à l'IRM cérébrale qu'il a été nécessaire de considérer dans les travaux que nous avons conduit, pour améliorer les performances des réseaux de neurones pour la segmentation d'images cérébrales et qui sont détaillés dans les parties II et III de ce manuscrit.

2.3 Spécificités de l'imagerie médicale

Le développement de nouvelles méthodologies pour l'imagerie cérébrale, aussi bien en segmentation de structures, en reconstruction d'images, que dans d'autres domaines applicatifs, est orienté en fonction des ressources matérielles ou humaines, des avancées technologiques et du cadre législatif. Sans ces trois contraintes, la majorité des problèmes d'imagerie

pourrait être abordée librement par la communauté scientifique, ce n'est malheureusement pas le cas. Pour composer avec cela, des méthodes originales sont proposées pour répondre aux limitations actuelles.

Dans cette section, nous présentons quelles sont les spécificités propres au domaine médical. L'imagerie par résonance magnétique (IRM) possède aussi des caractéristiques qui lui sont propres et qu'il faut prendre en considération lors de la phase de pré-traitement d'un algorithme d'analyse. Enfin, nous listons les principales contraintes matérielles liées à l'utilisation de réseaux de neurones profonds.

2.3.1 Base de données en IRM cérébrale

La donnée médicale est le point central à tout projet d'étude clinique, aussi bien pour identifier l'effet d'un traitement sur un groupe de patients, que pour suivre l'évolution d'une pathologie. C'est d'autant plus le cas pour les méthodes de segmentation dont la qualité dépend des images et annotations utilisées lors de l'apprentissage des modèles sous-jacents, comme pour les travaux présentés dans ce manuscrit. L'étude clinique implique obligatoirement l'acquisition de données (imagerie, biologie, rapports médicaux) relevées sur un patient, rentrant alors dans un cadre législatif très réglementé en France comme en Europe. Ce cadre peut constituer un premier frein à l'accès aux données. Pour respecter les contraintes légales, les images ciblées sont anonymisées de manière à protéger l'identité des patients.

À cela s'ajoute la réglementation sur le stockage des données qui impose au secteur privé de faire appel à des hébergeurs certifiés pour la santé. Avant même de commencer le travail de recherche ou d'analyse, les ressources financières d'un projet impliquant des données médicales sont donc engagées sur les contraintes légales d'accès aux données et de stockage.

L'annotation des examens d'imagerie est un aspect crucial et délicat du processus d'acquisition d'une base de données. En effet, en fonction de la complexité du travail d'identification ou de délimitation nécessaire, l'étude de chaque image peut prendre de quelques secondes à plusieurs heures. Pour l'annotation des structures corticales et sous corticales (plus de 135 régions indépendantes) par exemple, qui sont parfois difficilement délimitées, la tâche prend plusieurs heures pour une seule image. La répétitivité et la complexité du protocole d'annotation rendent la réplication de cette tâche difficile à l'échelle d'une grande base d'imagerie. À ce titre, on observe couramment une forte variabilité entre les annotations de plusieurs experts, ou lorsque qu'un même expert analyse la même image. L'automatisation du travail d'identification visuelle permet donc de faire gagner un temps précieux aux praticiens, mais aussi de proposer un consensus commun.

Des défis de segmentation automatique de structures cérébrales sont régulièrement proposés tous les ans, généralement dans le cadre d'une conférence et facilitent grandement l'accès à des bases d'images annotées. Ces challenges sont organisés pour suivre les avancées

des méthodes automatiques ou comparer les contributions des participants et s'appuient sur des bases de données annotées. Ci-après, nous donnons trois exemples de bases de données de ce type.

OASIS [Marcus et al., 2010a] L'étude OASIS (Open Access Series of Imaging Studies) a pour but de rendre gratuitement accessible à la communauté scientifique, des jeux de données de neuro-imagerie. Pour cela des données sont collectées à travers plusieurs centres dans divers pays, puis partagées à travers une plateforme en ligne. L'étude "Cross sectional" que nous avons utilisée dans ce travail de thèse, est une étude longitudinale sur 150 patient de 60 à 96 ans, dont l'imagerie est effectuée en IRM pondéré T1. Chacun des patients a été scanné deux fois ou plus, pour un total de 373 images. Bien que cette base ne possède pas d'annotation manuelle, elle peut être exploitée dans un objectif de validation visuelle d'un algorithme, ou encore dans un cadre semi-supervisé.

MICCAI 2012 Lors de la conférence MICCAI organisée en 2012, s'est tenu le workshop d'annotation multi-atlas, où les organisateurs ont proposé de mesurer l'efficacité des méthodes d'annotation pour les structures corticales et sous-corticales du cerveau, pour un total de 135 régions distinctes. C'est à notre connaissance la base de données en IRM cérébrale possédant le plus de structures annotées manuellement. La base d'entraînement publique utilisée pour la modélisation est composée de 15 IRM en pondéré T1. La base de test sur laquelle les algorithmes proposés ont été évalués, est composée de 20 images annotées manuellement, issues de 15 patients distincts. Les IRM originelles ont été obtenues à partir de l'étude OASIS, puis anonymisées par la suppression de la face. Le pré-traitement consiste en une correction d'inhomogénéité de champs et une ré-orientation des images dans le même espace.

IBSR V2 La base IBSR (Internet Brain Segmentation Repository) version 2, d'imagerie IRM cérébrale en pondéré T1, regroupe un ensemble de 18 images dont 39 structures sont annotées manuellement par des experts.

L'IRM possède également des spécificités qu'il faut prendre en compte, par exemple lors du pré-traitement des données.

2.3.2 Spécificités de l'IRM pour le cérébrale

L'IRM est considérée comme l'examen de référence pour l'imagerie anatomique des structures cérébrales. Ce choix est justifié par un meilleur contraste qu'en tomodensitométrie (TDM), l'autre examen de référence en neurologie. La TDM est toutefois plus accessible que l'IRM en raison d'un coût d'acquisition et de maintenance plus faible. L'examen TDM est souvent prescrit en première instance lors du diagnostic, mais se trouve limité lorsqu'une exploration précise des tissus doit avoir lieu. On peut observer dans la figure 2.6 la différence de contraste et de résolution entre une IRM et une image TDM.

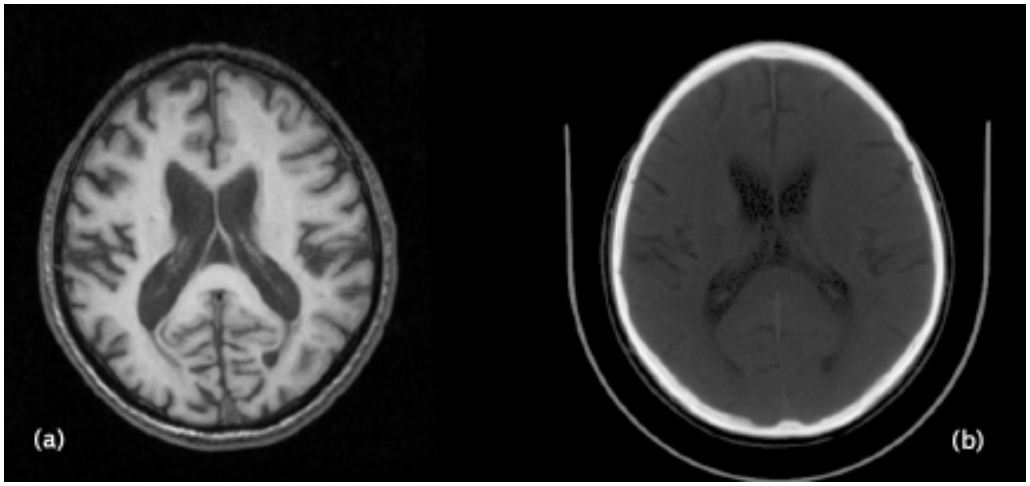


FIGURE 2.6 – Comparaison d'une image cérébrale acquise en TDM (b) recalée sur l'IRM (a). Source [Kuczyński et al., 2010].

En imagerie médicale, il est souhaité que pour une modalité (IRM, TDM, US) donnée, les tissus partageant des propriétés anatomiques similaires partagent également les mêmes intensités, afin de simplifier leur étude et comparaison. L'imagerie TDM fait correspondre les valeurs d'intensité dans l'image à l'échelle Hounsfield qui est standardisée. En théorie on doit donc retrouver les mêmes intensités pour un même patient pour des acquisition faites sur des machines de plusieurs constructeurs. L'IRM qui est une imagerie de contraste relatif, ne dispose pas des mêmes propriétés, à savoir que pour un même patient on n'obtient pas les mêmes images sur plusieurs machines. C'est problématique car les systèmes d'aide à la décision sont souvent paramétrés et testés en se basant sur des bases de données ne couvrant pas toutes les marques et modèles disponibles. Pour corriger cette lacune, des méthodes de normalisation de l'intensité ont été proposées [Madabhushi and Udupa, 2006, Nyul et al., 2000, Collewet et al., 2004] et sont mises en oeuvre lors de l'étape de pré-traitement, après acquisition. La plus simple étant la normalisation z-score, qui consiste à centrer et réduire les données. On peut aussi appliquer une normalisation linéaire, bien que cela ne soit en général pas adapté car trop simpliste. On utilise de nos jours des méthodes basées sur l'alignement des intensités vers une distribution cible [Nyul et al., 2000] (histogram matching). Toutefois, l'inconsistance des intensités des IRM en fonction des machines et des constructeurs est toujours un problème ouvert, qui complique le développement de solutions généralisables sur tous types d'équipements.

Dans de nombreuses modalités d'imagerie médicale, il est courant d'observer l'apparition d'artefacts au cours de l'acquisition (figure 2.7). Ils peuvent être liés à un mouvement du patient, du bruit ou encore une pièce métallique comme une vis. Certaines applications cliniques comme la cardiologie ou la pneumologie sont plus affectées par les artefacts de mouvement, en raison du déplacement des organes comme les poumons et le coeur. Ces artefacts compliquent l'interprétation de l'image en créant des incertitudes, ils peuvent parfois être corrigés en fonction de leurs positions et volumes.

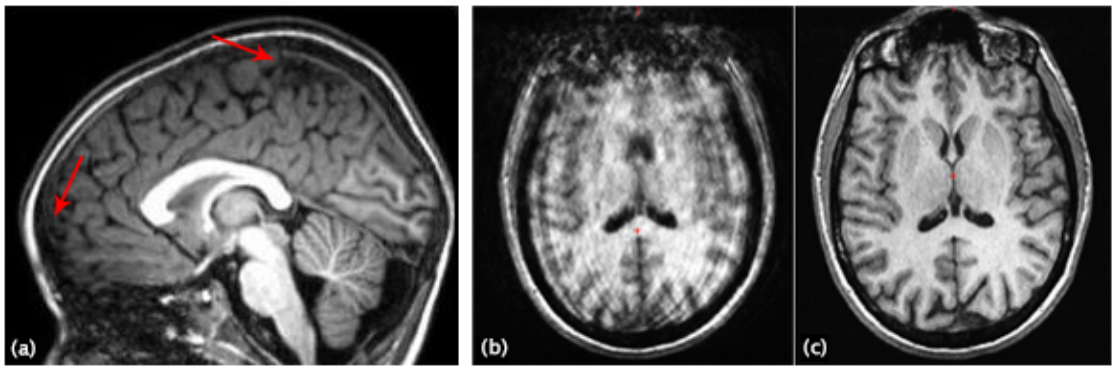


FIGURE 2.7 – Exemple d'IRM présentant des artefacts d'inhomogénéité de champ (a) et de mouvement avant (b) et après correction (c). Source [Phan et al., 2017], Janet Cochrane Miller.

L'IRM est aussi affectée par un autre type d'artefact, appelé inhomogénéité de champ. Cette dernière se créée du fait que le champ magnétique traversant les tissus du patient sera atténué de façon inhomogène, variant en fonction des organes, de la position du patient et de l'aimant. Visuellement, l'inhomogénéité de champ s'observe par des zones de l'image où l'intensité varie non-uniformément (figure 2.7), malgré l'absence de changement anatomique ou fonctionnel des tissus. Plusieurs méthodes [Sled et al., 1998, Belaroussi et al., 2006, Simkó et al., 2019] ont été proposées pour corriger ce problème. La plus répandue, parce qu'automatique et simple d'utilisation, est la normalisation non-paramétrique non-uniforme (N3) [Sled et al., 1998].

L'accès à des ressources de calcul adaptées aux méthodes de segmentation permet de faciliter le développement et l'expérimentation de nouvelles approches. C'est le cas pour les réseaux de neurones profonds qui sont utilisés dans nos travaux.

2.3.3 Ressources de calcul

Les ressources de calcul constituent un des premiers critères lors de l'implémentation d'une solution automatisée d'analyse, d'une part pour faciliter le développement à travers des cycles plus court, mais aussi du fait que les mêmes ressources ne sont pas toujours à disposition pour les utilisateurs finaux (hôpital, cabinet libéral). Les réseaux de neurones convolutifs (section 3.1), principale famille de classifieur automatique abordée dans ce manuscrit, requièrent pour fonctionner efficacement des ressources de calcul spécialisées, en particulier des cartes graphiques, qui se sont très fortement développées au cours des dix dernières années. Une carte graphique est une unité de calcul spécialisée dans l'exécution d'opérations matricielles à faible empreinte mémoire, qu'elle effectue en parallélisant les opérations sur un grand nombre de coeurs. L'utilisation de la carte graphique (GPU) se différencie d'un processeur (CPU) de par son nombre de coeurs (4608 pour une carte Nvidia Titan RTX contre 28 pour un processeur Intel Xeon W-3275) et du cache mémoire qui peut être alloué pour chaque coeur. La carte graphique est donc une ressource spécialisée dans

l'exécution d'un grand nombre de threads parallèles avec peu d'échanges mémoire, tandis que le processeur peut gérer l'exécution de plusieurs threads avec des besoins en mémoire important. Le facteur d'accélération pour des applications numériques qui reposent sur l'algébre linéaire, tel que l'apprentissage automatique ou la vision par ordinateur, est jusqu'à cent fois plus important que l'équivalent sur processeur. Toutefois, l'acquisition de ce type de matériel n'est pas toujours possible, car il demande une infrastructure et des moyens humains pour la maintenir. Des solutions pour porter des RNCs sur des ressources de calcul plus faibles ont été développées pour diminuer les besoins énergétiques et de mémoire de réseaux de neurones convolutifs. En particulier, la recherche automatique d'architecture sous contrainte et la compression du réseau de neurones [Yang et al., 2017, Han et al., 2015],

2.4 Conclusion

Au vu de la progression des approches de segmentation anatomique au cours des dernières années, on voit la qualité des modèles progresser, surtout en apprentissage automatique où le développement rapide des réseaux de neurones profonds est en train de bouleverser l'état de l'art. Toutefois, toutes méthodes confondues, les défis de l'imagerie sont restés les mêmes et s'accentuent parfois en fonction des ressources nécessaires à la bonne implémentation d'un algorithme. Par exemple, pour l'apprentissage de réseaux de neurones, les quantités d'image annotées ne sont pas encore suffisantes pour garantir des niveaux de généralisation suffisants. C'est dans ce contexte qu'émerge des nouvelles thématiques de recherche, pour adapter les avancées de apprentissage profond aux contraintes de l'imagerie médicale.

Dans le chapitre suivant, nous présentons les spécificités des réseaux de neurones profonds, qui sont aujourd'hui utilisés à travers de nombreux travaux de segmentation d'images médicales, et qui sont à la base des travaux réalisés au cours de cette thèse.

Chapitre 3

Apprentissage profond en segmentation d'image

En vision par ordinateur, la tâche de segmentation est souvent associée à un besoin de reconnaissance d'objets. Dans ce sens, elle a longtemps été traitée en deux étapes, avec dans un premier temps l'extraction de caractéristiques, qui formeront l'ensemble de descripteurs de l'image. Puis, dans un second temps, par l'apprentissage d'un modèle supervisé, exploitant les descripteurs extraits à l'étape précédente pour déterminer les règles de décision. La qualité du modèle final dépend donc de ces deux étapes, toutes deux complexes. Le développement des RNCs a simplifié la résolution de problèmes de vision par ordinateur en intégrant l'extraction des descripteurs et la classification dans le même modèle, cela en apprenant automatiquement les caractéristiques visuelles des images à partir des erreurs de classification.

L'imagerie médicale connaît actuellement une effervescence à travers le développement de méthodes basées sur l'apprentissage profond de réseaux de neurones. Des travaux centrés sur l'imagerie [BenTaieb and Hamarneh, 2016, Kervadec et al., 2019, Painchaud et al., 2019] font émerger des solutions aux contraintes propres à la communauté médicale, tels que le manque de données, l'intégration de connaissances de forme ou morphologiques, pour améliorer la robustesse des systèmes d'aide à la décision. Tous ces travaux reposent sur les avancées en apprentissage profond depuis les années 80 jusqu'à aujourd'hui, nous en présentons dans ce chapitre les principales approches.

Nous introduisons toutes les notions utiles à la compréhension et au développement d'outils de segmentation d'images médicales basés sur l'apprentissage de réseaux de neurones profonds. Dans un premier temps, les généralités sont abordées pour introduire les concepts de base (section 3.1). Nous détaillons ensuite dans la section 3.2 les architectures de réseaux de neurones qui ont influencé la communauté, ainsi que les fonctions de coût

propres à l'optimisation du problème de segmentation (section 3.4). Dans les sections 3.5 et 3.6, nous définissons les métriques d'évaluations communément utilisées et les stratégies pour valider les modèles proposés, en prenant en compte les problématiques spécifiques du domaine médical.

3.1 Généralités

Les réseaux de neurones artificiels, aussi appelés plus simplement réseaux de neurones, sont inspirés d'une modélisation du système nerveux. L'exemple caractéristique de cette réflexion est le perceptron, proposé dans les années 60 dans [Rosenblatt, 1958], qui simule le fonctionnement d'un neurone. Ce dernier reçoit plusieurs signaux à travers ses dendrites, pour les transformer en un signal de sortie, qui est finalement transmis à d'autres neurones par le biais de synapses. La puissance synaptique entre plusieurs neurones définit le degré d'interaction, un phénomène que l'on modélise à l'aide des paramètres (ou poids) $\mathbf{w}$. En biologie, les signaux d'entrée qui transitent par les dendrites sont sommés à l'intérieur du neurone, puis un signal de sortie est produit lorsque qu'un certain seuil est atteint. Le but du perceptron est donc d'approximer une fonction f, de sorte à produire une sortie y (catégorie ou valeur réelle) tel que $y = f(\mathbf{x}, \mathbf{w})$ à partir d'un vecteur d'entrée $\mathbf{x}$ et d'un vecteur de paramètres $\mathbf{w}$. Le perceptron est donc de la forme suivante :

$$f(\mathbf{x}, \mathbf{w}) = g(\mathbf{x}^T \mathbf{w} + b), \tag{3.1}$$

avec g une fonction dite d'activation non-linéaire, reproduisant le phénomène de seuillage du neurone et b le paramètre de biais. Les paramètres $\mathbf{w}$ et b sont déterminés itérativement lors de l'apprentissage du modèle, généralement par une méthode d'optimisation par descente de gradient nommée Stochastic Gradient Descent (SGD) [Bottou, 2010].

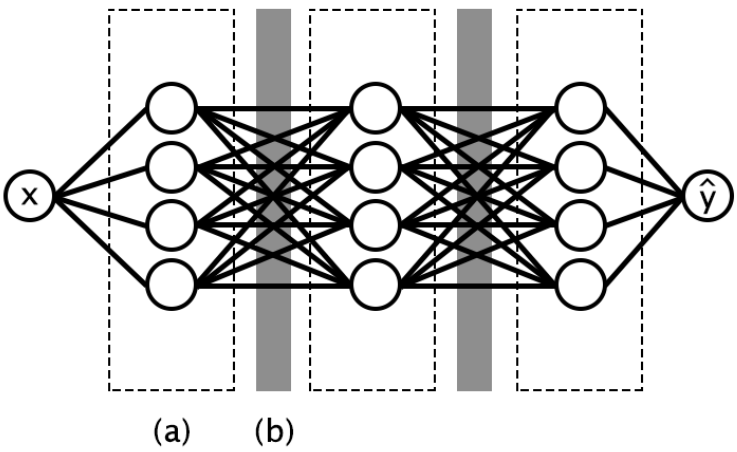


FIGURE 3.1 – Représentation d'un perceptron multi-couche sous la forme d'un graphe, où les couches inter-connectées (a) appliquent successivement transformations linéaires et fonction d'activation non-linéaire (b), afin de produire en sortie du modèle une prédiction $\hat{y}$, pouvant être une valeur réelle ou un vecteur probabiliste en cas de classification.

Dans un réseau de neurones profond, le concept du perceptron est étendu à plusieurs couches (multi-layer perceptron en figure 3.1), où une première couche $f^{(1)}$ reçoit en entrée $\mathbf{x}$ et transmet la sortie vers une deuxième couche $f^{(2)}$, de même pour $f^{(3)}$. $f^{(1)}$, $f^{(2)}$ et $f^{(3)}$ sont trois couches disposant de paramètres individuels, assemblées ensemble pour former un réseau multi-couche (MLP) du type $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$, la profondeur du réseau dépend du nombre de couche de f, d'où l'utilisation des termes apprentissage profond ou réseau de neurones profond. Dans le contexte d'un perceptron multi-couche, on appelle aussi le perceptron couche totalement connectée (fully connected layer), en raison du fait que dans celle-ci, chaque neurone est connecté à tous les neurones de la couche précédente. Tous les modèles de réseaux de neurones explorés dans ce manuscrit sont de type "feedforward", du fait qu'il n'y a pas de retour d'information vers un noeud précédant, constituant ainsi un graphe acyclique dirigé. Les réseaux feedforward représentent les approches les plus répandues en vision par ordinateur, l'exemple principal étant les réseaux de neurones convolutifs (RNC) présentés en section 3.1.3. L'apprentissage d'un réseau de neurones profond nécessite le calcul des gradients de fonctions composées, on utilise pour cela l'algorithme de rétropropagation qui permet une évaluation efficace de ces gradients.

3.1.1 Apprentissage des paramètres

L'apprentissage des paramètres d'un réseau de neurones est semblable à l'optimisation d'un modèle supervisé par descente de gradient. Le processus d'apprentissage se décompose en deux étapes, la propagation des données par l'avant (forward step) puis par l'arrière (backward step) que nous détaillons à la suite.

Propagation avant Soit $\mathbf{x}$ un vecteur dans $\mathbb{R}^m$ issu de la base d'apprentissage, qui est donné en entrée de la première couche et circule ensuite dans les couches cachées i en produisant un signal post-activation $h^{(i)}$, pour finalement sortir un résultat $\hat{y}$ en fin de réseau. Ce résultat est comparé à la vérité terrain y, qui est une catégorie ou une valeur réelle (classification ou régression). Une fonction de coût $L(\hat{y}, y)$ mesure l'erreur de prédiction, qui est la différence entre la prédiction et la vérité, pour produire un coût total C. L'étape de propagation du flux d'information par l'avant est détaillée dans l'algorithme 1.

Propagation arrière Comme on souhaite minimiser l'erreur de prédiction du modèle, les paramètres $\mathbf{w}$ du modèle sont mis à jour de manière à réduire l'erreur. Pour cela, nous évaluons la dérivée de la fonction de coût L par rapport à la sortie $\hat{y}$, on cherche alors $\nabla_{\hat{y}} L(\hat{y}, y)$ le gradient de $L(\hat{y}, y)$ par rapport à $\hat{y}$.

Pour minimiser L, on cherche la direction dans laquelle L diminue le plus rapidement, sachant que le gradient pointe dans la direction opposée à la pente, on fait varier $\hat{y}$ dans

Algorithme 1 : Algorithme de propagation par l'avant (forward propagation ou forward pass) d'un réseau de neurones. Le modèle prend en entrée $\mathbf{x}$ et prédit une sortie $\hat{\mathbf{y}}$ qui est comparée à la vérité terrain $\mathbf{y}$ par $L(\hat{\mathbf{y}}, \mathbf{y})$ la fonction de coût. Le coût total C comprend l'erreur de classification, ainsi qu'un terme de régularisation des paramètres $\Omega(\mathbf{W}, \mathbf{b})$ pondéré par λ. Algorithme traduit de [Goodfellow et al., 2016].

Initialisation : l, la profondeur du réseau
Initialisation : $\mathbf{W}^{(i)}, i \in \{1, ..., l\}$, les matrices de paramètres du modèle
Initialisation : $\mathbf{b}^{(i)}, i \in \{1, ..., l\}$, le vecteur de paramètres de biais du modèle
Initialisation : $\mathbf{x}$, l'entrée du réseau
Initialisation : $\mathbf{y}$, la sortie du réseau
$\mathbf{h}^{(0)} = \mathbf{x}$
for $k = 1$ *to* l **do**
  $\mathbf{a}^{(k)} = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}$
  $\mathbf{h}^{(k)} = g(\mathbf{a}^{(k)})$
end
$\hat{\mathbf{y}} = \mathbf{h}^{(l)}$
$C = L(\hat{\mathbf{y}}, \mathbf{y}) + \lambda\Omega(\mathbf{W}, \mathbf{b})$

la direction opposée au gradient, c'est la descente de gradient :

$$\mathbf{w}' = \mathbf{w} - \epsilon\nabla_{\mathbf{w}} L(\mathbf{x}), \tag{3.2}$$

avec ϵ la vitesse d'apprentissage (learning rate), précisant l'importance de la mise à jour des paramètres. Cette approche converge vers une solution lorsque le gradient devient proche de zéro. L'algorithme de rétropropagation permet une implémentation efficace du calcul du gradient.

Rétropropagation Au cours de la phase de propagation par l'arrière, c'est l'algorithme de rétropropagation proposé dans [Rumelhart et al., 1995], qui permet à l'information de circuler dans le sens opposé. Cette solution propose une évaluation simple du gradient $\nabla_{\hat{\mathbf{y}}} L(\hat{y}, y)$ en décomposant la fonction à dériver. La rétropropagation est basée sur l'application du théorème de dérivation des fonctions composées (chain rule), soit $\mathbf{y} = g(\mathbf{x})$ et $z = f(\mathbf{y})$, avec $g : \mathbb{R}^m \to \mathbb{R}^n$ et $f : \mathbb{R}^n \to \mathbb{R}$ alors pour obtenir le gradient de z par rapport à $\mathbf{x}$: $\nabla_{\mathbf{x}} z = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right)^{\top} \nabla_{\mathbf{y}} z$

La formule des dérivées de fonctions composées permet de calculer les dérivées de la fonction objectif par rapport aux paramètres de chaque couches, en décomposant le gradient couche par couche : en commençant depuis la fonction de coût, puis en propageant au couches précédentes. La propagation arrière ainsi que l'utilisation de la rétropropagation sont décrits dans l'algorithme 2 à la suite.

Le calcul du gradient de chacune des couches nous indique comment les paramètres du modèle doivent changer afin de minimiser l'erreur globale. L'étape de mise à jour des paramètres est déterminée par un algorithme d'optimisation numérique, tel que la descente de gradient stochastique ou une de ses variantes.

Algorithme 2 : Algorithme de rétropropagation (back-propagation) d'un réseau de neurones. À partir de la perte C mesurant l'erreur obtenue lors de la propagation par l'avant, on cherche à obtenir le gradient des fonctions d'activation $\mathbf{a}^{(k)}$ pour toutes les couches k, en partant de la couche de sortie, jusqu'à la couche d'entrée. On peut ainsi déterminer le gradient en fonction des paramètres $\mathbf{W}$ et $\mathbf{b}$ de chacune des couches k, pour minimiser globalement la fonction de coût L. L'estimation des gradients est utilisée pour la mise à jour des paramètres, par un algorithme d'optimisation numérique tel que SGD. Algorithme traduit de [Goodfellow et al., 2016].

À partir de l'erreur C, calculer le gradient de la dernière couche
$\mathbf{q} \leftarrow \nabla_{\hat{\mathbf{y}}} C = \nabla_{\hat{\mathbf{y}}} L(\hat{\mathbf{y}}, \mathbf{y})$
for $k = l$ *to* 1 **do**
  Convertir le gradient en sortie de couche en gradient pré-activation :
  $\mathbf{q} \leftarrow \nabla_{\mathbf{a}^{(k)}} C = \mathbf{q} \odot g'(\mathbf{a}^{(k)})$
  Calculer les gradients pour les paramètres $\mathbf{W}$ et $\mathbf{b}$:
  $\nabla_{\mathbf{b}^{(k)}} C = \mathbf{q} + \lambda \nabla_{\mathbf{b}^{(k)}} \Omega(\mathbf{W}, \mathbf{b})$
  $\nabla_{\mathbf{W}^{(k)}} C = \mathbf{q}\ \mathbf{h}^{(k-1)\top} + \lambda \nabla_{\mathbf{W}^{(k)}} \Omega(\mathbf{W}, \mathbf{b})$
  Propager le gradient à la couche précédente :
  $\mathbf{q} \leftarrow \nabla_{\mathbf{h}^{(k-1)}} C = \mathbf{W}^{(k)\top} \mathbf{q}$
end

Stochastic Gradient Descent (SGD) La descente de gradient stochastique est une méthode d'optimisation itérative très utilisée pour l'apprentissage des réseaux de neurones. La descente de gradient stochastique est une extension de la descente de gradient, adaptée pour des problèmes d'apprentissage supervisé disposant d'une large base de données, compliquant le calcul du gradient en raison de la complexité spatiale linéairement dépendante de la taille de la base. La descente de gradient stochastique émet l'hypothèse que l'on peut approximer le gradient en utilisant un sous-ensemble plus restreint de la base. L'algorithme propose d'échantillonner à chaque itération un nouveau mini-batch B (sous-ensemble de la base) de taille m variant de 1 à plusieurs milliers individus. Plus la taille du mini-batch augmente, plus la variance des mises à jour des paramètres est réduite sous l'effet du moyennage des gradients. L'estimation du gradient $\nabla_{\hat{\mathbf{y}}} L(\hat{\mathbf{y}}, \mathbf{y})$ devient :

$$\nabla_{\hat{\mathbf{y}}} L(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{|B|} \sum_{i \in B} \nabla_{\hat{\mathbf{y}}} L\left(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}\right). \tag{3.3}$$

La mise à jour des paramètres du modèle s'effectue de la façon suivante :

$$\mathbf{W} \leftarrow \mathbf{W} - \epsilon \nabla_W C \tag{3.4}$$

$$\mathbf{b} \leftarrow \mathbf{b} - \epsilon \nabla_b C \tag{3.5}$$

avec ϵ l'hyperparamètre contrôlant la vitesse d'apprentissage. Les paramètres sont mis à jour uniquement une fois que tous les mini-batchs ont été traités, à la fin de l'époque (temps nécessaire au calcul de tous les mini-batchs). La recherche d'une vitesse d'apprentissage optimale est un problème non trivial, une vitesse trop élevée provoque des instabilités

dans le processus d'optimisation, conduisant à une des performances sous-optimales, alors qu'une vitesse trop faible ralentira l'apprentissage inutilement. Les stratégies de mise à jour diminuent la vitesse d'apprentissage en fonction du nombre d'époques, comme la méthode polynomiale [Chen et al., 2016]. Les premières époques bénéficient d'une forte mise à jour, puis sont réduites progressivement pour stabiliser l'apprentissage, jusqu'à varier faiblement à la fin de l'étape d'optimisation.

Pour l'optimisation par SGD, la complexité temporelle ne dépend plus de la taille de la base de données, toutefois il est en général nécessaire d'augmenter le nombre de paramètres pour accroître la capacité du modèle, c'est à dire son aptitude à modéliser une grande variété de fonctions. L'utilisation d'une base de données plus large, oblige également à augmenter le nombre d'itérations nécessaires à la convergence du processus d'optimisation.

SGD est sensible aux conditions d'initialisation des paramètres, qui peuvent affecter la convergence du processus d'optimisation. Pour cette raison des stratégies d'initialisation robustes des paramètres ont été proposées dans [Glorot and Bengio, 2010, He et al., 2016].

3.1.2 Fonctions d'activation

Le choix d'une fonction d'activation non-linéaire g n'est pas évident, face aux classiques fonctions sigmoïde et tangente hyperbolique des alternatives ont été proposées dans [Nair and Hinton, 2010, Maas et al., , He et al., 2015, Goodfellow et al., 2013]. Dans la suite, nous présentons les raisons qui en font des solutions préférées dans la quasi-totalité des travaux en apprentissage profond.

La fonction *sigmoid* est définie pour une entrée x par $g(x) = \frac{1}{1+e^{-x}}$ avec $g : \mathbb{R} \to [0; 1]$, alors que la fonction tangente hyperbolique est définie par $g(x) = \tanh(x)$ avec $g : \mathbb{R} \to [-1; 1]$. Cette dernière pouvant être reformulée à partir de la fonction sigmoïde, on considère *tanh* comme une sigmoïdale. Ces deux fonctions d'activation non-linéaires étaient couramment utilisées dans les débuts de l'apprentissage profond, malgré le fait qu'elles ont tendance à saturer, en éliminant une partie importante du signal. En l'occurrence pour la fonction sigmoïde, la valeur retournée si $x = 100$ ou $x = 100000$ est la même, ce comportement est identique pour les valeurs négatives. Ce phénomène de saturation peut compliquer l'apprentissage par descente de gradient et générer une situation appelée disparition du gradient, où les changements appliqués par le gradient font évoluer faiblement la sortie du réseau. Pour corriger ce défaut, une nouvelle fonction d'activation linéaire par morceaux appelée ReLU a été proposée dans [Nair and Hinton, 2010] pour remplacer les approches sigmoïdales. Elle est définie par $g(x) = max(0, x)$, toutes les valeurs négatives sont seuillées à zéro, alors que les valeurs positives sont retournées à l'identité. ReLU ne souffre pas du problème de disparition du gradient et accélère également la convergence lors de l'apprentissage, une qualité qui est attribuée au fait que les sorties post-activations sont non-creuses, c'est à dire que l'information est répartie quasi-uniformément à travers les poids. D'autres fonctions inspirées de ReLU ont été proposées dans [Maas et al., , He

et al., 2015, Goodfellow et al., 2013], suggérant l'apprentissage de paramètres supplémentaires pour mieux caractériser la non-linéarité, par exemple lorsque x est négatif.

3.1.3 Réseau de Neurones Convolutif (RNC)

Les réseaux de neurones convolutif (RNC) [LeCun and Bengio, 1998] sont une classe de modèles basés sur l'apprentissage profond, dont la particularité est d'employer des opérations de convolution à la place de la couche totalement connectée du perceptron. Les RNCs sont adaptés pour les problèmes d'apprentissage sur des données structurées, organisées sous forme de grille 1D, 2D, et 3D (signal audio, les images ou encore la vidéo). Ce type de réseau de neurones se différencie par l'utilisation d'une opération de convolution, appliquée successivement sur des zones restreintes de l'image, avec un partage des paramètres. Un RNC est habituellement constitué de trois couches principales, la convolution (section 3.1.3.1) suivie par une fonction d'activation non-linéaire de type ReLU (section 3.1.2), puis un sous-échantillonnage des images de descripteurs (section 3.1.3.2).

3.1.3.1 Convolution

Dans un réseau de neurones convolutif, l'opération de convolution est appliquée en fonction de deux éléments : l'entrée I, une matrice multidimensionnelle et le noyau K, une matrice multidimensionnelle contenant les paramètres $\mathbf{W}$ à apprendre. La convolution classique se définit dans le cas discret de la façon suivante :

$$(I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n). \tag{3.6}$$

Le résultat d'une convolution est nommée carte de descripteurs (feature map). C'est l'encodage de l'image d'entrée I par le noyau K, aussi appelé filtre. Pour des raisons pratiques d'implémentation, c'est toutefois la corrélation croisée qui est utilisée sous le nom de couche de convolution dans la majorité des librairies d'apprentissage profond. De même la convolution (du point de vue de l'apprentissage profond) est appliquée sur l'ensemble des cartes de descripteur en entrée. Une couche de convolution en 2D est donc équivalente à une convolution en 3D couvrant tout le volume d'entrée. Lorsque l'on spécifie la taille du noyau de convolution, on désigne donc les dimensions spatiales du noyau en considérant toutes les cartes de cescripteurs en entrée.

Le produit de convolution est une solution commune à plusieurs problèmes de vision par ordinateur (classification, détection, segmentation, reconstruction), toutefois ce sont les concepts de champ récepteur et de partage des paramètres qui justifient que l'utilisation de couches de convolution soit plus efficace pour des données audio-visuelles.

Champ récepteur Les réseaux de neurones profonds classiques, tel que les perceptrons multi-couche (MLP) utilisent dans la couche totalement connectée (fully connected) un paramètre pour décrire chacune des interactions entre les éléments de l'image d'entrée et de sortie de la couche, à l'aide du produit suivant : $\mathbf{W}^{(k)}\mathbf{h}^{(k-1)}$, avec $\mathbf{h}$ la sortie de la couche précédente (cf algorithme 1). Les RNCs exploitent une propriété issue de la biologie, qui spécifie que les neurones répondent uniquement à un stimuli visuel issu d'une zone limitée de la rétine, appelée champ récepteur. Par analogie, le support du noyau de convolution est associé au champ récepteur. Il permet à un réseau de neurones de limiter le champ d'interaction entre les paramètres et l'image d'entrée. Pour cela, on définit la taille du champ récepteur qui sera exploité pour détecter des caractéristiques locales. Par exemple dans la figure 3.2 à gauche, on peut voir l'image d'un chien de race cocker qui dispose de caractéristiques visuelles locales à des zones de l'image, que l'on peut utiliser pour identifier l'animal, autrement dit il n'est pas nécessaire d'utiliser l'image entière pour reconnaître automatiquement un attribut.

La limitation de la taille du champ récepteur implique que la matrice de paramètres $\mathbf{W}$ est également de taille réduite, ce qui diminue l'espace nécessaire au stockage des paramètres (complexité temporelle) et le nombre d'opérations de calcul (complexité spatiale). Le nombre de paramètres appris est déterminé par la taille du champ récepteur (le plus souvent de 3x3), le pas de déplacement (P) entre une position de la fenêtre du noyau et la suivante, ainsi que d'autres hyperparamètres tel que le remplissage (padding) sur les bordures de l'image (voir figure 3.3 à gauche). Pour calculer la taille finale d'une carte de descripteur en sortie d'une couche de convolution, on peut utiliser la formule suivante :

$$(T - C + 2B)/P + 1, \tag{3.7}$$

avec T la taille de l'image, C la taille du champ récepteur, B la bordure de remplissage et P le pas.

Partage des paramètres Le partage de paramètres est une force majeure des RNCs. Elle consiste en la ré-utilisation d'un ensemble de paramètres plusieurs fois. Cette notion prend tout son sens en visualisant l'image centrale de la figure 3.2, où l'on observe une foule avec une vue de haut, pour distinguer les visages qui sont très ressemblants à cette échelle, on peut utiliser à plusieurs endroits le même descripteur visuel. Plus simplement, du fait qu'une caractéristique visuelle peut apparaître à plusieurs endroits dans une même image, ou à des positions différentes dans plusieurs images, il est naturel d'exploiter plusieurs fois le même descripteur.

Cette propriété est bénéfique aux RNCs car elle réduit encore une fois la complexité temporelle et spatiale de la méthode, mais aussi car le classifieur basé sur un RNC sera plus robuste aux invariances par translation. Si par exemple, dans le cas de l'image centrale de la figure 3.2, on souhaite apprendre un classifieur pour identifier les visages mais que l'on ne

dispose pour l'entraînement que de quelques exemples. Alors la ré-utilisation à plusieurs endroits de l'image, d'un descripteur visuel appris sur les exemples, devrait augmenter l'efficacité du modèle sur des nouvelles images du même type.

Le nombre de paramètres utilisés dans une couche totalement connectée est significativement supérieur à celui d'une couche de convolution, car dans la première, le calcul de ce nombre dépend de la taille de l'entrée. Or dans le cas de la convolution, la dimension spatiale de l'entrée n'influence pas le nombre de filtre à apprendre. Par exemple, pour une image d'entrée de taille 224x224 avec trois canaux, une couche totalement connectée à 256 sorties possède 38 535 169 paramètres. Avec la même entrée, un convolution avec 32 filtres de taille 3x3 possède 896 paramètres, grâce à la portée réduite du champ récepteur et du partage des paramètres.

FIGURE 3.2 – Images illustrant la pertinence de l'utilisation de la convolution pour extraire des descripteurs. À gauche l'image d'un cocker, où des champs récepteurs de taille limitée englobent des zones discriminantes de l'image, ce qui démontre la possibilité d'utiliser un nombre réduit de paramètres pour identifier des zones discriminantes. À droite, une vue de haut d'une foule où l'on peut voir des visages grossièrement similaires, justifiant l'intérêt de ré-utiliser des paramètres à plusieurs endroits dans l'image.

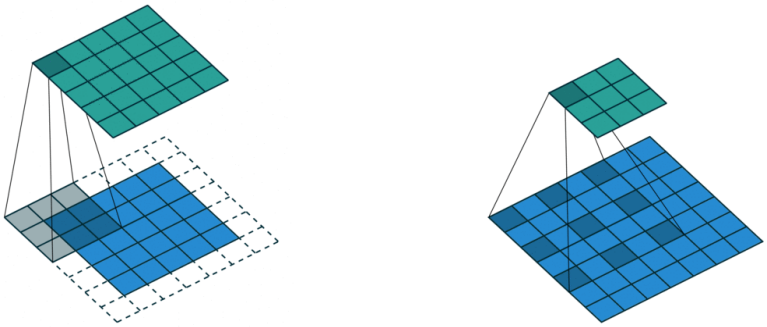


FIGURE 3.3 – À gauche le schéma d'une couche de convolution où un champ récepteur (en gris) glisse sur l'image (en bleu), pour donner une carte de descripteurs (en vert). À droite, une couche de convolution avec dilatation du champ récepteur. Source : Vincent Dumoulin, Francesco Visin.

Récemment, des travaux ont proposé des innovations concernant la couche de convolution, tel que la convolution dilatée [Yu and Koltun, 2015] qui élargit le champ récepteur pour prendre en compte un voisinage plus large, sans pour autant augmenter le nombre de paramètres à apprendre. Un effet de damier appliqué sur le champ récepteur (cf figure 3.3), alterne les zones avec ou sans paramètre appris, un concept repris dans plusieurs travaux connus de segmentation d'images naturelles [Chen et al., 2014, Chen et al., 2017a, Chen et al., 2017b].

Une utilisation alternative de la couche de convolution a été étudiée dans [Lin et al., 2013], où les auteurs proposent une configuration particulière de la fenêtre de convolution, qui a l'effet d'appliquer une transformation linéaire sur l'espace des descripteurs (en profondeur) avec un noyau de taille 1. L'intérêt de la couche de convolution 1x1 est de combiner localement en chaque position de l'entrée, les canaux ou descripteurs du volume, où les pondérations sont apprises automatiquement.

3.1.3.2 Sous-échantillonnage

La couche de sous-échantillonnage (pooling layer) est habituellement la dernière d'un bloc de convolution, elle transforme la sortie de la fonction d'activation en remplaçant certaines valeurs de la carte de descripteurs par des statistiques locales comme le minimum, le maximum ou la moyenne. La statistique la plus communément utilisée est le maximum. Elle est appliquée sur le principe d'une fenêtre glissante de taille T qui se déplace avec un pas P, où la valeur échantillonnée correspond au maximum local à la fenêtre. Cette opération permet progressivement de réduire l'espace des descripteurs pour trouver une représentation compacte de l'information. Cela contribue également à améliorer l'invariance par translation, du fait que l'information d'activation soit résumée localement. Généralement, on fixe la taille de la fenêtre glissante à 2x2 avec un pas de 2, dans ce cas une seule valeur est retenue sur les 4 possibles. Une autre variation possible est de chevaucher les fenêtres en augmentant la taille du champ à 3x3 pour un pas de 2x2. Le maximum n'est pas la seule statistique de sous-échantillonnage, la moyenne est aussi utilisée, bien que moins couramment par exemple dans [Lin et al., 2013], où un échantillonnage global est exploité en substitution d'une couche totalement connectée, pour renforcer l'invariance spatiale et réduire le nombre de paramètres.

3.1.4 Couche softmax

La couche softmax est la dernière couche d'un RNC entraîné pour une tâche de classification ou de segmentation, elle est formulée de la façon suivante :

$$softmax(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}. \tag{3.8}$$

Dans le cadre de la segmentation d'une image 2D de largeur I_L et de hauteur I_H, où l'on souhaite discriminer ℓ régions dans l'image, la sortie d'un RNC produira une matrice

de taille $I_L \times I_H \times \ell$, soit en chaque position de l'image un vecteur probabiliste de taille ℓ. Pour obtenir la segmentation finale, on détermine pour chaque vecteur de $\phi(\mathbf{x})_{ij}$, l'indice de la classe qui possède la valeur la plus forte à l'aide de la fonction *argmax* (illustration dans la figure 3.4), avec $\phi(\mathbf{x})$ les cartes de probabilités données par le réseau pour une image $\mathbf{x}$ d'entrée. La couche softmax (exponentielle normalisée, eq. 3.8) est directement tirée de la régression logistique, où on l'utilise pour la classification en la combinant avec la vraisemblance négative, pour donner l'entropie croisée (section 3.4), la fonction de coût. La fonction softmax est aussi considérée comme une couche de normalisation car elle a l'effet de produire un résultat qui a une interprétation probabiliste, en transformant l'entrée qui peut être un ensemble de descripteurs, en un vecteur dont chaque indice donne la probabilité d'appartenance à la classe associée et dont la somme de tous les éléments est 1.

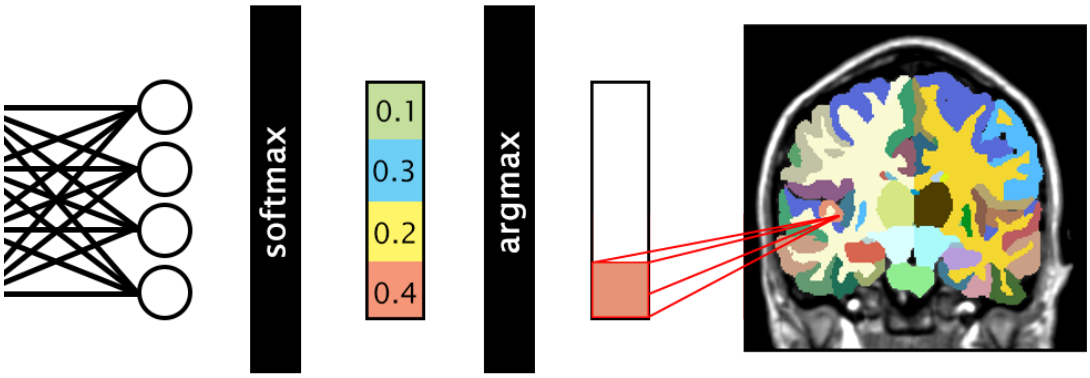


FIGURE 3.4 – Après application de la fonction softmax (dernière couche du réseau), on obtient en sortie du modèle, les cartes de probabilités des régions. Pour trouver la segmentation finale de l'image (à droite), la fonction *argmax* est appliquée pour la recherche de l'indice de la classe la plus probable. Dans ce schéma, on simplifie le problème à la segmentation d'un seul pixel.

3.1.5 Sélection des hyperparamètres

Au cours des processus d'apprentissage et d'inférence, tous les choix de paramètres liés au design de l'architecture du réseau et à l'algorithme d'apprentissage vont influencer les performances finales. Pour ces hyperparamètres (cf tableau 15.2), il n'existe pas de méthode analytique pour déterminer les valeurs optimales, d'autant que le changement d'un peut affecter tous les autres, en faisant un problème combinatoire. Bien que la tendance des dernières années a montré qu'une augmentation du nombre de paramètres améliore les performances pour des bases de données conséquentes, la majorité des choix d'hyperparamètres doit être guidée par l'expérimentation en étudiant les variations des métriques sur les bases d'apprentissage et de validation. Le design d'un RNC est en effet influencé par le caractère du problème à résoudre (classification, régression) et par les données (type, nombre d'exemples annotés, présence de bruit).

description	trop faible	trop élevé
taille du batch	temps d'apprentissage plus long, mauvaise estimation du gradient par SGD	augmente l'espace mémoire GPU
vitesse d'apprentissage	convergence lente, modèle sous-optimal	apprentissage instable
taille du champ récepteur	ne capture aucune information discriminante	augmente le nombre de paramètres inutilement
pas de la fenêtre	$\emptyset$	perte d'information entre deux fenêtres successives
taille de la bordure	$\emptyset$	créer une information inutile au problème à résoudre
taille de la dilatation	$\emptyset$	perte d'information au centre du champ récepteur
taille de la fenêtre de sous-échantillonnage	ne réduit pas assez la taille de l'espace de représentation	perte d'information
taille du pas de la fenêtre	$\emptyset$	perte d'information entre deux fenêtres successives
nombre d'unités dans une couche totalement connectée	limite la capacité de représentation	augmente inutilement le nombre de paramètres

TABLE 3.1 – Liste des hyperparamètres principaux et effets associés à un mauvais réglage (en dessous ou au dessus de leur valeur optimale).

Des méthodes de recherche automatiques (autoML) des hyperparamètres [Snoek et al., 2012, Li et al., 2016] se sont développées ces dernières années, avec l'apparition d'outils comme Ray [Moritz et al., 2018] qui facilitent leurs utilisations en pratique. Ces approches ne sont toutefois pas adaptées au design complet de l'architecture complexe d'un RNC, en raison du nombre d'hyperparamètres à sélectionner, comme le nombre de couches de convolution, la taille du champ récepteur, les connections éventuelles avec d'autres unités. Très récemment, nous avons vu apparaître un nouveau champ de recherche spécialisé dans le design automatique d'architecture de réseaux de neurones, pour la reconnaissance visuelles ou le traitement automatique du langage. Ces travaux [Zoph and Le, 2016, Liu et al., 2018, Sciuto et al., 2019, Xie et al., 2019], ont apporté des avancées impressionnantes en terme de performance, à la fois pour trouver des modèles qui sont aujourd'hui à l'état de l'art, mais aussi pour obtenir des designs avec un nombre de paramètres significativement inférieur à ceux des architectures connues, avec des performances équivalentes ou proches de ces dernières. La recherche automatique d'architecture (Neural Architecture Search) est toutefois un domaine d'étude réputé pour nécessiter beaucoup de ressources matérielles et énergétiques au cours de leur optimisation.

La plupart des RNCs proposés en sciences appliquées, comme en imagerie médicale, s'inspire des architectures existantes pour limiter les efforts de recherche et se concentrer sur

les adaptations spécifiques au domaine. La ré-utilisation des architectures est une solution qui fonctionne en général très bien, en faisant varier quelques paramètres comme la batch normalization au cas par cas. Dans la section suivante, nous présentons les architectures les plus répandues, ainsi que les nouveautés qu'elles apportent.

3.2 Architectures des RNCs

Le terme architecture est maintenant couramment employé par la communauté apprentissage profond, pour décrire un ensemble particulier de couches d'un réseau de neurones. Pour la reconnaissance d'images, les principales architectures sont présentées dans la section 3.2.1. En segmentation d'images, deux travaux ont influencé l'orientation des recherches de ces dernières années : les réseaux de types Fully Convolutional Network (section 3.2.2) et encodeur-décodeur (section 3.2.3). En dernier lieu, nous présentons en section 3.2.4 le transfert d'apprentissage, une méthode qui consiste à ré-utiliser les paramètres appris sur une base de données, pour un autre problème/type de données.

3.2.1 AlexNet, ResNet, DenseNet

Alexnet [Krizhevsky et al., 2012] est la première architecture de RNC à remporter avec une aussi large marge un challenge de classification automatique. C'est probablement le travail qui a re-démocratiser l'utilisation des réseaux de neurones convolutifs en vision par ordinateur, en obtenant pour le challenge ImageNet ILSVRC en 2012, la première place avec une amélioration de l'erreur de +10% par rapport au deuxième, alors que les avancées successives ne dépassaient pas 5% depuis plusieurs années. Bien que l'architecture AlexNet (figure 3.5) soit globalement similaire à LeNet [LeCun and Bengio, 1998], la profondeur du réseau est augmentée, comprend plus de filtres et une succession de couches de convolution avec sous-échantillonnage.

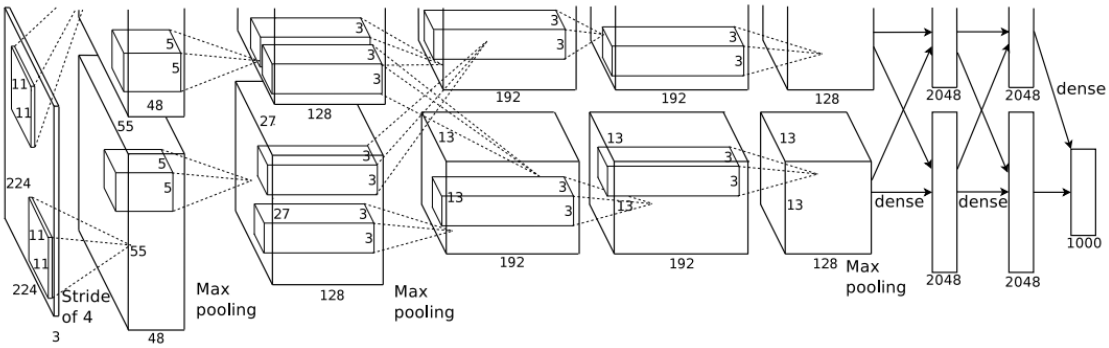


FIGURE 3.5 – Architecture du RNC AlexNet. Source [Krizhevsky et al., 2012]

ResNet [He et al., 2016] Les réseaux résiduels (dits ResNet) ont remporté le challenge de reconnaissance visuelle ILSVRC 2015 avec un taux d'erreur de 3.6%, en entraînant un réseau comprenant 152 couches. La particularité de cette architecture est l'utilisation

de connections résiduelles (skip connection) pour faciliter l'apprentissage de réseaux très profonds (plusieurs dizaines à quelques centaines de couches). Sans cela, en augmentant la profondeur d'un réseau, on peut parfois observer une stagnation des performances, puis une dégradation si l'on continue à ajouter des couches. Pour améliorer la rétropropagation du gradient depuis la sortie vers les premières couches, les auteurs proposent un bloc résiduel qui concatène les cartes de descripteurs issues directement de couches précédentes (figure 3.6). Dans l'architectures ResNet les auteurs utilisent massivement la batch normalization pour s'assurer que l'information propagée a bien une variance non-nulle.

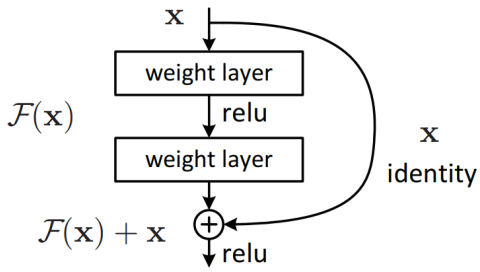


FIGURE 3.6 – Bloc résiduel utilisé dans ResNet pour faciliter la propagation de l'information dans un réseau très profond. Source [He et al., 2016]

DenseNet [Huang et al., 2017] Dans le même esprit que ResNet, DenseNet propose une solution au problème d'optimisation des réseaux profonds, à la différence que les connections résiduelles sont remplacées par des connections denses. Une connection dense connecte chaque couche à toutes les autres couches qui sont à l'intérieur du même bloc, sans créer de cycle. Plus précisément pour une couche c donnée, c recoit en entrée toutes les cartes de descripteurs des couches denses précédentes et de même pour les couches suivantes. Un bloc dense est constitué des deux couches de convolution de taille de champ récepteur 1x1 et 3x3, puis suivies toutes les deux d'une activation ReLU et de batch normalization. L'architecture DenseNet (figure 3.7) favorise la circulation du gradient, la ré-utilisation des paramètres et diminue le nombre de paramètres, en réduisant le nombre de filtres appris dans les blocs denses.

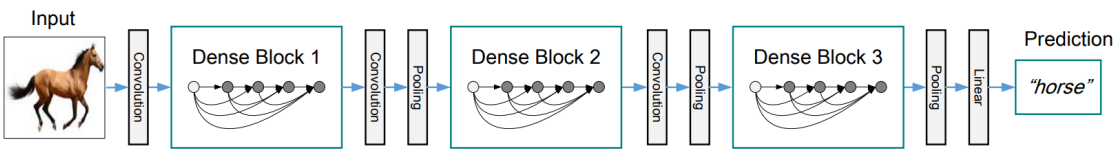


FIGURE 3.7 – Ré-utilisation des cartes de descripteurs dans un réseau DenseNet. Source [Huang et al., 2017]

La classification d'image a été un des premiers domaines à bénéficier des avancées en apprentissage profond. D'autres problèmes comme la détection d'objets et la segmentation

ont aussi été influencés, par des travaux phares comme les réseaux de type fully convolutional et l'architecture encodeur-décodeur.

3.2.2 Fully Convolutional Network (FCN)

Appliquée dans le contexte de la segmentation par RNC, l'approche par patch consiste à faire glisser sur l'image un RNC qui classifie chacun des pixels à partir des patchs extraits. On balaye donc l'image en appliquant un RNC avec une architecture telle que AlexNet ou ResNet sur chaque patch, pour obtenir la carte de segmentation finale. Cependant au fur et à mesure que la taille de l'image grandit, le nombre de patchs à extraire de l'image augmente, augmentant ainsi le nombre de patch à classifier pour segmenter l'image entière. L'objectif de l'approche entièrement convolutive (fully convolutional) FCN [Long et al., 2015, Noh et al., 2015] est de proposer un modèle de segmentation bout-à-bout (end-to-end), qui s'adapte à n'importe quelle taille d'image et produit une carte d'annotations de résolution similaire à l'entrée. Dans un FCN l'image d'entrée est encodée à l'aide de convolutions puis sur-échantillonnée afin de produire une segmentation à la même résolution que l'entrée, cela permet l'inclusion de termes de perte liés à des informations de domaine 2D et 3D. Pour la segmentation 2D/3D, [Milletari et al., 2016] a formulé une perte Dice généralisée (section 3.4.2) robuste aux problèmes avec des volumes de classes déséquilibrés. L'architecture FCN propose également d'adapter les RNCs entraînés pour la classification d'images, en modèle de segmentation à l'aide d'une étape de fine-tuning. L'intérêt des FCNs est de combiner la capacité de représentation visuelle des RNCs avec une solution efficace à l'apprentissage et à l'inférence, réduisant significativement la complexité temporelle de l'étape de segmentation en comparaison à l'approche par patch.

Dans [Long et al., 2015], les auteurs proposent une architecture de RNC pour la segmentation d'image, inspirée de la définition classique d'un RNC (sucession de convolution, activation non-linéaire, sous-échantillonnage et enfin couche totalement connectée. L'appellation "entièrement convolutif" vient du fait que les couches connectées (fully connected), compressant l'information à la fin du réseau, sont remplacées par des couches de convolution pour obtenir une image en sortie du modèle. On peut donc a priori transformer n'importe quelle architecture de RNC pour la classification, pour en faire un FCN dans une finalité de segmentation. Le but de la suppression des couches connectées est de préserver l'information de position et de contexte qui est perdue lorsque les cartes de descripteurs sont aplaties (passage d'une représentation 2D/3D à 1D). Les descripteurs de position sont donc naturellement présents dans les RNC pré-entraînés, l'ajout de couches de convolution à la place des couches connectées permet de remonter vers la résolution de sortie, en reconstruisant progressivement une image de sortie, tout en préservant les informations sémantiques et les détails fins ou grossiers.

L'adaptation d'un RNC classique en FCN n'est toutefois pas optimale, car les sorties sont généralement grossières en raison de la perte de contexte au fur et à mesure

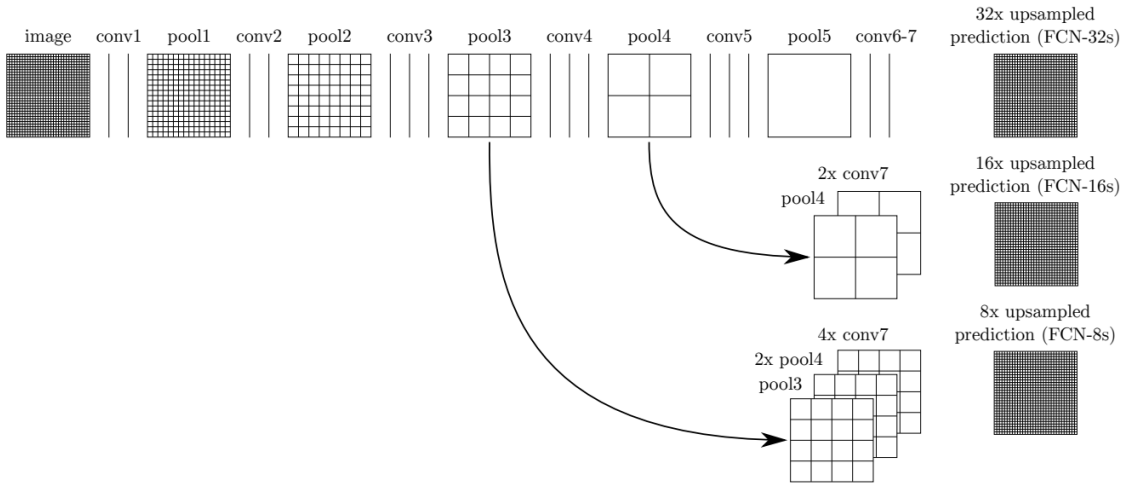


FIGURE 3.8 – Architecture de trois FCNs (un par ligne) où sont décris les résolutions en sortie de chaque couches. Le modèle FCN-32 correspond à une architecture où la dernière carte de descripteurs est sur-échantillonnée 32 fois, pour retrouver la taille de l'image d'entrée. Les modèles FCN-16 et FCN-8 combinent ces mêmes cartes avec d'autres obtenues à des résolutions supérieures dans les couches précédentes, pour retrouver des informations contextuelles et sémantiques. Source [Long et al., 2015]

que la résolution des descripteurs diminue (sous-échantillonnage). Les auteurs proposent donc une architecture avec plusieurs variantes (figure 3.8) FCN-32, FCN-16 et FCN-8, où des couches de déconvolution sont ajoutées pour progressivement reconstruire l'image de sortie. La couche de déconvolution, appelée plus formellement couche de convolution transposée, est une fonction dont le but est de déterminer une image dense à partir d'une version sous-échantillonnée de cette dernière. Cette opération d'obtention d'une image sur-échantillonnée peut être réalisée en apprenant les paramètres de la couche de déconvolution, cependant d'autres solutions ont démontré leur efficacité. La plus évidente est l'application d'une méthode de sur-échantillonnage bilinéaire. Une autre approche comparable à la déconvolution est la couche de unpooling [Noh et al., 2015, Badrinarayanan et al., 2017] (figure 3.9), qui consiste à ré-utiliser la position des activations maximales obtenue lors des étapes de sous-échantillonnage, pour le sur-échantillonnage. Cette dernière approche exploitée dans l'architecture SegNet [Badrinarayanan et al., 2017], ne nécessite aucun apprentissage de paramètres supplémentaires, mais se trouve plus intéressante qu'un échantillonnage bi-linéaire, de par la ré-utilisation des indices qui permettent une meilleure interpolation des cartes de descripteurs.

Une force de l'utilisation des FCNs est la possibilité d'appliquer des fonctions de coûts qui prennent en compte l'aspect global de la segmentation produite par le modèle, pour considérer des relations sémantiques dans l'image. Une adaptation de la métrique Dice (cf section 3.5) a été formulée dans [Milletari et al., 2016] pour comparer le résultat d'un modèle de segmentation à une vérité terrain, sous la forme d'une fonction dérivable.

Les architectures FCN proposées dans [Long et al., 2015] font parties de la famille

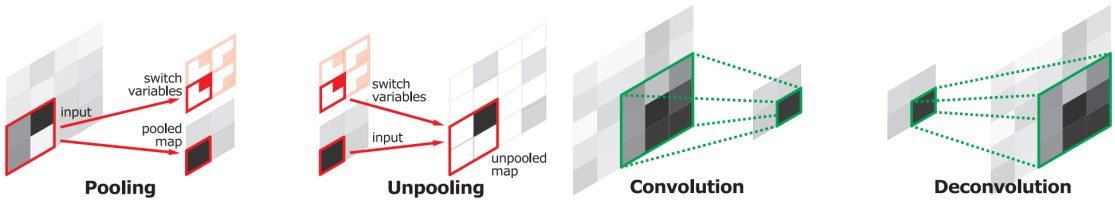


FIGURE 3.9 – Les deux approches principales de sur-échantillonnage utilisées dans les FCNs, la déconvolution et le unpooling. La couche de déconvolution optimise les paramètres des filtres alors que la couche de unpooling utilise les indices des activations obtenues lors du sous-échantillonnage. Source [Noh et al., 2015]

des architectures encodeur-décodeur, dont l'utilisation c'est généralisée en segmentation sémantique d'images.

3.2.3 Encodeur-décodeur

L'architecture encodeur-décodeur est composée de deux parties distinctes, un encodeur et un décodeur, le premier a le rôle d'encoder les attributs visuels et sémantiques en compressant la représentation, tandis que le deuxième reconstruit progressivement les cartes de descripteurs jusqu'à la résolution d'entrée. Plusieurs travaux [Ronneberger et al., 2015, Badrinarayanan et al., 2017, Noh et al., 2015] reposent sur ce formalisme, avec des variations pour l'architecture du réseau encodeur, ainsi que la méthode de reconstruction du décodeur, qui alterne entre couche de déconvolution avec connections résiduelles pour le U-Net et unpooling pour SegNet.

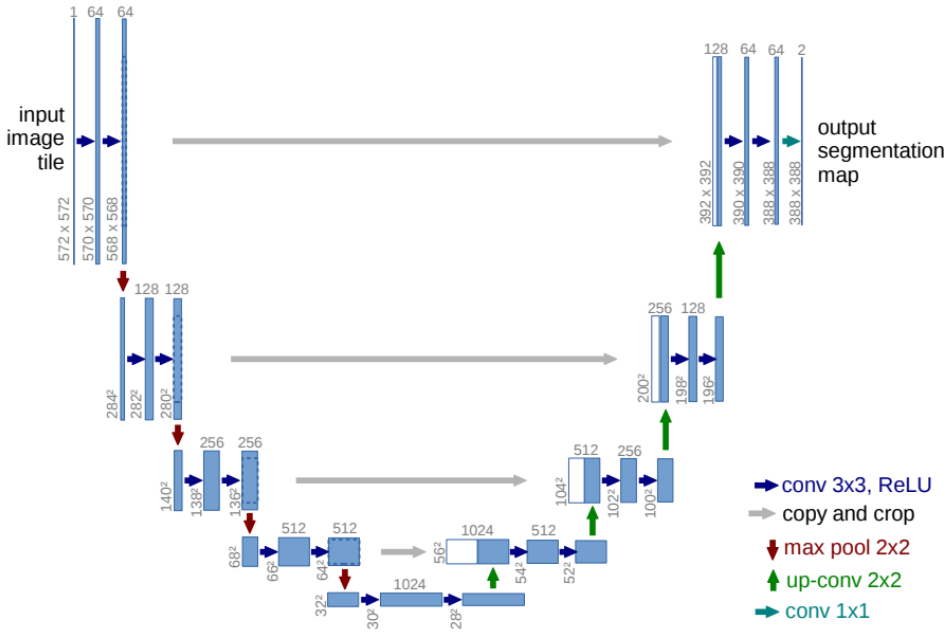


FIGURE 3.10 – Architecture du réseau U-Net, de type encodeur-décodeur. Source [Ronneberger et al., 2015]

Le réseau U-Net [Ronneberger et al., 2015] (figure 3.10) a été développé pour résoudre des problématiques de segmentation bio-médicale et connaît aujourd'hui un succès fort avec

des applications dans de nombreux domaines en biologie, santé et dans l'industrie. Il se distingue par l'utilisation de connections résiduelles à chaque résolution, qui re-transmettent les cartes de descripteurs du bloc encodeur vers le bloc décodeur correspondant, pour améliorer la localisation de descripteurs de haut-niveau.

Les réseaux de type FCN dont l'encodeur est issu d'une architecture connues (ResNet, VGG), utilisent couramment une astuce qui consiste à apprendre les paramètres du réseau sur une base de données annotées, puis à ré-utiliser ces poids pour un problème différent. Cette technique nommée transfert d'apprentissage est utilisable pour la majorité des réseaux de neurones.

3.2.4 Transfert d'apprentissage

Le transfert d'apprentissage est une méthode courante qui consiste à optimiser les paramètres du réseau pour une tâche A, puis à ré-utiliser cette configuration pour résoudre une tâche B. Le transfert d'apprentissage permet dans certaines configurations d'améliorer les performances du modèle B, en bénéficiant d'une base de données A avec de nombreuses annotations, ou encore dans le cas où les tâches A et B sont proches (même type de données), même si on peut observer de nombreux travaux utilisant le transfert d'apprentissage avec succès, où les données issues des tâches A (ex : image naturelle) et B (ex : imagerie médicale) sont clairement distinctes.

En pratique, après avoir entraîné un modèle sur la tâche A, on peut soit le ré-utiliser en tant qu'extracteur de descripteurs, en supprimant la dernière couche totalement connectée, pour utiliser les cartes de caractéristiques en entrée du modèle proposé pour résoudre la tâche B. Une alternative est de remplacer la couche connectée par une nouvelle dont la sortie est adaptée au nombre de classe du problème B, puis d'optimiser ses paramètres en fixant tous les autres, généralement en réduisant la vitesse d'apprentissage par rapport à l'apprentissage original du réseau.

3.3 Régularisation

En apprentissage automatique comme en apprentissage profond, on utilise la régularisation pour améliorer les performances d'un modèle sur des nouvelles données, soit la capacité de généralisation. La solution la plus commune pour cela est d'imposer une norme sur les paramètres du modèle, qui va les contraindre à varier avec une certaine amplitude. D'autres méthodes propres aux réseaux de neurones existent également, comme l'augmentation de données qui consiste à créer des données simulées à partir d'exemples réels.

Dans cette section, nous présentons les principales approches de régularisation, en commençant par la norme des paramètres (section 3.3.1), suivie de l'augmentation de données (section 3.3.2), puis l'arrêt précoce de l'apprentissage (section 3.3.3) et enfin la méthode dropout [Srivastava et al., 2014] (section 3.3.4).

3.3.1 Norme des paramètres

La contrainte de norme des paramètres est une approche classique pour limiter le phénomène de sur-apprentissage d'un modèle supervisé. Les régularisations de norme L^1 et L^2 sont les plus répandues, elles se résument à l'ajout d'un terme de pénalité $\Omega(\mathbf{w})$ à la fonction de coût L, pondéré par un hyperparamétre λ dont la valeur est positive ou nulle, en fonction de l'importance de la pondération. Soit C la fonction objectif globale :

$$C = L(\hat{y}, y) + \lambda\Omega(\mathbf{w}). \tag{3.9}$$

On note que la régularisation est en général uniquement appliquée sur les paramètres $\mathbf{w}$, tandis que le biais reste inchangé. Chaque norme va avoir un rôle spécifique sur l'évolution des paramètres, soit en terme de parcimonie, soit en terme d'amplitude.

Régularisation L^1 Elle est définie par l'application d'une norme 1 sur les paramètres $\mathbf{w}$ d'un réseau de neurones, soit la somme des valeurs absolues de l'ensemble des poids :

$$\Omega(\mathbf{w}) = ||\mathbf{w}||_1 = \sum_i |w_i|. \tag{3.10}$$

La régularisation L^1 produit un effet de parcimonie qui pousse les paramètres les moins utiles vers 0, agissant comme une méthode de sélection automatique de descripteurs, par exemple avec la régression Lasso dans le cas d'un modèle linéaire. Au cours de l'optimisation, un sous-ensemble des poids tend vers 0, indiquant que ces derniers ont peu d'influence dans la prédiction d'une valeur.

Régularisation L^2 C'est la régularisation la plus répandue en apprentissage profond (on la nomme aussi weight decay), elle est définie par le terme de pénalité suivant :

$$\Omega(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2. \tag{3.11}$$

Elle se traduit par l'application d'une norme euclidienne sur les poids du réseau, qui va pénaliser plus particulièrement les paramètres de fortes amplitudes, pour privilégier un lissage globale des valeurs des poids.

3.3.2 Augmentation de données

L'augmentation de données est une forme de régularisation visant à améliorér la généralisation d'un modèle en simulant des versions réalistes des données d'une base. On agit directement sur les entrées et pas sur les poids comme précédemment. En augmentant le nombre de données disponibles pour l'apprentissage d'un modèle, on améliore sa capacité à généraliser ses performances sur de nouvelles observations, dans la mesure où la distribution des données simulées est similaire à la réalité. L'augmentation de données est donc une méthode simple pour améliorer les performances dans les situations où la base d'apprentissage (images ou annotations) est de taille limitée.

En fonction du problème à résoudre, il peut être plus ou moins simple de créer de fausses données. Par exemple dans le cas de la classification, où l'on dispose d'une étiquette associée à une image, des transformations simples de l'image sont utilisées comme la rotation, la mise à l'échelle, le recadrage, la modification de la teinte ou de la saturation ou encore une combinaison de plusieurs de ces transformations. En segmentation d'images, les mêmes transformations peuvent être appliquées sur l'image si elles préservent les annotations du masque. Afin de simuler de nouvelles formes géométriques, il est possible de déformer l'image et le masque d'annotation, avec une méthode telle que la déformation élastique [Simard et al., 2003].

Pour toutes ces transformations, il faut toutefois prendre garde à préserver la finalité de la tâche et ne pas corrompre les données en introduisant un biais. En segmentation d'images cérébrales par exemple, la majorité des méthodes utilisent en pré-traitement un recalage affine des images pour les orienter toutes dans le même espace, il est donc inutile d'appliquer une forte rotation de l'image et du masque sur les données d'entraînement, car ce type d'image ne sera pas observé à l'inférence si le même pré-traitement est utilisé.

3.3.3 Arrêt précoce de l'apprentissage

Lors de l'apprentissage d'un modèle supervisé, on regarde habituellement l'évolution des métriques sur la base d'apprentissage et sur une base de validation externe. On peut alors observer l'amélioration des performances au fil des itérations et s'arrêter lorsqu'un plateau est atteint. Cependant lorsqu'un modèle dont la capacité de représentation est supérieure ou égale au problème, est entraîné sur une base de données, il a tendance à sur-apprendre la base d'apprentissage (sur-apprentissage). On observe dans ce cas, une diminution de l'erreur sur la base d'apprentissage et une augmentation de cette dernière sur la base de validation.

C'est un problème courant pour les réseaux de neurones profonds. Pour le limiter il existe une stratégie simple appelée l'arrêt précoce de l'apprentissage. Elle consiste à arrêter l'apprentissage du réseau lorsque les performances sur la base de validation ne progressent plus depuis plusieurs itérations. On retient alors les poids du modèle issus de la dernière meilleure itération mesurée sur la base de validation. Malgré sa simplicité d'application qui ne nécessite aucun changement majeur pour être mis en application, l'arrêt précoce de l'apprentissage est une méthode de régularisation efficace en apprentissage profond, en combinaison des autres approches comme le weight decay (norme L^2, section 3.3.1) ou le dropout.

3.3.4 Dropout

La couche dropout [Srivastava et al., 2014] est une méthode simple de régularisation, qui limite le phénomène de sur-apprentissage, en favorisant l'activité de la majorité des poids, plutôt qu'une centralisation de l'influence de la prédiction sur le même ensemble de

paramètres. Pour cela, lors de la phase d'apprentissage, la couche dropout agit en annulant aléatoirement l'information en sortie de celle-ci, dans le but de favoriser la création ou le développement d'autres descripteurs. Dans les RNCs, on applique généralement la couche de dropout dans les dernières couches du modèle, avant la couche totalement connectée ou convolution 1x1.

3.4 Fonctions de coût pour la segmentation d'image

Pour l'apprentissage d'un RNC pour la segmentation, les deux fonctions de coût principales sont l'entropie croisée et le dice généralisé. Elles peuvent être utilisées indépendamment ou les deux à la fois, chacun ayant un rôle complémentaire. Nous définissons l'entropie croisée dans la section 3.4.1, puis le dice généralisé dans la section 3.4.2.

3.4.1 Entropie croisée

L'entropie croisée H est une mesure permettant de comparer la similarité entre deux distributions de probabilités discrètes p et q, inspirée de la divergence de Kullback-Leibler. Elle prend comme valeur 0 lorsque p et q se ressemble et $+\infty$ lorsque qu'elles sont différentes. Elle est définie de la façon suivante pour un problème à C classes :

$$H(p,q) = -\sum_{i}^{C} p_i \log q_i \tag{3.12}$$

On l'utilise couramment en classification et en segmentation, comme fonction de coût lors de l'apprentissage du réseau, pour évaluer $H(p,q)$ la différence de distribution entre la vérité terrain et la prédiction. Dans le cas binaire, où on a l'étiquette $y \in \{0,1\}$, $p \in \{y, 1-y\}$ la distribution réelle et $q \in \{\hat{y}, 1-\hat{y}\}$ la distribution estimée, alors l'entropie croisée est égale à :

$$H(p,q) = -y \log \hat{y} - (1-y) \log(1-\hat{y}). \tag{3.13}$$

On observe dans l'équation 3.13 précédente, que seule l'erreur de la classe cible y est prise en compte, le premier et deuxième terme s'annulant en fonction de l'étiquette y. On peut donc la simplifier en supprimant la somme sur les classes C, pour retenir uniquement le terme associé à la classe cible d'une entrée $\mathbf{x}$, on trouve alors :

$$EC(\mathbf{x}, y) = \mathbf{poids}_y \left(-\log(\phi(\mathbf{x})_y)\right). \tag{3.14}$$

avec $\mathbf{poids}_y$ la pondération de la classe y, $\phi(\mathbf{x})$ le vecteur de scores en sortie du réseau (après normalisation softmax). La pondération est utile pour corriger les problèmes d'équilibre de classes qui sont très présents dans les bases de données médicales, où certaines structures anatomiques occupent une surface plus élevée que d'autres. Ce problème

a tendance à diminuer la sensibilité du modèle pour les structures petites et moyennes, si aucune pondération n'est appliquée.

En pratique, on minimise le logarithme de la vraisemblance négative ce qui est équivalent à maximiser la vraisemblance et aussi à minimiser l'entropie croisée. Pour évaluer la vraisemblance, on utilise la fonction softmax (section 3.1.4) en sortie du réseau. En développant l'équation 3.14, on obtient alors la fonction de coût finale :

$$EC(\mathbf{x}, y) = \mathbf{poids}_y \times \left(-\log \left(\frac{\exp(S(\mathbf{x})_y)}{\sum_j \exp(S(\mathbf{x})_j)} \right) \right) \tag{3.15}$$

$$= \mathbf{poids}_y \times \left(-S(\mathbf{x})_y + \log \left(\sum_j \exp(S(\mathbf{x})_j) \right) \right) \tag{3.16}$$

avec $\mathbf{poids}_y$ la pondération de la classe y et $S(\mathbf{x})_i$ la sortie du réseau (pré-softmax) à l'indice i.

Lors de l'apprentissage, l'entropie croisée est moyennée sur le mini-batch $(\mathbf{X}, \mathbf{y})$ de taille N, soit la fonction objectif globale L :

$$L(\mathbf{X}, \mathbf{y}) = \frac{1}{N} \sum_i^N EC(\mathbf{X}_i, \mathbf{y}_i) + \lambda \Omega(\mathbf{w}) \tag{3.17}$$

3.4.2 Dice dérivable

La mesure de similarité Dice (équation 3.19) est un indicateur classique en segmentation médicale, toutefois sa définition originale implique l'utilisation des classes obtenues après recherche des indices maximaux grâce à l'opérateur *argmax*. Ce dernier ne possède pas un gradient utilisable en pratique, une version dérivable multi-classe est proposée dans [Milletari et al., 2016] :

$$soft\ dice = \frac{1}{N} \sum_{(\mathbf{x},\mathbf{y}) \in (\mathbf{X},\mathbf{Y})} \frac{2 \sum_c^C \phi(\mathbf{x})_c \times onehot(\mathbf{y})_c}{\sum_c^C \phi(\mathbf{x})_c + onehot(\mathbf{y})_c}, \tag{3.18}$$

avec $(\mathbf{x}, \mathbf{y})$ le couple image et carte d'annotations issues d'une base de données annotée manuellement, $\phi(\mathbf{x})_c$ la carte de probabilité en sortie du réseau pour la classe c, $onehot(\mathbf{y})$ la carte d'annotations cible encodée sous forme one-hot et N le nombre d'exemples dans la base. L'encodage one-hot consiste à représenter la classe sous la forme d'un vecteur où l'indice correspondant à la classe est égal à 1 et 0 sinon.

En étudiant la variation du numérateur et du dénominateur de l'équation 3.18, on constate que le numérateur tend vers 2 lorsque la segmentation est proche de la vérité terrain et vers 0 dans le cas contraire, de même pour le dénominateur. La fonction de coût $soft\ dice$ varie donc entre 0 et 1, comme la fonction Dice originale et peut s'utiliser seule ou en complément d'une autre fonction, telle que l'entropie croisée.

3.5 Évaluation

L'évaluation d'un algorithme de segmentation est l'étape principale permettant de comprendre les forces et faiblesses de ce dernier. Les métriques utilisées peuvent mettre en avant des performances à l'échelle globale, pour un patient ou une structure donnée, ce qui est utile dans l'orientation des travaux de recherches. L'indice de similarité Dice (section 3.5.1) et les distances surfaciques (section 3.5.2) sont les principales mesures employées en segmentation d'images médicales pour quantifier les performances.

3.5.1 Dice

Le coefficient Sørensen–Dice (aussi appelé F-score) est un indicateur pour mesurer la similarité de deux ensembles (cf figure 3.11 à gauche). Il est très largement utilisé en imagerie médicale comme mesure de qualité globale de segmentation. Pour deux ensembles X et Y, le Dice est donné par la formule suivante :

$$dice(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|}. \tag{3.19}$$

Même si il est utilisé dans de nombreux travaux de segmentation, il n'est pas rare que cette mesure soit complétée par d'autres, telle que la distance surfacique, qui caractérise l'amplitude de l'erreur.

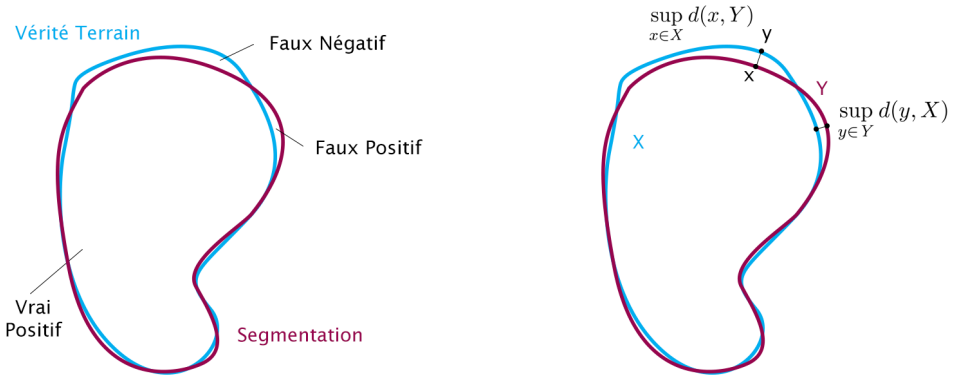


FIGURE 3.11 – Illustration du calcul du Dice à gauche, à travers l'intersection d'ensembles, si l'ensemble des points faux négatifs et faux positifs sont vides, alors les segmentations sont alignées et le Dice vaut 1. À droite, explication du calcul de la distance de Hausdorff, qui consiste à recherche la valeur maximale des distances minimales séparant deux points x et y avec $x \in X$ et $y \in Y$.

3.5.2 Distance surfacique

La distance surfacique moyenne d_{dsm} et la distance de Hausdorff d_H sont les deux mesures de surfaces pour quantifier la dissimilarité entre deux ensembles X et Y. Alors que la distance surfacique moyenne indique globalement à quelle distance un masque binaire est aligné par rapport à un autre, la distance de Hausdorff est beaucoup plus pénalisante

en retenant la distance maximale qui les séparent (cf figure 3.11 à droite). C'est un outil pratique pour détecter une anomalie de segmentation, tel qu'un petit groupe de pixels mal identifié, ou encore une structure éloignée de sa véritable position. Les mesures de distance surfacique sont définies de la façon suivante :

$$d_H(X,Y) = \max\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(y,X)\}, \tag{3.20}$$

$$d_{dsm}(X,Y) = \frac{1}{2}\left(\sum_{x\in X}\frac{d(x,Y)}{|X|} + \sum_{y\in Y}\frac{d(y,X)}{|Y|}\right), \tag{3.21}$$

avec d la distance minimale entre un point et un ensemble.

À ces métriques de performances quantitatives, il est toujours prudent d'associer une analyse visuelle des images, car certains défauts visuels sont difficilement détectés par ces mesures. Même si les métriques de performances telles que le Dice et la distance de Hausdorff forment une méthode stable pour identifier des problèmes de segmentation et les spécificités d'algorithmes automatiques, les données exploitées lors de l'apprentissage ont aussi une forte influence sur les résultats.

3.6 Données

Les nombreux aspects qui caractérisent la qualité d'un jeu de données entrent en compte lors de la conception d'une méthode de segmentation automatique. Face aux problématiques du domaine médical, nous mettons en place des protocoles de séparation des données pour mieux estimer les performances de ces méthodes.

Un sujet central en apprentissage automatique est la capacité d'un modèle entraîné pour une tâche, à fournir des performances qui sont équivalentes sur un nouvel ensemble de données jamais observé, on appelle cette propriété la généralisation. Des stratégies de séparation des données sont mises en place pour estimer au mieux la capacité de généralisation d'une approche. La plus simple consiste à séparer la base de données globale en trois sous-ensembles de tailles variables :

— La base d'entraînement qui contient les données (images et annotations) utilisées exclusivement pour l'optimisation des paramètres propres à l'algorithme et le calcul des performances de la phase d'apprentissage.

— La base de validation qui contient une portion plus faible de données que pour l'entraînement. Elle est utilisée exclusivement pour observer les performances du modèle à la fin d'une itération d'apprentissage. Les mesures propres à la base de validation démontrent la capacité de généralisation sur des données jamais observées, on l'utilise dans le cadre des RNCs pour identifier le phénomène de sur-apprentissage.

— La base de test est utilisée pour valider les performances sur des données annexes à la fin de la période d'apprentissage. On minimise au maximum l'accès à cette base

lors du développement d'un modèle pour limiter l'identification d'hyperparamètres propres à celle-ci. C'est habituellement cette base qui est utilisée lors des challenges d'imagerie médicale pour comparer équitablement les performances des compétiteurs.

Le sur-apprentissage est un problème courant en apprentissage automatique, très répandu en apprentissage profond du fait de la capacité souvent élevé des modèles. Cela s'observe par une marge importante entre les performances sur la base d'apprentissage et de validation, qui témoigne que le modèle sur-apprend les données d'apprentissage, au détriment de la distribution générale des données. Les méthodes de régularisation présentées en section 3.3 sont des solutions efficaces au problème de sur-apprentissage.

Le domaine de la segmentation cérébrale possède également des problématiques spécifiques. En plus d'une complexité d'annotation des images plus élevée que pour la majorité des problèmes de segmentation, en raison du nombre de structures présentes dans le cerveau et parfois au manque de contraste permettant de discerner des frontières anatomiques, l'acquisition de large base de données annotées est très compliquée. Il est donc nécessaire de prendre en compte l'accès aux données annotées comme facteur limitant lors du développement méthodologique, en prenant par exemple appui sur des domaines utilisant plusieurs sources de données lors de l'entraînement, tel que l'apprentissage semi-supervisé. L'augmentation de données, les méthodes de pénalisation et le transfert d'apprentissage, jouent également un rôle important dans la généralisation en segmentation d'images médicales, ces outils font partie intégrante des solutions que l'on peut appliquer rapidement lors du développement. Les fortes variations de volume entre les régions du cerveau sont également un problème pour l'apprentissage de modèles en raison de l'importance prépondérante de certaines régions par rapport à d'autres plus discrètes, des méthodes de pondération des classes doivent alors être intégrées pour limiter ce phénomène.

Ces contraintes propres à l'apprentissage automatique et à l'imagerie cérébrale doivent être prises en compte à l'aide de méthodes développées particulièrement pour y répondre, c'est tout l'objectif des travaux présentés dans ce manuscrit.

3.7 Conclusion

Dans ce chapitre, nous avons détaillé les bases des réseaux de neurones profonds, ainsi que le processus d'apprentissage des paramètres. Puis, nous sommes entrés dans les détails des RNCs, en montrant pourquoi ils sont adaptés pour les problèmes de vision par ordinateur. Les grandes architectures de RNCs pour la classification et de FCNs pour la segmentation ont été présentés en détails. Afin d'améliorer la propriété de généralisation d'un RNC, nous avons listé les méthodes de régularisation couramment mises en place lors de l'apprentissage. Pour optimiser les architectures de segmentation, nous avons décrit les deux fonctions de coût les plus utilisées, ainsi que les méthodes d'évaluation des segmentations produites par un RNC. Finalement, sont énoncées les problématiques majeures des

données dans le milieu médical, qu'il faut prendre en considération lors du développement d'une méthode de traitement.

Chapitre 4

Conclusion

La segmentation basée atlas avec un recalage difféomorphique [Ashburner, 2007, Sdika, 2008, Sdika, 2013, Vercauteren et al., 2009] a longtemps été un choix robuste pour la segmentation d'images médicales, proposant une délimitation cohérente des structures, préservant à la fois les topologies et les relations entre les régions. Pour compléter ces approches, des méthodes d'apprentissage automatique sont utilisées après l'étape de fusion traditionnelle multi-atlas, [Wang and Yushkevich, 2013, Coupé et al., 2011, Sdika, 2010], pour corriger les erreurs de fusion potentielles. Toutefois, ces méthodes multi-atlas nécessitent un temps de calcul élevé du fait de la nécessité de recaler les images d'entrée vers chacun des atlas. En comparaison, les réseaux de neurones convolutifs (RNC) se révèlent à la fois efficaces et précis. Toutefois les deux grandes familles de RNC pour la segmentation, à savoir les approches par patchs et «entièrement convolutif (fully convolutional)» [Long et al., 2015], possèdent chacune leurs propres avantages et inconvénients vis à vis du contexte médical en segmentation.

Dans cette thèse, nous avons développé des méthodologies pour intégrer des *a prioris* en tant qu'entrée du RNC (partie II) ou lors de son apprentissage (partie III). Ces connaissances externes extraites à partir des données, apportent des contraintes de haut-niveau pour régulariser l'ensemble des segmentations que peut produire le modèle, limitant donc la présence de résultats anormaux par rapport à l'anatomie.

Dans la partie suivante, nous développons une approche pour réduire les erreurs de segmentation dans un réseau par patch, en intégrant diverses informations extraites de la base de données.

II Intégration de la connaissance spatiale en segmentation par patch

Chapitre 5

Introduction

Dans cette partie, nous présentons comment nous avons modifié, à travers une approche de segmentation par patch, un réseau de segmentation multi-échelle pour délimiter les structures cérébrales. L'approche de segmentation par patch d'un RNC ne prend pas en compte la position du patch à segmenter, une information pourtant précieuse en l'imagerie médicale, où il existe une certaine invariance anatomique qui pourrait être exploitée. Dans le but de prendre en compte la position du pixel à labeliser dans le cerveau, nous proposons dans cette partie une représentation de la position exploitable par un réseau de neurones convolutif. Nous proposons également d'autres améliorations pour à prendre en compte l'aspect 3D ainsi que l'*a priori* probabiliste extrait d'un atlas moyen.

Dans le chapitre 6 nous présentons l'architecture du RNC (section 3.1.3) multi-échelle par patch appliqué sur des images en deux dimensions. Pour cette architecture, des améliorations sont proposées afin de la rendre plus efficace et simple de réutilisation. Puis nous modifions l'approche multi-échelle dans le but de prendre en compte le contexte autour du patch à classifier. L'apport d'une branche complémentaire, intégrant le vecteur de caractéristiques issu d'un patch 3D est évalué. Dans le chapitre 7, nous abordons l'intégration de l'*a priori* spatial comme une entrée du modèle, exploitant cette donnée lors de l'apprentissage des paramètres du réseau. Cette approche est combinée à l'utilisation d'un atlas probabiliste des structures cérébrales. Le protocole expérimental mis en place lors des expériences est détaillé dans le chapitre 8. Enfin, dans le chapitre 9, nous présentons et commentons les résultats obtenus en vue de mesurer l'intérêt de l'approche proposée, en particulier concernant l'apport de la contrainte spatiale dans la résolution du problème de segmentation de structures cérébrales.

Chapitre 6

Approche par patch multi-résolution pour la segmentation

6.1 Introduction

Dans ce chapitre, nous présentons l'architecture 2D par patch que nous avons développée et que nous appelons par la suite PatchNet. Elle est inspirée des travaux de [Moeskops et al., 2016] qui sont rappelés dans la section 6.2. Par la suite, des modifications du réseau précédent sont suggérées dans la section 6.3 à travers l'utilisation d'une architecture unique pour toutes les résolutions. Dans cette même section, nous proposons également l'intégration d'un patch 3D capturant le volume à l'intérieur du patch à classifier.

6.2 Architecture de référence

Les deux types de méthodes de segmentation par réseau de neurones que nous avons présenté en 3.1 sont l'approche basée sur les patchs et l'approche volumique. En opposition à la première, la seconde produit la segmentation du volume d'entrée complet. Nous choisissons dans cette partie de nous intéresser à la méthode par patch, qui pour un modèle d'apprentissage automatique donné, prédit la probabilité d'appartenance à chacune des structures du pixel central du patch d'entrée. Plusieurs travaux en segmentation d'images cérébrales par apprentissage profond ont proposé d'exploiter l'approche par patch [Lee et al., 2011, Moeskops et al., 2016, de Brebisson and Montana, 2015, Havaei et al., 2017] en utilisant pour modèle de classification, un réseau de neurones convolutif (cf section 3.1.3). Dans [Moeskops et al., 2016], les auteurs suggèrent un modèle pouvant prendre en compte plusieurs échelles d'informations (voir figure 6.1), à travers 3 branches ayant en entrée des patchs de résolutions croissantes (25^2, 51^2, 75^2). Chacune de ces trois branches dispose de fenêtres de convolution spécifiques afin de s'adapter aux résolutions variables

des images d'entrée. Les branches produisent un vecteur de caractéristiques de taille 256 représentant un encodage du patch à la résolution donnée. Ces derniers sont concaténés pour produire le vecteur de probabilités finale d'appartenance à chaque région du cerveau. Le patch à classifier est ainsi encodé avec plusieurs échelles, dans le but d'apporter une meilleure représentation de la structure anatomique.

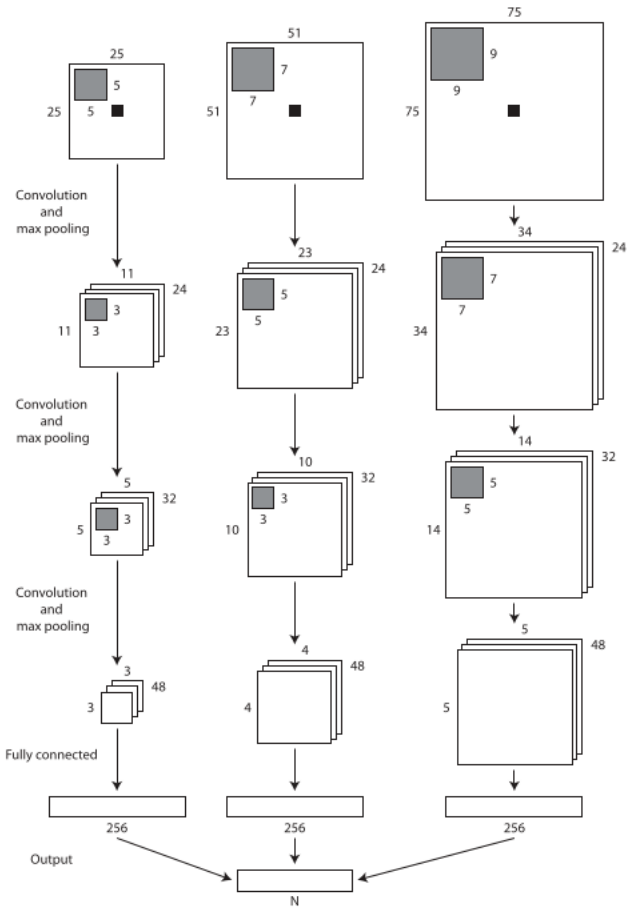


Fig. 1. Schematic overview of the convolutional neural network. The number of output classes, N, was set to 9 (8 tissue classes and background) for the neonatal images, to 8 (7 tissue classes and background) for the ageing adult images, and to 7 (6 tissue classes and background) for the young adult images. After the third convolution layer, max-pooling is only performed for the two largest patch sizes.

FIGURE 6.1 – Architecture du réseau neuronal convolutif multi-échelles proposé par [Moeskops et al., 2016] sur lequel cette partie est basée.

L'architecture proposée par [Moeskops et al., 2016] dont nous nous sommes inspirés par la suite, pose cependant les limitations suivantes :

— En fonction de la taille de l'image d'entrée, le nombre de descripteurs varie. Si on souhaite maintenir un nombre de descripteurs fixe (256 dans [Moeskops et al., 2016]), il est nécessaire d'adapter les paramètres de convolution et le nombre de filtres indépendamment pour chaque branche.

— Du fait de la dépendance de la taille du patch sur la configuration, la complexité algorithmique temporelle (temps d'exécution) et spatiale (nombre de paramètres) va

croître en fonction de la taille, augmentant le nombre de convolution et/ou la taille des noyaux de convolution afin de compresser les descripteurs en entrée.

6.3 Modifications de l'architecture 2D

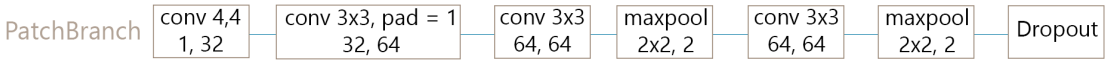


FIGURE 6.2 – Architecture de la branche 2D utilisée pour encoder chacune des résolutions du réseau final PatchNet.

Pour répondre aux limitations énoncées précédemment et renforcer l'efficacité et la flexibilité, nous proposons les modifications suivantes à l'architecture proposée par [Moeskops et al., 2016] :

— Dans la section 6.3.1, pour chaque résolution de patch est exploité un bloc unique (suite de convolution et de sous-échantillonage) aussi appelé branche (Fig. 6.2), ayant une complexité spatiale et temporelle inférieure à la solution de [Moeskops et al., 2016]. La configuration de la branche reste inchangée quelle que soit la résolution du patch d'entrée. Elle est utilisée pour créer un nouveau réseau multi-résolution 2D PatchNet.

— L'intégration d'un patch 3D est détaillé dans la section 6.3.2, où l'on présente la branche spécifique nommée 3dBranch.

6.3.1 Réseau multi-échelle 2D

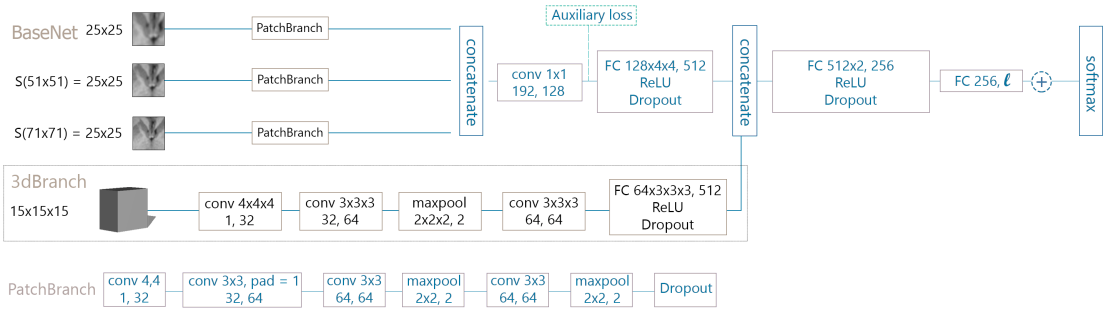


FIGURE 6.3 – Architecture du réseau 2D multi-résolution PatchNet avec la branche 3dBranch intégrant le patch 3D.

Dans la Fig. 6.3, nous décrivons le principal classifieur 2D multi-résolution nommé PatchNet. Il est composé de 3 branches (ou blocs) du réseau modifié présenté Fig. 6.2. Les trois branches sont similaires, avec pour entrée des patchs de taille 25^2 issus de trois échelles. Les deux plus grands patchs (de taille 51^2 et 71^2) sont sous-échantillonnés, de sorte qu'un contexte plus large soit pris en compte sans augmenter le nombre de paramètres. Les vecteurs de caractéristiques produits par les trois branches sont concaténés, puis suivis par

une convolution avec un noyau de taille 1x1, de manière à réduire la dimension de l'espace de représentation. Puis trois couches consécutives entièrement connectées sont utilisées pour produire le vecteur probabiliste de sortie de taille ℓ, après application de la fonction softmax.

Modèle	Branche 1	Branche 2	Branche 3	Classifieur	Total
[Moeskops et al., 2016]	284 096	542 720	930 560	**34 695**	1 792 071
[Ganaye et al., 2018b]	**92 896**	**92 896**	**92 896**	971 495	**1 250 183**

TABLE 6.1 – Nombre de paramètres par branche pour le réseau original [Moeskops et al., 2016] et notre version modifiée [Ganaye et al., 2018b].

Dans le tableau 6.1, nous donnons le nombre de paramètres à optimiser pour l'ensemble des branches et du classifieur final (couche totalement connectée). On constate que dans le réseau proposé par [Moeskops et al., 2016], le nombre de paramètres augmente significativement (facteur x2), en raison du nombre de paramètres des couches convolution, qui sont adaptées à la taille des images d'entrée. La nouvelle branche proposée dans la figure 6.2 reste à nombre de paramètres constant car les patchs sont sous-échantillonnés préalablement. La réduction de la taille des noyaux de convolutions, variant entre 5^2, 7^2 et 9^2 dans [Moeskops et al., 2016], à 4^2 dans la branche proposée, diminue significativement la taille des filtres appris et le nombre d'opérations de calcul. Même si nous faisons le choix de moins compresser l'espace de représentation des caractéristiques dans les couches totalement connectées, le réseau PatchNet contient 30.2% de paramètres de moins que la version homologue de [Moeskops et al., 2016]. L'architecture PatchNet est utilisée comme réseau initial tout au long de cette partie et servira de référence pour mesurer les performances des différentes branches complémentaires proposées.

Les données extraites à plusieurs échelles augmentent le pouvoir discriminant mais ne capturent pas le caractère volumique présent dans l'image d'origine. Nous proposons pour cela une branche optionnelle qui peut être intégrée dans PatchNet.

6.3.2 Intégration du patch 3D

Pour tirer partie de la nature volumique des données, nous avons introduit une branche 3D nommée 3dBranch. Comme illustré sur la figure 6.4, un patch de taille 15^3 centré sur le pixel à classifier est extrait, puis encodé par le biais de convolutions 3D et sous-échantillonné par max-pooling. Enfin les descripteurs sont fusionnés dans le réseau en les concaténant avec la sortie intermédiaire de PatchNet (figure 6.3.1). La taille du patch 3D est choisie de sorte à limiter l'augmentation du nombre de paramètres, due aux convolutions à noyaux 3D.

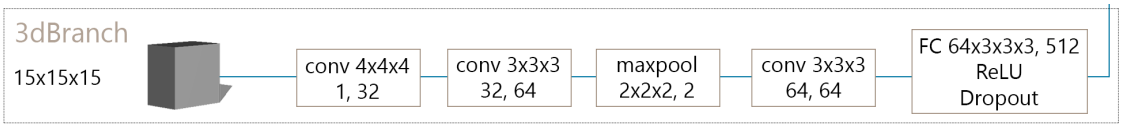


FIGURE 6.4 – Architecture de la branche 3dBranch encodant le patch 3D de taille 15^3.

6.4 Conclusion

Nous avons présenté dans ce chapitre une architecture par patch multi-échelle nommée PatchNet, dans laquelle nous avons proposé d'incorporer une branche permettant la prise en compte du contexte 3D environnant le pixel central d'un patch. Dans le chapitre suivant, nous intégrons à cette architecture des stratégies de prise en compte de l'*a priori* de position du patch.

Chapitre 7

Prise en compte de la connaissance spatiale

7.1 Introduction

Les approches traditionnelles de segmentation basées sur des atlas [Sdika, 2010, Cordier et al., 2016, Wang and Yushkevich, 2013, Sdika, 2015, Heckemann et al., 2006, Klein et al., 2017] bénéficient intrinsèquement d'une cohérence spatiale lors de la phase de fusion, où tous les pixels d'un même voisinage sont pris en compte pour chaque atlas recalés afin de déterminer l'étiquette du pixel central. À la différence des approches basées sur des atlas, les réseaux de neurones convolutifs classifiant des patchs extraits de l'image originale n'exploitent pas la position du patch par rapport au volume. Dans le cas où toutes les images sont recalées dans le même espace, par exemple avec un recalage affine, il est évident que connaître la position relative du patch dans l'image apporte un *a priori* discriminant, allant même jusqu'à exclure la probabilité d'apparition d'une région pour certains cas.

Dans ce chapitre, dans un premier temps nous présentons l'état de l'art des méthodes basées sur l'*a priori* de position des structures anatomiques (section 7.2.1). Puis nous introduisons une méthode d'encodage de la position à travers une image de distance (section 7.2.2), que nous utilisons comme nouvelle donnée d'entrée du RNC PatchNet présenté au chapitre précédent. Nous proposons d'intégrer l'information issue d'un atlas probabiliste (section 7.3) et de la combiner avec la position.

7.2 Représentation de la position

Après avoir présenté les principaux travaux qui ont intégré la position du patch à l'intérieur d'un modèle de segmentation pour l'imagerie médicale, nous expliciterons la méthode que nous proposons.

7.2.1 Intégration de l'information de position pour le médical

L'intégration d'information de position dans une approche de segmentation cérébrale a été explorée dans [Anbeek et al., 2005], pour la délimitation de 8 structures en IRM multi-modalités (T1, pondéré T2). Ce travail repose sur une approche supervisée par plus proche voisin et se base sur des descripteurs d'intensité et de position pour identifier les groupes de pixels partageant des caractéristiques proches. En pratique, pour chaque voxel, les intensités dans les IRMs T1 et pondéré T2 sont extraites avec les coordonnées spatiales (x, y, z), puis ces descripteurs sont donnés en entrée du classifieur par plus proche voisin, afin de trouver le voxel le plus similaire et de propager son étiquette. Son utilisation reste limitée par la simplicité des descripteurs visuels (intensité du voxel) utilisés, cette limitation pourrait être réduite par l'utilisation d'un RNC qui permet l'apprentissage automatique de descripteurs visuels pertinents.

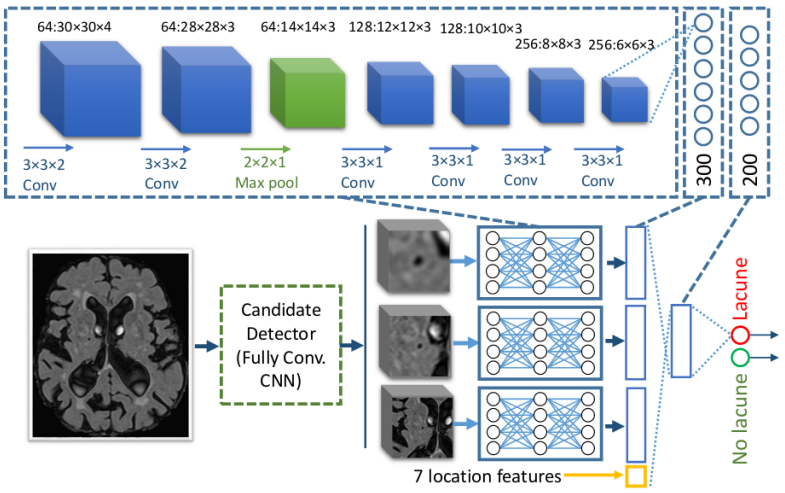


FIGURE 7.1 – Architecture du deuxième réseau neuronal convolutif multi-échelle proposé par [Ghafoorian et al., 2017b] intégrant des descripteurs de position.

Dans [Ghafoorian et al., 2017b], les auteurs présentent un modèle en cascade pour classifier automatiquement la présence de lacunes vasculaires qui peuvent parfois être confondues avec une région anatomique. Dans le pipeline de traitement proposé, un premier RNC suggère des zones candidates à la détection, puis le second réseau 3D affine les résultats en éliminant les faux positifs. La particularité de cette partie est l'intégration dans le deuxième RNC de 7 descripteurs liés à la position dans l'image (x, y, z), ainsi que 4 distances à des structures cérébrales. Ces informations sont fusionnées dans l'avant dernière couche totalement connectée (figure 7.1). Malgré la simplicité de l'approche, elle oblige une pré-annotation manuelle des 4 structures cérébrales utilisées comme repère de distance.

En segmentation cérébrale des structures corticales et sous-corticales, dans [de Brebisson and Montana, 2015] les auteurs ont proposé un réseau par patch multi-plan et multi-échelle (figure 7.2). Ce dernier reçoit également en entrée la distance du patch à 134 points centroïdes qui correspondent aux centres de masse de toutes les régions anatomiques à

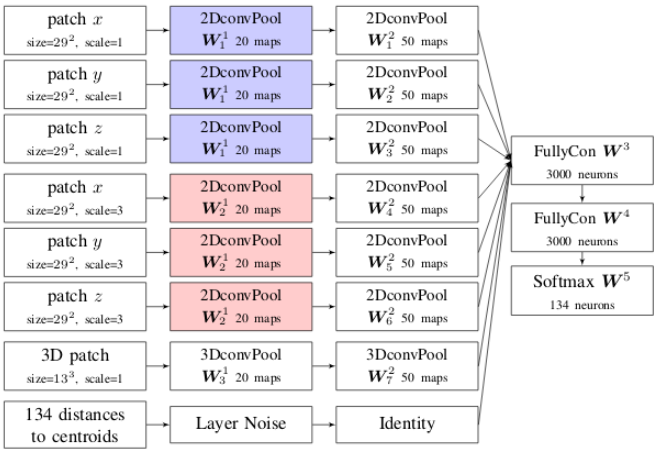


FIGURE 7.2 – Architecture du RNC multi-échelles proposé par [de Brebisson and Montana, 2015], intégrant des distances relatives à des centroïdes de structures.

segmenter. Les centroïdes sont estimés itérativement lors de l'apprentissage des paramètres. Cette méthode nécessite une estimation itérative pour déterminer les centroïdes, ce qui augmente le temps d'apprentissage. Dans certains cas où les régions sont imbriquées (ex structure en oignon), les centroïdes de ces régions seront approximativement à la même position, ce qui n'apporte aucune information complémentaire.

Dans la section suivante, à la différence de [de Brebisson and Montana, 2015, Ghafoorian et al., 2017b], nous proposons une méthode pour encoder et intégrer la connaissance de positions dans n'importe quel système de segmentation automatique par réseau de neurones basé sur des patchs, en utilisant une carte de distance à des points d'intérêts.

7.2.2 Encodage et intégration de la position dans un RNC

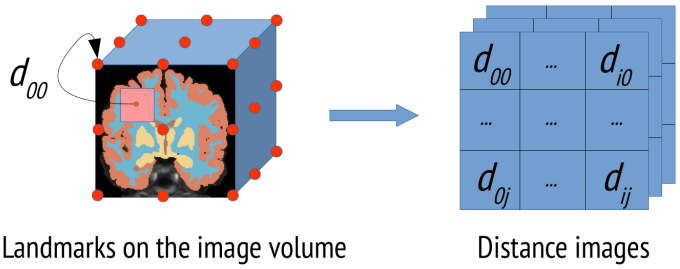


FIGURE 7.3 – Illustration du calcul des images de distance (à droite) à partir des points d'intérêts (points rouges) placés uniformément sur le volume (à gauche).

Dans le but d'encoder la position du pixel central, nous exploitons la distance du patch par rapport à $\mathcal{L}$, un ensemble de points fixes définis uniformément selon chacun des axes du volume d'entrée, avec $\mathcal{L} \in \mathbb{R}^3$. Pour un patch centré à la position x, on évalue $D \in \mathbb{R}^{|\mathcal{L}|}$, le vecteur de distance de x à chaque point d'intérêt dans $\mathcal{L}$. Ces points d'intérêt sont distribués uniformément en fonction de chaque axe, ils ne sont pas associés à des régions du cerveau et ne nécessitent donc pas une pré-segmentation à la différence de [Ghafoorian et al.,

2017b, de Brebisson and Montana, 2015]. Cette méthode est robuste aux cas dégénérés, par exemple dans le cas où deux structures concentriques sont incluses l'une dans l'autre, l'information de position telle que formulée dans [Ghafoorian et al., 2017a, de Brebisson and Montana, 2015] indique le même centroïde pour les deux régions. Comme les points d'intérêts sont répartis sur une grille d'échelle régulière, l'image de distance D peut être représentée comme une image en 3D (voir Fig. 7.3), où chaque pixel représente l'information de position relative à un marqueur spatial. Le calcul de l'image D est illustré dans la figure Fig. 7.3, où l'on voit que pour calculer la valeur $d_{0,0,0}$ de D, la distance euclidienne du patch au repère en position $(0, 0, 0)$ est évaluée. Cet encodage d'un repère spatial sous forme d'image est adopté par la suite, il permet d'extraire des descripteurs visuels à travers l'apprentissage des filtres de convolutions, disposant eux-mêmes d'un pouvoir discriminant optimal pour les données d'imagerie.

Pour normaliser l'image de distance, nous proposons d'appliquer une fonction de base radiale (RBF) sur D avec :

$$rbf(D) = \exp(-\alpha D) \quad \alpha \in \mathbb{R}^+. \tag{7.1}$$

La figure 7.4 montre l'influence de α sur les valeurs à normaliser. Pour des valeurs de coordonnées spatiales comprises entre 0 et 300 (comme dans la base MICCAI12), on observe que $\alpha = 0.01$ est la valeur la plus adaptée, ne créant pas de perte d'information en raison d'une normalisation trop forte, comme pour $\alpha = 0.1$.

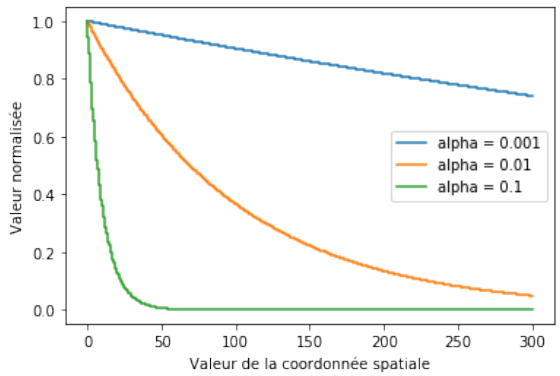


FIGURE 7.4 – Courbes de la fonction de base radiale, pour plusieurs valeurs de α, où l'on observe en ordonnée la valeur normalisée (entre 0 et 1) de la coordonnée spatiale (en abscisse).

Cette image de distance normalisée D sert d'entrée à DistBranch (figure 7.5), la branche du RNC encodant l'information de distance. Elle est composée de convolutions avec des noyaux de tailles variables dont les vecteurs de caractéristiques sont fusionnés, s'inspirant de l'architecture inception de [Szegedy et al., 2017]. Les descripteurs de DistBranch sont fusionnés avec PatchNet dans la seconde couche totalement connectée du réseau principal (cf figure 7.7).

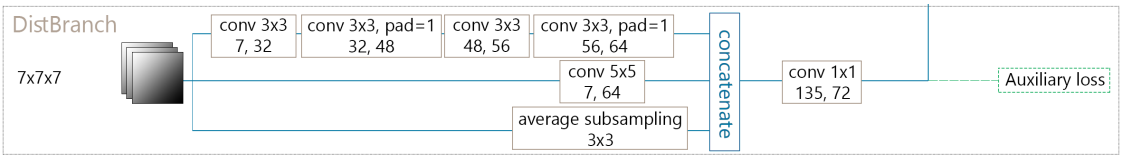


FIGURE 7.5 – Architecture de DistBranch, le bloc intégrant l'image de distance D sous la forme d'une entrée dans le réseau PatchNet.

En complément de cette information de position, nous avons proposé d'intégrer un *a priori* d'appartenance aux structures par le biais d'un atlas probabiliste.

7.3 *a priori* issu d'un atlas probabiliste

FIGURE 7.6 – Architecture de ProbBranch, le bloc intégrant le vecteur de probabilité conditionnelle $\mathbf{p}$ à ℓ classes du patch à classifier, dans le réseau PatchNet.

Un *a priori* d'appartenance des pixels à chaque région peut être modélisé à partir des atlas de la base d'apprentissage, en construisant un atlas probabiliste. Ce dernier donne ainsi la probabilité conditionnelle $\Pr(y \mid x)$ d'appartenance à une région y sachant la position du pixel x, centré sur le patch. L'intuition derrière l'utilisation d'un atlas probabiliste est de capturer la distribution moyenne des régions, pour l'introduire en complément de la prédiction du réseau, dans le but de supprimer les incertitudes liées à des patchs dont les descripteurs ne sont pas assez discriminants. La finalité de cet *a priori* n'est pas la même que pour l'intégration de la position proposée précédemment. Ici on exploite directement la position du patch dans l'atlas probabiliste, pour prendre en compte les chances d'appartenance aux régions qui lui sont associées. Nous utilisons le vecteur probabiliste $\mathbf{p}$ de taille l (soit le nombre de régions à classifier) issu de $\Pr(y \mid x)$ comme nouvelle donnée d'entrée. Ce dernier est introduit à travers une nouvelle branche nommée ProbBranch (figure 7.6) encodant le caractère décisionnel. Pour un voxel donné, le vecteur $\mathbf{p}$ passe par trois couches entièrement connectées de taille ℓ chacune, où ℓ est le nombre de classes. Le vecteur de caractéristiques de la branche est sommé avec la sortie de PatchNet (cf figure 7.7) avant la normalisation softmax (section 3.1.4, figure 3.4). Nous faisons volontairement le choix de conserver la dimension originale de l'espace des descripteurs, puis de sommer (contrairement à concaténer) les caractéristiques avec la sortie principale. En effet pour combiner deux données probabilistes, la somme fait naturellement sens (en échelle log avant softmax). Ces choix sont confirmés par les tests de choix d'architecture effectués au cours du développement.

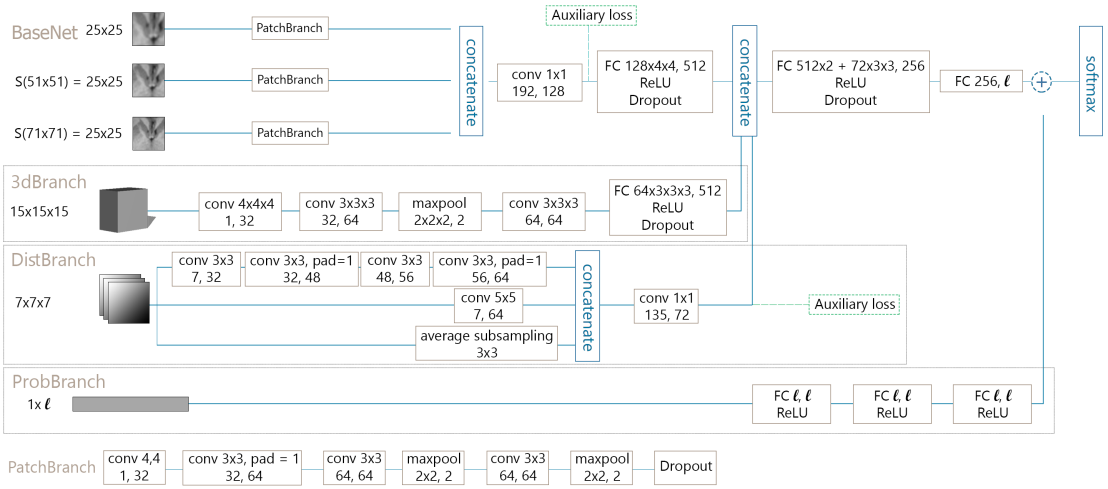


FIGURE 7.7 – Architecture du réseau 2D multi-résolution PatchNet, intégrant les branches 3dBranch, DistBranch et ProbBranch.

7.4 Conclusion

Pour prendre en compte la position du patch à classifier et ainsi limiter les erreurs anatomiques aberrantes, nous introduisons une carte des distances à un ensemble de points de repère. Ces cartes sont considérées comme des images et sont donc naturellement encodables par des convolutions 2D. Contrairement aux travaux précédents ayant intégrés la position dans un RNC par patch, la solution mise en oeuvre est robuste aux cas dégénérés. Notre méthode est en effet adaptée pour des arrangements de structures complexes comme les structures imbriquées et ne requière pas d'annotation manuelle ou itérative.

Nous avons également proposé une approche adaptée à l'invariance anatomique propre à l'imagerie médicale, dans laquelle on construit un atlas probabiliste des régions. Cette carte de probabilité encode l'organisation moyenne des structures à travers une branche dédiée. Pour ces nouvelles sources de données deux branches sont proposées : DistBranch pour encoder les images de distance et ProbBranch pour intégrer l'*a priori* probabiliste. Ces deux branches sont fusionnées dans le réseau 2D multi-résolution PatchNet.

Dans le chapitre suivant, nous présenterons le protocole expérimental qui permettra de tester l'efficacité des approches proposées.

Chapitre 8

Protocole expérimental

8.1 Introduction

Dans ce chapitre nous détaillons le protocole expérimental mis en place pour conduire nos expériences. Nous présentons dans la section 8.2, la base de données cérébrale utilisée dans les expériences. Dans la section 8.3, on expose les détails d'implémentation du réseau et de ses branches, puis la sélection des hyperparamètres.

8.2 Base de données

La base de données cérébrale MICCAI 2012 est composée d'IRM 1.5T de taille 287 x 256 x 256 du projet OASIS, elle a été distribuée lors du défi de segmentation multi-atlas de la conférence MICCAI 2012. Les images ont été annotées manuellement en $\ell = 135$ classes (structures et arrière-plan). L'ensemble de données d'entraînement original (15 images) a été divisé en deux ensembles distincts : apprentissage (10 images) et validation (5 images). L'ensemble de données de test (20 images) est utilisé pour évaluer les performances des modèles sur des données jamais observées.

L'intégration de la position nécessite au minimum que les patients soient orientés dans la même direction pour que les coordonnées correspondent approximativement. Toutes les images ont été réalignées par recalage affine avec FSL Flirt [Jenkinson et al., 2002] vers un atlas de référence sur l'espace de référence MNI T_1 $1 \times 1 \times 1mm$. La boite crânienne a été extraite des images afin de ne considérer que les patchs à l'intérieur du crâne au cours de l'échantillonnage. Ce traitement est nécessaire pour accélérer l'apprentissage, en effet dans la base de données MICCAI 2012 qui est exploitée à travers cette partie, dans l'ensemble d'entraînement l'arrière plan couvre en moyenne 76% de l'image. Enfin pour normaliser les images, la moyenne et l'écart type ont été estimés sur l'ensemble d'apprentissage, toutes

les images ont été finalement centrées et réduites. La taille finale des images après prétraitement est de 182x218x182.

8.3 Détails d'implémentation

La fonction de coût principale utilisée pour optimiser les paramètres de tous les modèles testés est l'entropie croisée. L'optimisation numérique a été réalisée par descente de gradient stochastique, avec un taux d'apprentissage initial $lr_0 = 1e-3$ et un momentum de 0,9. Comme dans [Chen et al., 2016], le taux d'apprentissage (équation 3.4) a été mis à jour à chaque époque avec un coefficient polynomial [Chen et al., 2016] :

$$poly(iter) = lr_0 * \left(1 - \frac{iter}{max_{iter}}\right)^{power}, \tag{8.1}$$

où $iter$ est l'indice de l'itération, max_{iter} le nombre maximum d'itérations et $power = 0.9$ le facteur de diminution. La taille du lot (mini-batch, cf section 3.1.1) est un paramètre qui influence significativement la convergence et donc la performance finale. Dans ces expériences, après des tests, nous avons retenu la taille de 256 patchs par lots.

Au cours des expériences, nous testons incrémentalement l'apport de chacune des branches (3dBranch, distBranch, probBranch) par rapport au modèle testé précédemment. Notre modèle "Full" est composé de toutes les branches et utilise des fonctions de pertes auxiliaires ainsi que l'augmentation de données.

8.3.1 Régularisation et fonction de perte auxiliaire

Pour éviter le sur-apprentissage du modèle, nous avons appliqué une régularisation l_2 sur les paramètres w du réseau, aussi appelée weight decay (section 3.3.1), définie par un terme de pénalité sur la fonction objectif $\lambda \sum_i w_i^2$. Cette contrainte sur la somme des w_i est pondérée par λ, le paramètre contrôlant l'importance de la régularisation dans le problème d'optimisation global. En général λ est fixé à une valeur faible de l'ordre de $1e10^{-3}$. La fonction Dropout [Srivastava et al., 2014] a été utilisée pour la régularisation, celle-ci force à zéro de manière aléatoire certains neurones, favorisant l'utilisation de tous les paramètres.

Pour faciliter la convergence lors de l'apprentissage, lorsque la profondeur (nombre de couches) du réseau augmente, l'utilisation d'une fonction auxiliaire améliore les performances, comme proposé dans [Szegedy et al., 2017]. Une fonction de perte auxiliaire est constituée de deux éléments : une couche totalement connectée suivie de la fonction softmax. Nous testons ainsi par le biais d'une expérience, l'intérêt de l'ajout de deux fonctions de coût auxiliaires dans le réseau PatchNet et dans la branche DistBranch. Lorsque la fonction de perte auxiliaire est utilisée, la fonction de coût globale se compose de l'entropie croisée sur la sortie du réseau et de la somme pondérée des entropies croisées des sorties auxiliaires et de la pénalisation l_2. Sur la base de l'équation 8.2, on définit la fonction de coût de la façon suivante :

$$L(\mathbf{X},\mathbf{y}) = \frac{1}{N}\sum_{i}^{N} \underbrace{EC(\phi^{\mathbf{0}}(\mathbf{X}_i),\mathbf{y}_i)}_{\text{EC principale}} + \alpha_0 \underbrace{EC(\phi^{\mathbf{1}}(\mathbf{X}_i),\mathbf{y}_i)}_{\text{EC auxiliaire}} + \alpha_1 \underbrace{EC(\phi^{\mathbf{2}}(\mathbf{X}_i),\mathbf{y}_i)}_{\text{EC auxiliaire}} + \lambda \underbrace{\Omega(\mathbf{w})}_{\text{Régularisation}} , \tag{8.2}$$

avec $\mathbf{X}$ la matrice contenant les N patchs, $\mathbf{y}$ le vecteur l'étiquette des patchs, $\phi^{\mathbf{0}}(\mathbf{X}_i)$ la sortie principale du réseau, $\phi^{\mathbf{1}}(\mathbf{X}_i)$ et $\phi^{\mathbf{2}}(\mathbf{X}_i)$ les deux sorties auxiliaires et EC l'entropie croisée (section 3.4.1). α_0 et α_1 sont des termes de pondération (fixés manuellement), tout comme λ pour le contrôle de la régularisation des paramètres $\mathbf{w}$ du modèle.

8.3.2 Déséquilibre des classes et augmentation de données

Parce que les régions anatomiques du cerveau ont des volumes variables, l'échantillonnage de la distribution d'origine produit un déséquilibre des classes associées. Le diagramme en barres dans la figure 8.1, montre en effet qu'il existe une répartition inégale dans le nombre de pixels par région, ce qui peut influencer le réseau à mieux classifier certaines régions, au détriment d'autres dont le volume est plus faible. Pour compenser cet effet, nous avons testé un équilibrage les classes en les pondérant inversement dans l'entropie croisée. Afin d'augmenter la variabilité des patchs extraits et améliorer la généralisation de la méthode à de nouvelles images, des augmentations de données aléatoires ont été appliquées sur le patch 2D original, en combinant une mise à l'échelle avec un facteur compris dans la plage [0.9 ; 1.1] et une rotation comprise dans la plage [-10 ; 10] degrés.

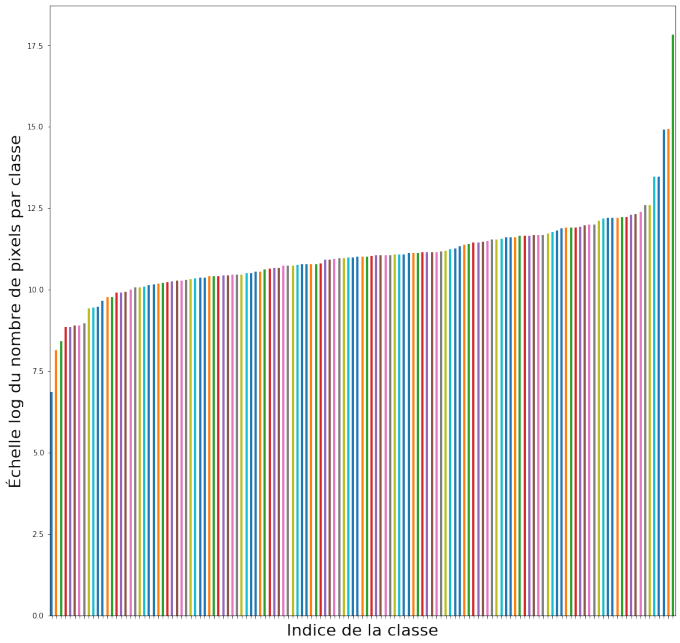


FIGURE 8.1 – Comparaison du volume des structures cérébrales pour les patients de la base MICCAI12. Diagramme en barre du nombre de pixels (échelle log) pour chacune des régions cérébrales de la base MICCAI 2012.

Pour compenser cet effet, nous avons testé un équilibrage les classes en les pondérant

inversement dans l'entropie croisée. Afin d'augmenter la variabilité des patchs extraits et améliorer la généralisation de la méthode à de nouvelles images, des augmentations de données aléatoires ont été appliquées sur le patch 2D original, en combinant une mise à l'échelle avec un facteur compris dans la plage [0.9 ; 1.1] et une rotation comprise dans la plage [-10 ; 10] degrés.

8.3.3 Image de distances

Pour le calcul des cartes de distance, le nombre de points de repère le long de chaque axe a été ajusté à titre expérimental sur l'ensemble de validation, en faisant varier ce paramètre de 3^3 à 10^3 points. Nous avons constaté une stabilisation de la précision avec 7 marqueurs de position par axe, soit un total 7^3 points. Ce paramètre a été conservé pour un bon équilibre entre performances et temps de traitement. En effet en augmentant le nombre de points, on augmente aussi la taille de l'image de distances ainsi que le nombre d'opérations de calcul nécessaires.

Plusieurs représentations de l'image de distance D ont été évaluées : un vecteur 1D, un ensemble d'images 2D et un volume 3D. L'approche 2D s'est révélée être adaptée entre les performances modérées du modèle 1D et le coût non justifiable du modèle 3D (433 192 paramètres pour la branche DistBranch en 3D contre 93 544 pour la version en 2D). Lors de la normalisation de l'image de distances avec la fonction de base radiale, la valeur de α (eq. 7.1) a été définie à $0,01$.

8.4 Conclusion

Avec l'implantation de notre méthode telle que détaillée dans ce chapitre, et appliquée sur la base de données MICCAI 2012, nous obtenons un ensemble de résultats qui sont présentés dans le chapitre suivant.

Chapitre 9

Résultats

9.1 Introduction

Dans ce chapitre nous présentons les résultats des expériences sur l'intégration d'*a priori* liés à la position dans un RNC. Nous détaillons pour chacune des branches qui ont été ajoutées au modèle de référence PatchNet proposé (figure 6.3), l'apport de performance en terme de similarité par rapport à la vérité terrain, les distances surfaciques moyennes et maximales sont également mesurées.

Dans un premier temps, en section 9.2 nous étudions les performances de l'architecture multi-résolution PatchNet par rapport à des modèles de l'état de l'art. À la suite de quoi, toutes les améliorations proposées seront comparées, à savoir :

— le bloc 3dBranch (section 9.3) pour le patch 3D.

— le bloc distBranch (section 9.4) pour intégrer une représentation de la position du patch dans le volume.

— le bloc probBranch (section 9.5) pour prendre en compte la connaissance probabiliste de position des régions.

— la combinaison de toutes les branches précédentes, ainsi que de l'augmentation de données et des fonctions de coût auxiliaires (section 9.6).

9.2 Modèle de référence PatchNet

L'architecture 2D multi-résolution (dite PatchNet) proposée dans la section 6.3.1 est comparée à une approche encodeur-décodeur [Badrinarayanan et al., 2017] nommée SegNet. Cette dernière est entraînée à segmenter coupe par coupe avec les fonctions de coût suivantes : entropie croisée et perte basée sur le Dice. Le tableau 9.1 présente les résultats

obtenus. On note que l'approche 2D SegNet est plus efficace dans la ré-utilisation des filtres de convolution à l'échelle de l'image (52% de paramètres de moins par rapport à PatchNet), mais ne bénéficie pas de la contextualisation multi-échelle des patchs. Malgré ses propriétés intéressantes pour réduire le temps d'apprentissage et d'inférence, on voit dans le tableau 9.1 que le réseau SegNet est moins robuste aux erreurs aberrantes (mesuré par la distance de Hausdorff) que PatchNet. On peut attribuer cette différence à la combinaison des échelles utilisées par PatchNet aidant à réduire les incertitudes locales.

L'ajout de l'augmentation de données (rotation, mise à l'échelle) à PatchNet permet d'atteindre des performances équivalentes à SegNet pour le Dice (0.707 contre 0.708) et significativement meilleures pour les mesures de distances (Hausdorff : 35.47 contre 51.92, MSD : 1.66 contre 2.14).

Modèle	Dice	Hausdorff	MSD	N_{param}
PatchNet	0.694 ± 0.17	40.26 ± 40.12	1.74 ± 2.14	1 249 415
PatchNet (data augmentation)	0.707 ± 0.15	35.47 ± 39.16	1.66 ± 2.03	1 249 415
PatchNet + 3dBranch	0.708 ± 0.15	34.47 ± 30.66	1.54 ± 1.53	2 265 735
SegNet [Badrinarayanan et al., 2017]	0.708 ± 0.16	51.92 ± 40.73	2.14 ± 3.01	599 040

TABLE 9.1 – Mesure de l'apport de l'augmentation de données et de la branche 3dBranch par rapport au modèle de référence PatchNet. Métriques de distance et similarité moyennées sur la base de test. MSD est la distance surfacique moyenne et N_{param} représente le nombre total de paramètre à apprendre du modèle. (moyenne ± écart-type)

9.3 Patch 3D

On rappelle que la branche 3dBranch (section 6.3.2) intègre dans le réseau multi-résolution PatchNet, le vecteur de caractéristiques issu d'un patch 3D de taille 15^3 centré sur celui donné en entrée de PatchNet. Après une suite de convolutions 3D, la branche est concaténée dans le réseau. D'après les résultats de l'expérience présentés dans le tableau 9.1, comparé au réseau PatchNet, l'intégration de l'information volumétrique contribue à l'amélioration de la qualité de la segmentation (+2% de Dice moyen) et à la consistance des prédictions (-14% de Hausdorff moyen). Cependant, même si les résultats obtenus sont similaires à l'approche 2D avec augmentation de données, cette amélioration est pénalisée par l'accroissement du nombre de paramètres (81% de plus par rapport à PatchNet).

9.4 Représentation de la position

L'intégration d'une connaissance de la position du patch par rapport à l'image d'origine a pour objet de réduire les anomalies de segmentation. Pour démontrer l'utilité de la représentation de l'information à travers les expériences, on compare deux approches dans le tableau 9.2. La première intègre les coordonnées brutes (x, y, z) dans PatchNet par une suite de couches totalement connectées, on la nomme PatchNet + Position.

La seconde exploite l'architecture DistBranch en se basant sur les images de distances D, on nomme cette expérience PatchNet + DistBranch.

Modèle	Dice	Hausdorff	MSD	N_{param}
PatchNet	0.694 ± 0.17	40.26 ± 40.12	1.74 ± 2.14	1 249 415
PatchNet + Position	0.703 ± 0.17	15.84 ± 12.04	1.21 ± 0.76	1 337 223
PatchNet + DistBranch	0.720 ± 0.14	10.09 ± 5.41	1.10 ± 0.64	1 508 511

TABLE 9.2 – Mesure de l'apport de l'image de distances contre l'utilisation des coordonnées brutes. Métriques de distance et similarité moyennées sur la base de test. MSD est la distance surfacique moyenne et N_{param} représente le nombre total de paramètre à apprendre du modèle. (moyenne ± écart-type)

D'après les métriques de similarité et distances pour les deux représentations, présentées dans le tableau 9.2, on remarque que l'intégration de la position sous forme de coordonnées brutes corrige bien la distance de Haussdorff (-60% par rapport à PatchNet). L'encodage de la position sous forme d'une image de distances (PatchNet + DistBranch) avec l'utilisation de convolution assure un gain significatif (mesuré par test de student appairé avec un seuil de confiance de 95%) à la fois pour le Dice (+3%) et pour les mesures de distances (Hausdorff : -75%, MSD : -36%) par rapport à PatchNet.

La comparaison des deux approches d'intégration est nette, l'encodage de la position sous forme de cartes de distance est plus performant que l'utilisation des positions brutes, pour le dice moyen (+2.4%), le Hausdorff moyen (-36.3%) et le MSD (-9%).

On peut donc en conclure que l'ajout de la position est bénéfique à la réduction des anomalies au vu de la diminution de la distance de Hausdorf. De même, une amélioration de la qualité globale de segmentation est notée par l'augmentation du dice moyen. On souligne toutefois que c'est la représentation sous forme d'images et l'encodage par convolutions 2D qui sont essentielles pour interpréter au mieux cette information.

9.5 Connaissance probabiliste

Comme présenté en section 7.3, l'organisation moyenne des structures est une connaissance qu'il est possible de capturer par un atlas moyen des cartes de segmentations extrait sur l'ensemble d'apprentissage. On dispose alors, pour tous les pixels x dans le volume de la probabilité conditionnelle d'appartenance de x à chacune des régions anatomiques. Dans le modèle dit ProbBranch, on intègre pour le pixel centré sur le patch, le vecteur probabiliste de taille 135. Pour préserver au mieux l'importance de cet *a priori* dans le réseau, la sortie de ProbBranch est sommée avec la sortie principale du réseau PatchNet (cf figure 7.6) avant application de la fonction softmax.

Le tableau 9.3 compare les résultats obtenus entre le modèle PatchNet et le modèle ProbBranch. On observe que l'ajout de ProbBranch dans PatchNet réduit significativement la distance de Hausdorff moyenne et également l'écart-type, mais n'impacte pas le Dice. Elle ne nécessite pas d'annotation supplémentaire et requière uniquement 4.7% de

Modèle	Dice	Hausdorff	MSD	N_{param}
PatchNet	0.694 ± 0.17	40.26 ± 40.12	1.74 ± 2.14	1 249 415
PatchNet + ProbBranch	0.700 ± 0.17	32.38 ± 36.90	1.50 ± 1.80	1 304 090

TABLE 9.3 – Mesure de l'apport de la branche basée sur un atlas probabiliste (ProbBranch). Métriques de distance et similarité moyennées sur la base de test. MSD est la distance surfacique moyenne et N_{param} représente le nombre total de paramètre à apprendre du modèle. (moyenne ± écart-type)

paramètres supplémentaires. Le ratio coût/bénéfice justifie l'utilisation de cette branche dans un modèle final qui regroupe DistBranch et/ou 3dBranch.

9.6 Combinaison des branches

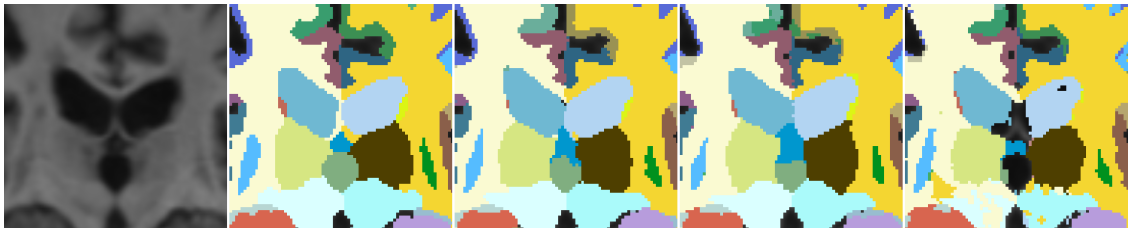
Après avoir mesuré l'influence des trois branches (3dBranch, DistBranch, ProbBranch) sur le RNC multi-échelle PatchNet, nous choisissons à présent de combiner ensemble les branches étudiées, avec l'augmentation de données et les fonctions de coût auxiliaires, afin d'analyser si leurs apports respectifs se cumulent.

Le tableau 9.4 présente les résultats obtenus. De toutes les branches, on peut déjà noter que c'est l'intégration de la distance sous forme d'image par DistBranch, qui apporte les gains les plus importants pour toutes les métriques, et ce avec une augmentation du nombre de paramètres de 20.7%.

En combinant ProbBranch avec DistBranch, on améliore un peu les résultats sans impacter le nombre de paramètres, ce qui suggère que la majorité des erreurs corrigées par ProbBranch sont déjà détectées par DistBranch. C'est finalement le modèle complet "Full" qui démontre les meilleurs résultats. Ce dernier regroupe toutes les branches, l'augmentation de données et les fonctions de coût auxiliaires. L'amélioration de la distance de Hausdorff du modèle complet par rapport à PatchNet + DistBranch est minime (+4.4%), la version "Full" ne justifie pas la complexité spatiale supplémentaire (+127% de paramètres). Cependant l'augmentation du Dice entre ces deux versions du réseau initial est plus intéressante si l'on souhaite parfaire la qualité globale de segmentation mesurée à travers le Dice moyen.

[de Brebisson and Montana, 2015] est le seul à notre connaissance à avoir utilisé une approche de segmentation par patch pour le problème initial à 135 classes, qui propose un modèle composé de 30 millions de paramètres et atteint un Dice moyen de 0.725. En comparaison, notre modèle a 10 fois moins de paramètres, avec un meilleur Dice moyen. La version "Full" de PatchNet aurait été classée 5ème du défi de segmentation multi-atlas à MICCAI 2012, avec un temps de segmentation par image d'environ 9 minutes, comparé à plusieurs heures pour le premier.

La figure 9.1 montre des exemples de cartes de segmentation produites par les modèles entraînés. On remarque une amélioration nette entre PatchNet (e) et PatchNet+DistBranch



(a) (b) (c) (d) (e)

FIGURE 9.1 – Exemples de cartes de segmentation pour plusieurs architectures. Coupe coronale de l'image (a) et ses cartes de segmentation associées : vérité terrain (b), Full (c), PatchNet+DistBranch (d) et PatchNet (e). Les segmentations sont issues d'un patient de la base de test.

Modèle	Dice	Hausdorff	MSD	N_{param}
PatchNet	0.694 ± 0.17	40.26 ± 40.12	1.74 ± 2.14	1 249 415
PatchNet + 3dBranch	0.708 ± 0.15	34.47 ± 30.66	1.54 ± 1.53	2 265 735
PatchNet + DistBranch	0.720 ± 0.14	10.09 ± 5.41	1.10 ± 0.64	1 508 511
PatchNet + ProbBranch	0.700 ± 0.17	32.38 ± 36.90	1.50 ± 1.80	1 304 090
PatchNet + DistBranch + ProbBranch	0.723 ± 0.14	9.95 ± 5.29	1.10 ± 0.65	1 563 186
PatchNet + DistBranch + ProbBranch + 3dBranch	0.733 ± 0.14	9.99 ± 5.63	1.07 ± 0.63	2 847 794
Full (all branches + augmentation + auxiliaire)	0.748 ± 0.14	9.66 ± 5.46	1.00 ± 0.59	2 847 794

TABLE 9.4 – Comparaison de l'apport des différentes branches et de leur association. Métriques de distance et similarité moyennées sur la base de test. MSD est la distance surfacique moyenne et N_{param} représente le nombre total de paramètre à apprendre du modèle. (moyenne ± écart-type)

(d), avec par exemple la correction des zones d'arrière-plan identifiées par le réseau dans les ventricules latéraux et une meilleure continuité des contours de régions.

Le tableau 9.5 permet d'étudier la complexité temporelle de chacun des modèles proposés, en indiquant le temps d'inférence (en minutes) pour une image, ce qui comprend l'extraction des patchs et des entrées propres à toutes les branches. Sans surprise le modèle final est le plus long à traiter une image, avec un temps de 15.2 min. C'est l'utilisation de la branche DistBranch, qui comprend l'extraction des images de distances et les étapes de calcul de la branche, qui augmente le plus le temps de calcul, d'environ 10 minutes par rapport au modèle PachNet. À noter également, un sursaut minime lors de l'utilisation de ProbBranch.

Modèle	Temps d'inférence	N_{param}
PatchNet	3.1	1 249 415
PatchNet + 3dBranch	4.1	2 265 735
PatchNet + DistBranch	13.1	1 508 511
PatchNet + ProbBranch	3.2	1 304 090
PatchNet + DistBranch + ProbBranch	15.0	1 563 186
PatchNet + DistBranch + ProbBranch + 3dBranch	15.2	2 847 794
Full (all branches + augmentation + auxiliaire)	15.2	2 847 794

TABLE 9.5 – Comparaison pour les différentes architectures du temps d'inférence en minute pour une image et nombre de paramètre.

9.7 Étude des cas problématiques

Afin de comprendre l'impact des améliorations proposées sur le problème de segmentation cérébral, nous approfondissons dans cette section l'étude des performances pour le réseau PatchNet et les deux versions proposées : PatchNet + DistBranch, PatchNet Full. On étudie pour cela la variation de la distance de Hausdorff en fonction des patients ou des structures, dans l'objectif de mettre en lumière les cas (patients ou structures) les plus problématiques et d'étudier l'évolution des performances pour ces derniers. La figure 9.2 indique pour chacun des patients de la base MICCAI12, les variations de la distance de Hausdorff toutes structures confondues, cela permet d'étudier globalement le réalisme de la segmentation proposée par les modèles. On constate globalement que quelques patients se détache par l'augmentation de la variance et de la médiane. Toutefois, l'utilisation de la branche DistBranch et du modèle complet apportent une forte baisse et une meilleure stabilité de la distance de Hausdorff et ce pour tous les patients.

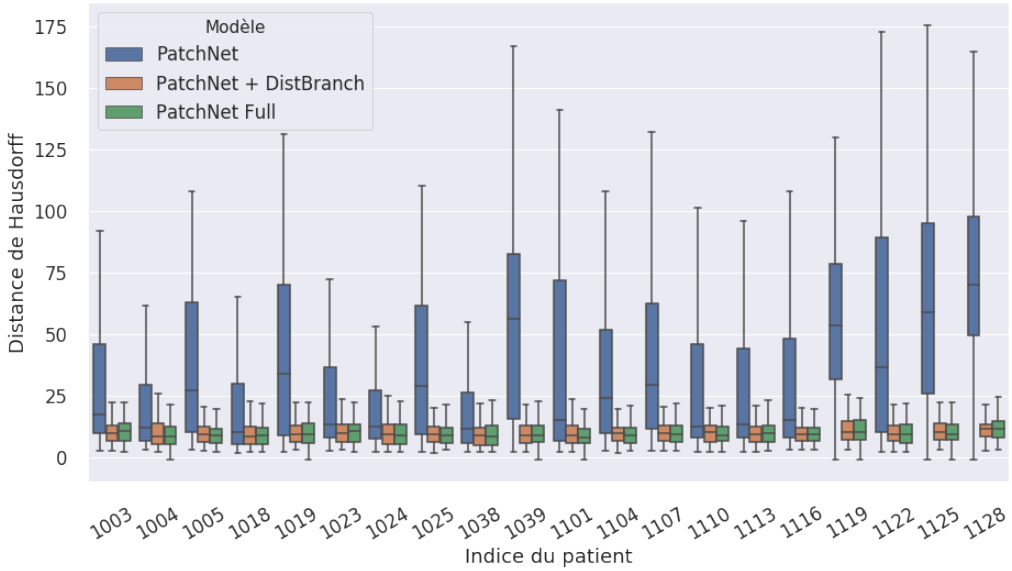


FIGURE 9.2 – Illustration de la dispersion de la distance de Hausdorff, pour toutes les structures de chaque patients de la base de test MICCAI12. On compare les performances du réseau par patch PatchNet, avec l'ajout de la position du patch et l'utilisation de toutes les branches.

On souhaite aussi connaître l'utilité de la méthode à l'échelle anatomique, pour déceler des améliorations plus précises. La figure 9.3 détaille la variation du Hausdorff sur chaque structures, tous patients confondus. Au vu du nombre important de structures cérébrales dans la base MICCAI12, nous faisons le choix de se concentrer sur les 15 régions ayant les plus mauvais résultats sur la métrique de Hausdorff, pour suivre leurs évolutions. De même que pour l'étude en fonction des patients, on note une baisse globale de la distance de Hausdorff et une stabilisation au même seuil pour les deux architectures avec DistBranch et Full.

Ces résultats en fonction des patients et des régions tendent à confirmer que l'utilisation de la connaissance de position apporte une amélioration nette de la segmentation produite par le modèle, en augmentant le réalisme par rapport à l'anatomie humaine.

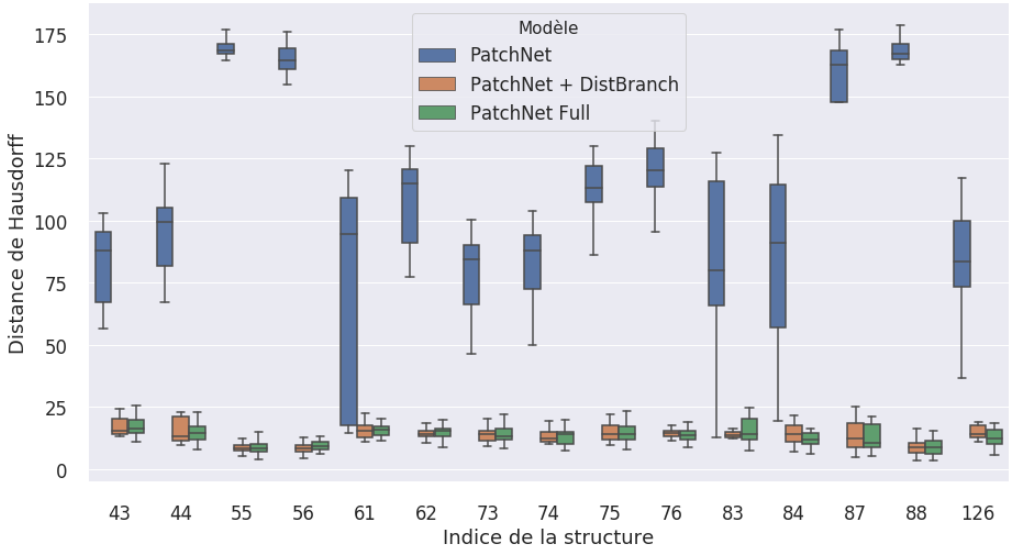


FIGURE 9.3 – Illustration de la dispersion de la distance de Hausdorff, pour tous les patients de la base de test MICCAI12, en fonction des 15 structures les plus mal délimitées. On compare les performances du réseau par patch PatchNet, avec l'ajout de la position du patch (DistBranch) et l'utilisation de toutes les branches (Full).

9.8 Conclusion

Dans ce chapitre, nous avons présenté les résultats des expériences menées pour comprendre l'influence des différentes améliorations proposées à notre architecture de RNC PatchNet. On constate que l'encodage de la position et l'encodage par le biais de la branche DistBranch apporte les plus fortes améliorations des anomalies de segmentation sur la distance de Hausdorff (-75% par rapport à PatchNet). La combinaison finale des approches conduit à un gain significatif pour le Dice (+7%) et la distande de Hausdorff (-76%), avec une augmentation du nombre de paramètres de l'ordre de 127%. Malgré la complexité spatiale et temporelle supérieure au réseau SegNet comparé en entrée, l'approche par patch proposée est plus compétitive sur les aspects de qualité de segmentation et de robustesse aux erreurs anormales.

Chapitre 10

Conclusion

Dans cette partie nous avons présenté nos contributions à la segmentation par patch avec des RNC. Une partie des résultats a été présentée en session orale à la conférence ISBI 2018 [Ganaye et al., 2018b].

Nous avons défini un cadre pour l'intégration de connaissances spatiales dans n'importe quel réseau de classification par patch, qui par défaut ne prend pas naturellement en compte la position dans l'image, pourtant si importante en imagerie médicale et utile pour éliminer les principales incohérences anatomiques. Nous avons montré que l'intégration des distances du patch aux points de repère, dans un RNC 2D multi-résolution peut aider à réduire les incohérences de segmentation et améliorer la qualité globale de la sortie. La représentation de la position sous la forme d'une image de distances, ainsi que l'utilisation de convolution pour l'extraction de descripteurs sur cette dernière est un facteur déterminant dans l'amélioration des résultats. L'ajout de données provenant d'un atlas probabiliste construit sur la base d'apprentissage et d'un patch 3D, ont fait progresser la qualité globale du modèle. Toutefois la distance de Hausdorff est très majoritairement diminuée par l'ajout de l'image de distances.

Une piste d'amélioration probable serait de combiner la méthode avec un recalage déformable pour obtenir des positions plus précises en recalant les patients sur un même atlas. Puis, à l'aide de ces coordonnées spatiales dans l'atlas déformé, nous pourrions construire une image de distances et l'utiliser en entrée du modèle PatchNet Full, on conserverait ainsi la méthode originale en affinant les positions.

Malgré les améliorations proposées, l'approche par patch connaît des limitations, à savoir que l'image est segmentée itérativement en classifiant chaque patchs séparément. Même si ces opérations peuvent être parallélisées pour plus d'efficacité, des architectures plus optimisées pour la segmentation ont été proposées [Long et al., 2015, Ronneberger et al.,

2015, Badrinarayanan et al., 2017]. Ces architectures dites "entièrement convolutives" ouvrent la voie à l'exploitation de contraintes globales à l'échelle de l'image, telles que la cohérence volumique et anatomique, ce qui est pour le moment impossible avec une méthode par patch.

Les approches basées sur des RNC par patch sont généralement entraînées dans le but de minimiser les erreurs de classification entre la segmentation prédite et la vérité terrain. Ces différences sont mesurées par des fonctions de coût telles que l'entropie croisée et le Dice, mais ne se basent pas sur les connaissances topologiques du domaine, pourtant très importantes pour renforcer le réalisme des délimitations. Leur précision est donc limitée à la fois par leur capacité de perception puis par la quantité et la qualité des données d'entraînement disponibles. Idéalement la base de données devrait refléter toute la variabilité inter-sujets et les annotations devraient connaître un large consensus entre experts. Ces deux conditions étant rarement réunies, les premières méthodes basées sur des RNC n'apportaient pas toujours une nette amélioration par rapport aux pipelines de segmentation traditionnels. Aussi diverses tentatives visant à exploiter des propriétés telles que l'invariance anatomique et la connaissance sémantique dans le cadre de RNC furent proposées en détection d'objets et segmentation d'images. L'introduction de contraintes spatiales spécifiques dans un RNC, avec pour objectif de réduire les incohérences de segmentation, a motivé le travail présenté dans la partie suivante de ce manuscrit.

III Prise en compte de contraintes anatomiques d'adjacence dans un RNC pour la segmentation d'images médicales

Chapitre 11

Introduction

Bien que les réseaux de neurones convolutifs aient démontré leurs capacités de reconnaissances visuelles et de reconstruction dans plusieurs domaines de l'imagerie médicale, ils restent à l'heure actuelle inaptes à capturer naturellement des contraintes anatomiques de haut niveau (voisinage inter-structures, inclusion, connexité), des connaissances pourtant évidentes pour un radiologue. Ce manque de robustesse peut mener à des anormalités de segmentation lorsque les données d'entrée présentent des caractéristiques encore inconnues au système. Par exemple dans la figure 11.1, on peut voir le résultat de plusieurs coupes où un réseau de segmentation entraîné sur des données cérébrales a produit des annotations inconsistantes par rapport à l'anatomie cérébrale (présence de groupes de pixels isolés, fuite).

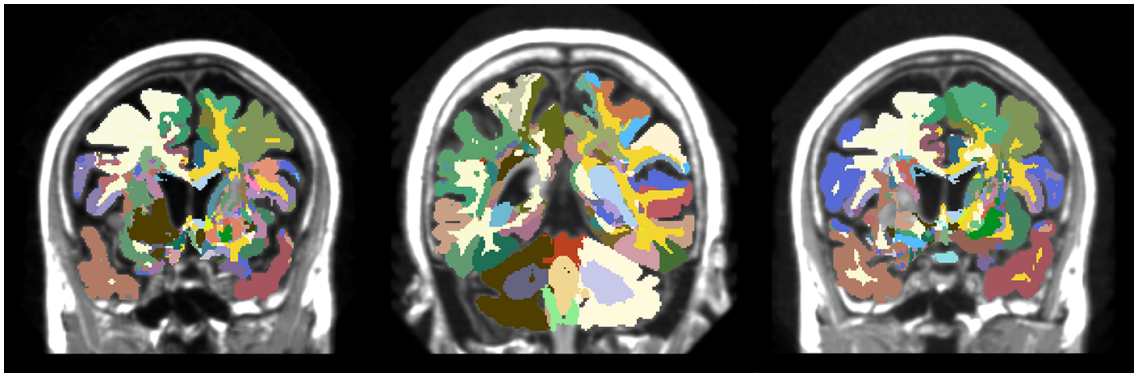


FIGURE 11.1 – Erreurs de segmentation aberrantes produites par un réseau de segmentation convolutif (PatchNet, cf section 6.3) sur la base cérébrale MICCAI 2012.

Pour rendre plus robustes les solutions basées sur un RNC, des contraintes de formes [Oktay et al., 2018, Ravishankar et al., 2017], de volume [Kervadec et al., 2019] ou d'adjacence [BenTaieb and Hamarneh, 2016] ont été proposées ces dernières années. Nous les

divisons en deux familles :

— apprentissage de connaissances de domaine : une représentation de l'espace des structures anatomiques est apprise, puis le réseau de segmentation final est entraîné en utilisant cette représentation latente pour régulariser l'apprentissage du réseau de segmentation.

— modélisation de l'*a priori* : une information topologique ou géométrique est modélisée à travers une fonction de coût différentiable. À l'aide d'un *a priori* déterminé par l'utilisateur ou extrait des données, le RNC est contraint de produire des résultats qui respecte la connaissance modélisée.

Dans la suite de ce chapitre, nous présentons des travaux issus de ces deux familles de RNC avec contraintes pour des applications médicales.

11.1 Apprentissage des connaissances de domaine

La première approche pour limiter les erreurs de segmentation consiste à apprendre une représentation latente des structures anatomiques, souvent par le biais d'un auto-encodeur, puis de l'utiliser pour quantifier la dissimilarité entre la sortie et la vérité terrain dans l'espace appris. Cette mesure est alors exploitée lors de l'apprentissage du réseau de segmentation final, sous la forme d'une pénalisation sur la sortie du modèle.

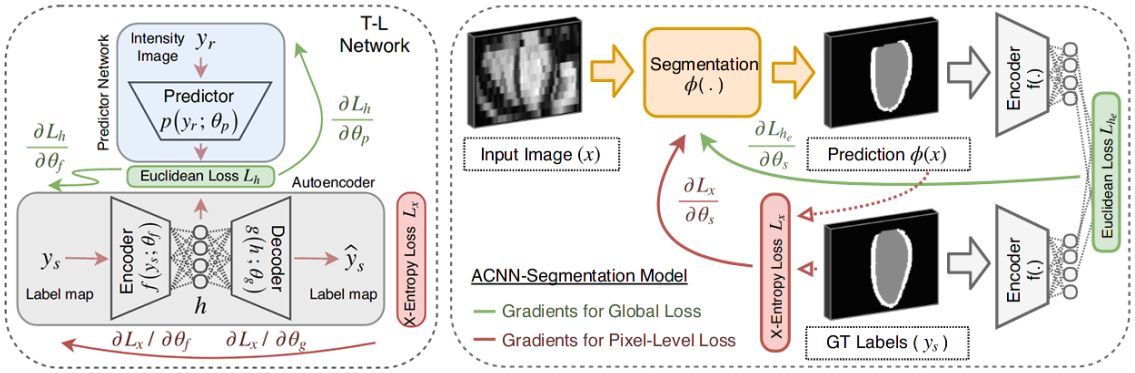


FIGURE 11.2 – Réseau de segmentation par RNC (à droite) avec contrainte de domaine apprise par auto-encodeur (à gauche). Figure issue de [Oktay et al., 2018].

L'approche proposée dans [Oktay et al., 2018] incite le modèle à suivre des propriétés anatomiques (forme, orientation) à travers une représentation latente apprise. Comme illustré dans la figure 11.2, dans un premier temps un auto-encodeur convolutif est entraîné sur les cartes de segmentation pour déterminer une représentation latente des structures. Enfin, un second réseau destiné à la tâche de segmentation, est entraîné en minimisant le terme d'entropie croisée classique, ainsi que la distance euclidienne entre la sortie du réseau et la vérité terrain dans l'espace latent fourni par l'auto-encodeur.

De la même manière que dans [Oktay et al., 2018], [Ravishankar et al., 2017] propose d'intégrer un *a priori* de forme lors de l'apprentissage du RNC, avec un premier auto-encodeur convolutif entraîné à l'aide d'une nouvelle stratégie d'augmentation de donnée. Cette dernière corrompt la qualité des cartes de segmentation originales, dans le but que le réseau apprenne à corriger des erreurs d'annotations commises par le modèle, tout en représentant ces entrées dans un espace latent. L'architecture complète est un réseau en cascade composée d'un premier RNC qui segmente l'image d'entrée, puis de l'auto-encodeur pré-entraîné qui affine la segmentation produite par le premier. La fonction de coût se compose de plusieurs termes minimisant la similitude entre la segmentation inférée et la vérité terrain, ainsi que leur distance dans l'espace latent.

À la différence des deux précédentes approches qui cherchent à maximiser la similarité structurelle, [Baumgartner et al., 2019] suggère de modéliser l'incertitude du résultat en proposant plusieurs sorties respectant la variabilité anatomique. La majorité des modèles de segmentation automatique ne prennent pas en compte l'incertitude ou le fait que plusieurs vérités terrain sont disponibles pour une même image, proposant une unique segmentation pour une image donnée. Seulement des incertitudes liées à la qualité des images, à la difficulté de perception de certaines structures ou encore aux divergences de points de vue entre les annotateurs, peuvent conduire à des ambiguïtés. Dans [Baumgartner et al., 2019], les auteurs modélisent la distribution de probabilité conditionnelle des segmentations en fonction d'une image d'entrée, à partir d'une approche probabiliste hiérarchique dans laquelle plusieurs espaces latents sont en charge de représenter la segmentation à plusieurs résolutions. L'approche se base sur un auto-encodeur variationnel pour contrôler l'espace latent dans lequel la segmentation est représentée, en variant le code, il est alors possible d'obtenir plusieurs segmentations probables de l'image.

11.2 Modélisation directe de l'*a priori*

Si une connaissance anatomique peut être modélisée directement ou approximée à l'aide d'une fonction mathématique différentiable, elle peut être directement appliquée comme contrainte sur la sortie du réseau, lors de l'apprentissage. À l'inverse de la famille de contraintes précédentes qui demande d'apprendre la représentation du domaine des structures, la modélisation directe ne requière pas forcément de données annotées. Par exemple dans [BenTaieb and Hamarneh, 2016], l'*a priori* peut être fourni : à partir des données en procédant à une extraction partielle ou directement par un praticien (radiologue, praticien spécialiste). La contrainte n'étant fonction que de la sortie du réseau et non de la vérité terrain associée à l'image d'entrée, cette approche ouvre la voie aux méthodes d'apprentissage semi-supervisé. On apprend ainsi à vérifier la contrainte y compris sur des images non-annotées, permettant ainsi une meilleure généralisation.

[Kervadec et al., 2019] introduit une fonction de pénalisation différentiable qui applique une contrainte d'inégalité sur la sortie du réseau. Elle est utilisable dans la mesure où des connaissances *a priori* sont disponibles, comme par exemple le volume d'une structure ou l'étiquette d'une région. De la même façon qu'un masque de segmentation, les étiquettes d'images sont utilisables comme forme d'apprentissage faiblement supervisée. Dans ce cas, la contrainte qu'une région d'intérêt soit présente ou absente se traduit aussi comme une inégalité. Ces contraintes d'inégalités sont moins strictes lors de l'apprentissage car elles n'obligent pas à être précis dans leurs définitions.

Dans [BenTaieb and Hamarneh, 2016], les auteurs proposent de modéliser des descripteurs d'image de haut niveau, tels que la continuité des contours et l'interaction entre les régions (inclusion et exclusion) dans le but de les intégrer dans l'apprentissage d'un réseau. Sur le même modèle qu'un CRF, un terme unaire est défini pour régulariser les erreurs d'inclusions entre les structures, ainsi qu'un terme paire à paire contrôlant la continuité des contours. Ces deux contraintes topologiques sont directement intégrées dans la fonction de coût et optimisées conjointement à l'apprentissage.

Nous proposons une nouvelle méthodologie qui réduit le nombre d'anomalies de segmentation en pénalisant les violations des relations d'adjacence entre les régions anatomiques. Dans le chapitre 12 nous présentons l'état de l'art des approches sous contraintes pour la segmentation médicale. Ensuite, la notion de contiguïté anatomique est définie à travers une matrice d'adjacence représentant les connections entre les paires de régions. Enfin le concept est étendu au cas multi-orienté où des dépendances spatiales se combinent aux adjacences.

Dans le chapitre 13, nous proposons une modélisation du calcul de l'adjacence à l'intérieur d'un réseau de neurones entièrement convolutif (section 3.2.2). Cette dernière rend possible l'apprentissage d'un réseau incluant la contrainte anatomique formulée, on nomme cette nouvelle fonction de coût NonAdjLoss. Nous associons à la NonAdjLoss, un algorithme inspiré des méthodes de pénalisation extérieure, pour contrôler le coefficient de la pénalisation, au fil de l'apprentissage. Nous montrons que la pénalité de non-adjacence peut également être utilisée de manière semi-supervisée, en complétant les données d'entraînement annotées par des images supplémentaires sans étiquette, en vue d'améliorer la généralisation sans compromettre la précision.

L'architecture encodeur-décodeur 2D exploitée pour intégrer notre modélisation est détaillée au chapitre 14. Nous adapterons notamment l'architecture du RNC 2D pour pouvoir prendre en compte les contraintes 3D.

Le protocole expérimental qui présente les données et le choix des hyperparamètres est listé dans le chapitre 15. Nous introduisons deux nouvelles métriques pour quantifier le type et le volume des adjacences incorrectes, des mesures qui ne sont pas directement prises en compte par les métriques classiques de segmentation médicale.

Enfin, les résultats des expériences sont commentés dans le chapitre 16, où l'on étudie l'apport de l'architecture 2D et 2.5D, de la contrainte anatomique, de la semi-supervision et de la multi-orientation, sur deux jeux de données de neuroimagerie : MICCAI 2012 [Landman, 2012], IBSR V2 [Worth, 2003] et un jeu de données multi-organes : Anatomy3 [Jimenez-del-Toro et al., 2016].

Chapitre 12

Contrainte anatomique et matrice d'adjacence

Dans le domaine médical, où la forme et la position des structures anatomiques sont un *a priori* fort, la définition de l'adjacence anatomique, sous la forme d'une contrainte en cas d'erreur par rapport à une vérité globale, permettrait de renforcer la robustesse de systèmes d'aide à la décision pour de nombreuses applications médicales. Dans ce chapitre, nous définissons la connaissance anatomique d'adjacence que nous utilisons par la suite, ainsi que la méthodologie pour l'extraire à partir d'une base de données annotée.

12.1 Définition de l'adjacence anatomique

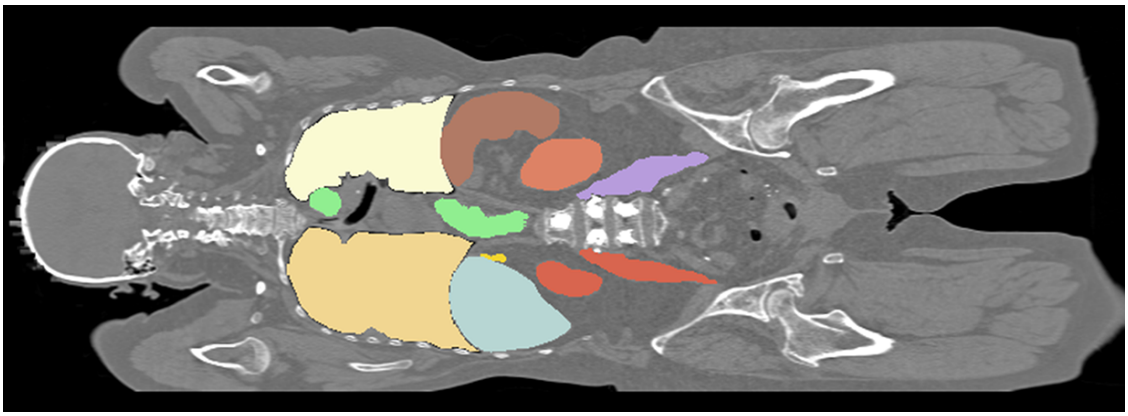


FIGURE 12.1 – Segmentation multi-organes d'un patient en TDM non-contrastée, issue de la base Anatomy3.

De toutes les informations topologiques disponibles dans les annotations des régions anatomiques, l'adjacence joue un rôle majeur. Elle définit la contiguïté entre les frontières de deux régions pour une taille de voisinage donnée. Par exemple, dans la figure 12.1, dont

les annotations sont issues de la base Anatomy3 [Jimenez-del-Toro et al., 2016], on peut observer la délimitation des organes thoraciques et abdominaux dont on déduit visuellement certaines adjacences (foie-poumon, poumon-trachée). Ces connaissances structurelles, si elles sont extraites des données ou fournies par un praticien, fournissent un nouvel indicateur quantifiable de la qualité de segmentation d'une image. Dans nos travaux, nous avons fait l'hypothèse que la position relative des structures anatomiques les unes par rapport aux autres est invariante. La satisfaction d'une contrainte d'adjacence pour la sortie d'une méthode de segmentation, permettrait de renforcer la robustesse de systèmes d'aide à la décision pour de nombreuses applications médicales. Dans la suite, nous présentons une méthode d'extraction des règles d'adjacence à partir des données annotées, puis nous modéliserons l'adjacence sous la forme d'une fonction différentiable, pour l'intégrer dans l'apprentissage d'un RNC.

12.1.1 Cas général matrice d'adjacence

Nous partons de l'hypothèse générale que tous les sujets ont les mêmes adjacences anatomiques et donc les mêmes connectivités inter-régions, même si leurs géométries (forme, volume) peuvent varier. Dans l'image, les relations de contiguïté entre chaque paire de régions i et j peuvent être représentées par une matrice d'adjacence $\mathbf{A}$, où $\mathbf{A}_{ij}$ est le nombre total de voxels sur la frontière entre les structures étiquetées i et j. Formellement, on peut exprimer les coefficients de la matrice $\mathbf{A}$ par :

$$\mathbf{A}_{ij} = \sum_{x} \sum_{v \in V} \delta_{i,s(x)} \delta_{j,s(x-v)}, \tag{12.1}$$

où x est un voxel, $s(x)$ est l'étiquette de x, δ est la fonction delta de Kronecker et V définit un voisinage local dont la taille est paramétrable. $\mathbf{A}$ encode la surface des contours partagés entre des paires de structures dans le volume 3D.

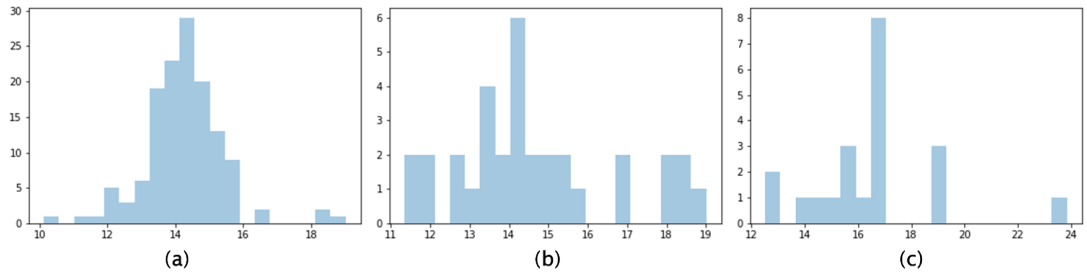


FIGURE 12.2 – Histogrammes des matrices d'adjacence $\mathbf{A}$ (échelle log) extraites pour les trois bases de données respectives MICCAI 2012 (a), IBSR V2 (b), Anatomy3 (c). Les figures illustrent le nombre de structures ayant un effectif similaire d'adjacences anatomiques.

La figure 12.2 présente la répartition du nombre d'adjacence (à l'échelle log) des structures, pour les bases de données MICCAI 12, IBSR et Anatomy3. On constate des disparités dans les effectifs, chaque région ne dispose pas du même nombre d'adjacences, cet effet est

lié aux volumes des structures. En effet, certaines régions étant naturellement plus étendues que d'autres, la surface totale augmente de la même manière le nombre d'adjacence. Appliquer une contrainte anatomique de connectivité se basant sur $\mathbf{A}$ aurait pour effet de corriger les adjacences, mais aussi d'influencer le volume de certaines régions, un effet qui n'est pas toujours souhaitable et complexe à maîtriser.

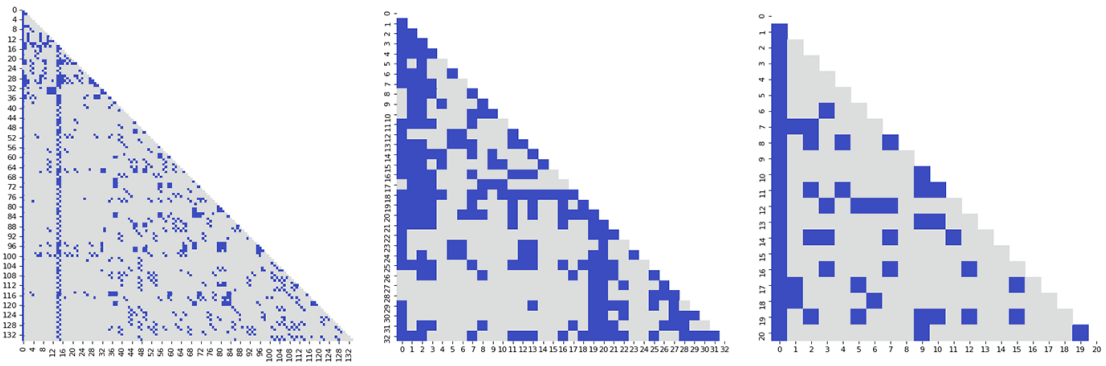


FIGURE 12.3 – Matrices d'adjacence binaires $\tilde{\mathbf{A}}_{ij}$ extraites à partir des jeux de données (de gauche à droite) : MICCAI 2012, IBSR V2, Anatomy3 ; possédant respectivement 135, 33 et 20 structures annotées manuellement. Les points bleus dénotent la présence d'une ou plusieurs adjacences entre les structures, dans un voisinage 3x3x3.

Le volume des structures anatomiques pouvant varier de manière significative entre les sujets en raison de la variabilité inter-patient et des pathologies, nous supposons que notre base d'apprentissage n'est pas assez riche pour capter toutes les variation de volume. Pour cette raison nous avons choisi de binariser $\mathbf{A}$ en $\tilde{\mathbf{A}} = (\mathbf{A} > 0)$. $\tilde{\mathbf{A}}$ est invariante aux déformations homéomorphes d'image et ne retient que le caractère qualitatif de l'adjacence entre les structures, à savoir la présence ou l'absence d'une connexion entre deux régions i et j. Des exemples de matrices $\tilde{\mathbf{A}}$ sont présentés dans la Fig. 12.3. On remarque le caractère creux de $\tilde{\mathbf{A}}$ lorsque le nombre de régions anatomiques délimitées dans la base diminue, par exemple dans la base Anatomy3.

Nous définissons l'ensemble des transitions structurelles impossibles comme suit :

$$F = \{(i, j) \mid \tilde{\mathbf{A}}_{ij} = 0\}, \tag{12.2}$$

pour $\tilde{\mathbf{A}}$ extrait à partir de la base d'entraînement. F définit l'ensemble des adjacences anatomiques que nous voulons éliminer lors de l'apprentissage du modèle. Cet ensemble est représenté dans la figure 12.3 par les cases grises. Il définit le critère de non-adjacence des structures que l'on souhaite faire respecter par tous types de modèle FCN.

12.1.2 Cas particuliers et calculs associés

Dans certaines applications médicales, une orientation spatiale complète les contraintes de connectivité, avec des règles spécifiques à l'orientation du voisinage (haut, bas, droite, etc) :

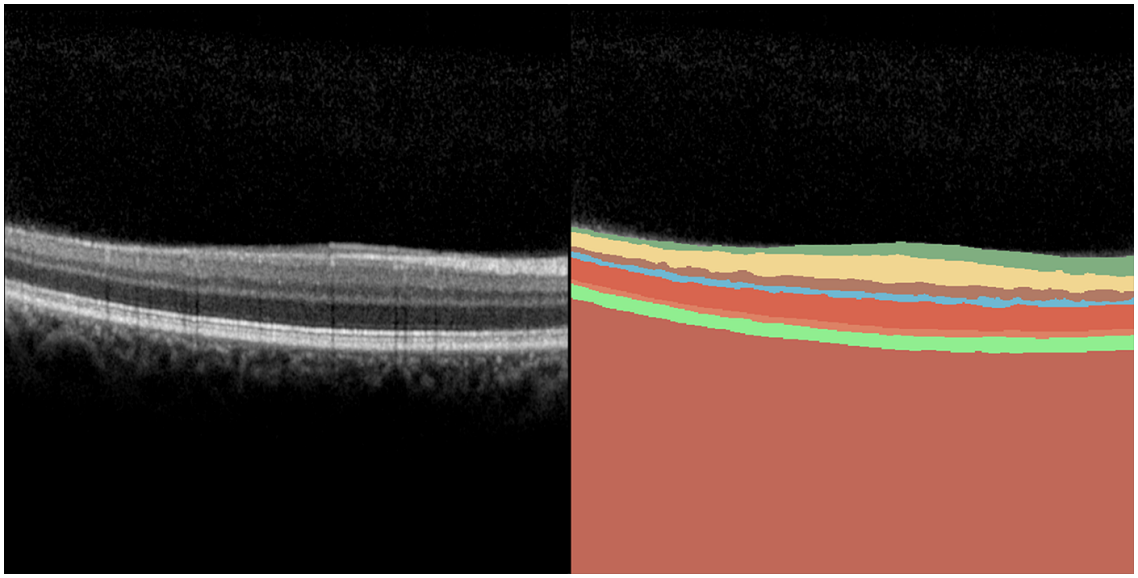


FIGURE 12.4 – Image du fond de la rétine acquise en tomographie par cohérence optique (gauche) et sa segmentation manuelle (droite). Source [Chiu et al., 2015].

— En ophtalmologie, la segmentation du dépôt maculaire de drusen sur des scans rétiniens (acquis en tomographie par cohérence optique : OCT) est utile pour comprendre les risques et la progression de la dégénérescence maculaire liée à l'âge. Dans la figure 12.4, la segmentation d'une image du fond de la rétine acquise par OCT issue de [Chiu et al., 2015], dans laquelle on note la structure en couche des régions qui suggère une contrainte d'adjacence dépendante de l'orientation (adjacence en fonction de l'orientation du voisinage), admettant pour chaque région uniquement deux zones anatomiques possibles, en dessous et au dessus de celle-ci.

— En neuroanatomie, l'hyper-complexité et la multitude des structures corticales et sous-corticales imposent de fortes contraintes d'adjacences et d'orientation, par exemple on sait que le putamen droit est adjacent au pallidum droit et se situe uniquement à sa droite.

— En radiothérapie thoracique, la disposition des organes (cf Fig. 12.1) implique des relations spatiales uniques (foie en dessous du poumon droit, trachée entre poumon gauche et droit, etc).

Jusqu'à maintenant la matrice d'adjacence $\mathbf{A}$ a été déterminée en évaluant, pour chaque pixel, les étiquettes de tous les voisins dans un voisinage 3D symétrique. Cette stratégie n'exploite pas pleinement les contraintes relatives au profil anatomique. On propose donc de renforcer la contrainte anatomique en remplaçant $\mathbf{A}$ par six matrices distinctes, une pour chacune des six orientations disponibles $o \in \mathcal{O} = \{$ avant, arrière, haut, bas, gauche , droite $\}$. Celles-ci sont construites de la même manière que $\mathbf{A}$, mais en utilisant des voisinages orientés qui encodent la contiguïté de chaque direction séparément, comme illustré dans la

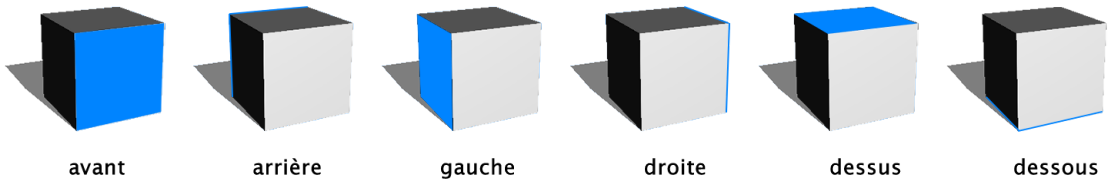


FIGURE 12.5 – Illustration de l'adjacence orientée, pour laquelle au lieu de considérer un *a priori* pour toutes les orientations, on extrait des contraintes propres à l'orientation spatiale du voisinage au point central.

figure 12.5. En reprenant la définition de $\mathbf{A}_{ij}$ dans l'équation 12.1, on trouve $\mathbf{A}^o_{ij}$ la matrice d'adjacence pour le voisinage orienté dans la direction o :

$$\mathbf{A}^o_{ij} = \sum_x \sum_{v \in V^o} \delta_{i,s(x)} \delta_{j,s(x-v)}, o \in \mathcal{O}. \tag{12.3}$$

12.2 Conclusion

Dans ce chapitre, nous avons défini plusieurs formes de matrices d'adjacence, donc certaines peuvent être orientées pour mieux décrire les dépendances spatiales. Ainsi, à travers les matrices d'adjacence, nous capturons un *a priori* anatomique, à savoir la continuité entre structures anatomiques. Cet *a priori* a l'avantage d'être facilement calculable à partir des données segmentées des bases d'images utilisées pour entraîner un RNC.

Dans le chapitre suivant, nous proposons l'intégration de l'adjacence dans un RNC, en considérant la dérivabilité de cette mesure et son intégration dans la fonction de coût.

Chapitre 13

Intégration de la contrainte d'adjacence dans un RNC

Ce chapitre propose l'intégration dans un RNC de la contrainte d'adjacence définie au chapitre précédent. La section 13.1 explicite la dérivation de l'adjacence. Dans la section 13.2, nous intégrons la mesure dans la fonction de coût sous la forme d'une contrainte de non-adjacence que l'on nomme NonAdjLoss. Enfin dans la section 13.3, nous montrons comment nous pouvons étendre notre approche à l'apprentissage semi-supervisé.

13.1 Dérivabilité de la mesure d'adjacence

L'objectif de ce travail est d'incorporer la mesure d'adjacence dans un système de segmentation basé sur un RNC. Nous souhaitons entraîner un RNC à produire des délimitations respectant les contraintes anatomiques encodées dans F (équation 12.2). À cette fin, nous définissons une fonction de contrainte $G(\mathbf{w})$, en fonction des paramètres du réseau $\mathbf{w}$. La valeur de $G(\mathbf{w})$ est nulle lorsque toutes les contraintes sont satisfaites pour toutes les images de la base de données et augmente avec le nombre d'incohérences. L'apprentissage du réseau consiste alors à résoudre le problème d'optimisation suivant :

$$\min_{G(\mathbf{w})=0} \frac{1}{|D_S|} \sum_{(\mathbf{I},\mathbf{S})\in D_S} L(\phi(\mathbf{I},\mathbf{w}),\mathbf{S})) \tag{13.1}$$

$$G(\mathbf{w}) = \sum_{\mathbf{I}\in D_G} \sum_{(i,j)\in F} a_{ij}(\phi(\mathbf{I},\mathbf{w})). \tag{13.2}$$

où $\mathbf{I}$ est une image en niveaux de gris et $\mathbf{S}$ sa carte de segmentation (vérité terrain). D_S et D_G sont les ensembles de données d'apprentissage utilisés respectivement pour la fonction de perte de segmentation et notre contrainte de non-adjacence, NonAdjLoss. L'ensemble D_G inclut généralement D_S, en plus d'images supplémentaires non annotées. ϕ est la

fonction définie par le réseau de neurones. Pour une image $\mathbf{I}$ et étant donné les poids du réseau $\mathbf{w}$, $\phi(\mathbf{I}, \mathbf{w})$ est donc la sortie du réseau : une image multicanal fournissant pour chaque pixel la probabilité d'appartenir à chacune des classes. La fonction a_{ij} qui mesure l'adjacence entre les régions à partir des cartes de probabilités est définie à la suite.

Mesure différentiable d'adjacence La fonction a_{ij} mesure l'adjacence entre les structures i et j à partir des cartes de probabilité produites par le réseau. Sa définition est inspirée de l'équation Eq. 12.1, cependant la fonction $\delta_{.,s(x)}$ doit être modifiée afin d'être exploitable dans le contexte des cartes de probabilité et de la descente de gradient.

En effet, dans l'équation Eq. 12.1, on note l'utilisation de l'opérateur δ pour compter le nombre d'adjacence entre les structures i et j. Il suppose que les étiquettes en chaque pixel x sont connues. Hors dans le cadre d'un réseau de neurones, l'obtention de l'étiquette de x se fait à partir du vecteur de probabilité associé issu de $\phi(\mathbf{I}, \mathbf{w})$, en cherchant l'indice de l'élément à plus forte valeur par le biais de l'opérateur argmax. Cependant la dérivée de ce dernier est nulle presque partout, ce qui rend son évaluation inutile en optimisation par descente de gradient, où l'on recherche la direction qui minimise l'erreur produite par la fonction de coût. Pour ces raisons, nous définissons a_{ij} (équation 13.3) comme une mesure d'adjacence qui tire partie de l'information probabiliste disponible dans les cartes de sortie.

Lorsque deux régions i et j ne doivent pas être adjacentes, i.e $(i, j) \in F$, alors la probabilité d'appartenir à i et à j doit être simultanément nulle pour un pixel et tous ses voisins. Soit $\phi_i(x)$ la carte de probabilité de l'étiquette i dans l'image $\mathbf{I}$, donnée par la sortie du réseau de neurones. Une modélisation mathématique de la contrainte peut s'exprimer par $\phi_i(x)\phi_j(x-v)$ faible pour tout x ainsi que ses voisins $x-v$. Pour appliquer cette règle sur toutes les images, nous définissons a_{ij} comme :

$$a_{ij}(\phi) = \sum_x \sum_{v \in V} \phi_i(x)\phi_j(x-v), \tag{13.3}$$

où ϕ est la carte de probabilité des étiquettes. Si nous définissons $\tilde{\phi} = \phi * \mathbb{1}_V$ comme la convolution de ϕ avec la fonction $\mathbb{1}_V$, indicatrice de l'élément de voisinage V (qui vaut 1 dans V et 0 ailleurs), cette expression peut alors être simplifiée pour un calcul plus efficace :

$$a_{ij}(\phi) = \sum_x \phi_i(x) \sum_{v \in V} \phi_j(x-v) \tag{13.4}$$

$$= \sum_x \phi_i(x)\tilde{\phi}_j(x) \tag{13.5}$$

Si le réseau fournit des sorties parfaitement discriminantes, tel que la valeur maximale des cartes de probabilités soit 1 et les autres 0, alors $\phi_i(x)$ est égal à $\delta_{i,\mathrm{argmax}_k\, p_k(x)}$ et $a_{ij}(\phi)$ devient $\mathbf{A}_{ij}(\phi)$. Avec la contrainte ainsi définie, la fonction de contrainte $G(\mathbf{w})$ (équation 13.2) n'affecte pas uniquement la prédiction de la structure la plus probable, elle pénalise

toutes les adjacences interdites par F en forçant leurs probabilités respectives à zéro. La figure 13.1 schématise le calcul de a_{ij} pour deux structures i et j issues des cartes de probabilités ϕ de sortie du réseau. L'adjacence représente ici la somme sur les pixels (en rose) à l'intersection des deux régions. La contrainte $G(\mathbf{w})$ est la somme des a_{ij} de toutes les adjacences interdites pour toutes les images de la base D_G. Comme la plupart des fonctions de coût qui sont utilisées pour entraîner des réseaux de neurones profonds (avec une fonction de coût basée sur le Dice ou l'entropie croisée (section 3.4) par exemple), la contrainte n'est pas convexe par rapport aux poids du réseau.

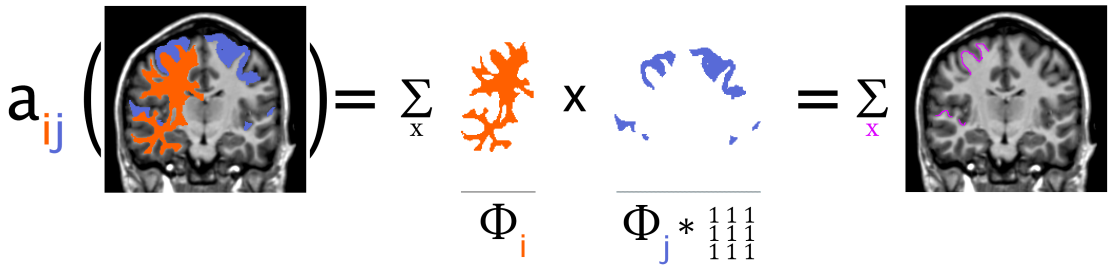


FIGURE 13.1 – Illustration du calcul de l'adjacence a_{ij} à partir de deux cartes de probabilités ϕ_i et ϕ_j.

13.2 Intégration dans la fonction de coût

En pratique, nous résolvons le problème d'optimisation sous contrainte en nous inspirant de méthodes de pénalisation extérieures en optimisation non-linéaire sous contrainte [Nocedal and Wright, 2006]. À savoir, dans un premier temps le réseau est entraîné à partir des fonctions de coûts de segmentation classiques (section 3.4), puis une fois la convergence atteinte, il est ajusté en ajoutant la contrainte $G(\mathbf{w})$ en tant que pénalisation, dont la pondération λ augmente progressivement en fonction des itérations :

$$\min_{\mathbf{w}} \frac{1}{|D_S|} \sum_{(\mathbf{I},y)\in D_S} L(\phi(\mathbf{I},\mathbf{w}),y)) + \lambda\, G(\mathbf{w}), \tag{13.6}$$

avec le premier terme représentant la moyenne de l'erreur de segmentation mesurée par rapport à la base annotée. Le deuxième terme de l'équation 13.1 est la contrainte NonAdjLoss évaluée par rapport aux sorties du réseau. Il s'avère important de pré-entraîner le réseau avec la fonction de perte de segmentation (soft Dice et entropie croisée, section 3.4) avant d'activer la contrainte NonAdjLoss. En effet, cette dernière incitant la sortie à ne produire aucune erreur d'adjacence, nous avons observé à plusieurs reprises des instabilités d'apprentissage où la sortie du réseau est une image composée d'une unique structure, lorsque la pénalisation arrive trop tôt ou est trop fortement pondérée. Dans ce cas, la pénalité d'adjacence se trouve être nulle, du fait de l'absence d'autres structures, toutefois la

fonction de perte de segmentation est non-nulle. Pour simplifier le cadre général de l'application de la NonAdjLoss, un réseau est pré-entraîné pour le problème de segmentation donné, puis affiné en ajoutant la pénalisation NonAdjLoss. Cette méthodologie est plus robuste aux instabilités d'apprentissage, du fait que la qualité du modèle est déjà bonne lors de la pénalisation, où des possibles erreurs anatomiques seront corrigées.

La procédure d'apprentissage est détaillée dans Algo. 3, où $train(\lambda)$ est le résultat du problème d'optimisation Eq. 13.6 pour la pondération λ de NonAdjLoss. L'objectif de l'algorithme proposé est de satisfaire de façon stable et progressive la contrainte NonAdjLoss en augmentant la pondération λ de la NonAdjLoss. Deux types d'instabilités peuvent être générées en raison d'une pondération trop forte : soit une erreur numérique due à l'explosion du gradient, soit une dégradation non-négligeable (par rapport à seuil ϵ fixé par l'utilisateur) du Dice. Dans ces deux cas, la procédure proposée diminue automatiquement λ si nécessaire.

Algorithme 3 : Algorithme d'apprentissage sous contrainte

Initialisation :
$L_0 \, , G_0 \; = train(0)$
$\lambda = \lambda_{ratio} \times \frac{L_0}{G_0}$
for $i = 0$ *to* $i = n_{epoque}$ **do**
  $L_i = train(\lambda)$ **if** $i \mod n_{update}$ **then**
    if $L_0 - L_i < \epsilon$ **then**
      $\lambda = \lambda * \lambda_{increase}$
    else
      $\lambda_{increase} = \lambda_{increase} * \lambda_{reduction\_factor}$
      $\lambda = \lambda * \lambda_{reduction}$
    end
  end
end

Dans l'algorithme 3, L_i et G_i sont respectivement le Dice moyen et la NonAdjLoss moyenne, calculés sur la base d'entraînement à la fin de l'itération i. Initialement, λ est défini de telle sorte à ce que la valeur de la pénalisation sur l'adjacence soit $lambda_{ratio}$ pour cent de la fonction de perte de segmentation - dans la pratique, 30% ($\lambda_{ratio} = 0.3$). En fixant $lambda_{ratio}$ à une valeur élevée (0.8 par exemple) cela pousserait le processus d'optimisation à corriger principalement les erreurs d'adjacence, affectant d'office la qualité de la segmentation (mesurée avec le Dice). À l'opposé, un λ_{ratio} trop bas ralentirait la convergence vers l'optimal de la contrainte d'adjacence. Durant l'entraînement, si les mesures de Dice sur la base de validation sont stables ou en amélioration, λ est augmenté de $\lambda_{increase}$ chaque n_{update} itération. Inversement, si le Dice diminue plus de ϵ en dessous de celui de l'itération initiale (non contrainte), λ est ramené à une valeur inférieure et la valeur du pas d'augmentation $\lambda_{increase}$ est également réduite. $\lambda_{reduction\_factor}$ est la constante utilisée pour réduire $\lambda_{increase}$ en cas de diminution du Dice, un faible $\lambda_{increase}$ ralentirait la

convergence tandis qu'une valeur trop élevée créerait des instabilités d'apprentissage.

Multi-orientation L'apprentissage de contraintes d'adjacence spatialement orientées (section 12.1.2) est une généralisation de la NonAdjLoss au cas où l'on dispose d'une matrice d'adjacence binarisée $\tilde{\mathbf{A}}_o$ spécifique à chacune des orientations o, elles même définies à travers un voisinage V_o pour toutes les orientations $o \in \mathcal{O}$. Au cours de l'apprentissage, chacune des 6 contraintes $G_o(\mathbf{w})$ est appliquée :

$$\min_{\forall o \in \mathcal{O},\ G_o(\mathbf{w})=0} \frac{1}{|D_S|} \sum_{(\mathbf{I},y)\in D_S} L(\phi(\mathbf{I},\mathbf{w}),y)). \tag{13.7}$$

La procédure d'optimisation numérique pour résoudre ce problème contraint est la même que la section 13.2 à la seule différence que la pénalité est la somme des six fonctions G_o.

Sélection de modèle multi-objectifs Afin de sélectionner le meilleur ensemble de paramètres $\mathbf{w}$ du réseau, il faut généralement rechercher l'itération à laquelle la métrique de validation atteint son meilleur niveau ou lorsque qu'elle ne progresse plus. Cependant, dans ce travail, nous nous intéressons à la fois aux métriques de segmentation et d'adjacence en proposant une règle de sélection multi-objectifs. Pour déterminer l'ensemble optimal des paramètres $\mathbf{w}$, nous identifions les itérations avec les cinq meilleurs scores de Dice sur la base de validation et choisissons le modèle présentant la plus faible erreur moyenne pour le critère de non-adjacence. Cette stratégie joue un rôle important dans la recherche de modèles optimaux vis-à-vis de la qualité de segmentation et de la contrainte de non-adjacence, cela contribue aussi à réduire le sur-apprentissage (section 3.6).

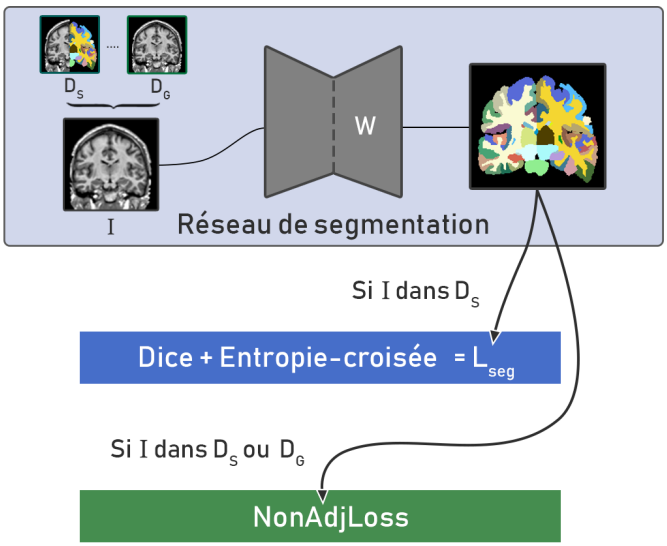


FIGURE 13.2 – Vue globale de la méthodologie d'apprentissage, où les paramètres du réseau $\mathbf{w}$ sont optimisés à partir de L_{seg} (pour les images annotées) et NonAdjLoss (pour tout type d'image).

13.3 Extension à l'apprentissage semi-supervisé

Une fois la matrice d'adjacence **A** extraite des cartes de segmentation de D_S, elle est considérée comme la vérité terrain de l'adjacence pour une image donnée I. On note également que la fonctions a_{ij} et par extension la pénalité G, ne dépendent pas de la segmentation manuelle de I. En utilisant le réseau avec a_{ij} comme mesure d'adjacence, nous pouvons estimer la connectivité de toutes les images, qu'elles soient annotées ou pas. Cette spécificité permet d'inclure des images non annotées dans la base de données d'apprentissage D_G (cf figure 13.2), utilisée lors de l'évaluation de l'adjacence : ce qui permet de nous placer dans le cadre de l'apprentissage semi-supervisé. À travers cette dernière, le réseau est simultanément entraîné à segmenter les régions en fonction d'annotations complètes lorsqu'elles sont disponibles, et à appliquer la NonAdjLoss à toutes les images, qu'elles soient annotées ou pas. Comme nous le montrerons dans les expériences (cf chapitre 16), cette modalité d'apprentissage donne la possibilité d'améliorer la fiabilité anatomique des étiquettes de sortie en incluant des jeux de données multi-centriques non annotés lors de l'apprentissage.

13.4 Conclusion

Nous avons définis une mesure d'adjacence qui peut être intégrée lors de l'apprentissage d'un réseau de neurones convolutifs, afin de pénaliser les segmentations qui ne respectent pas l'*a priori* de non-adjacence anatomique. Cette pénalisation est aussi généralisable dans le cas où l'application médicale bénéficie d'une structuration spatiale fine des adjacences, en calculant la NonAdjLoss en fonction des orientations. Une procédure d'optimisation qui contrôle l'importance de cette approche dans la fonction de coût a été proposée, avec pour but de stabiliser la phase d'apprentissage en prenant en compte progressivement la contrainte d'adjacence. Enfin une stratégie d'apprentissage semi-supervisé de la contrainte a été proposée, pour ainsi tirer partie de larges bases de données non-annotées et améliorer la généralisation de notre méthode. L'implémentation de la méthode ne requière pas l'utilisation d'une architecture particulière, si ce n'est le fait que la sortie du réseau doit être la segmentation d'une image complète ou partielle. Dans le chapitre suivant, nous présentons les architectures utilisées lors des expérimentations conduites au cours de cette thèse et analysées au chapitre 16.

Chapitre 14

Architectures des RNCs 2D et 3D

Dans ce chapitre nous présentons l'architecture du FCN que nous avons utilisée en complément de notre contrainte d'adjacence. Dans la section 14.1 nous détaillons l'architecture 2D, puis dans la section 14.2 l'architecture 3D.

14.1 Architecture encodeur-décodeur 2D : EDNet

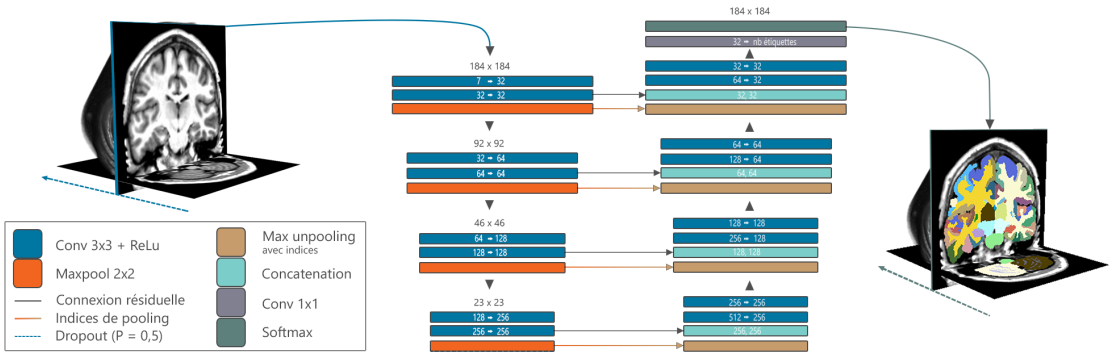


FIGURE 14.1 – Schéma de notre réseau de segmentation 2D EDNet. 7 coupes successives d'une image 3D sont données comme entrée du réseau de neurones, qui produit la carte de segmentation de la coupe centrale. Une architecture entièrement basée sur des convolutions de type encodeur-décodeur est utilisée pour obtenir une segmentation coupe par coupe du volume. Le réseau EDNet contient environs 3 millions de paramètres à optimiser.

Notre architecture de RNC est un encodeur-décodeur inspiré de [Roy et al., 2017], elle même basée sur [Badrinarayanan et al., 2017]. Notre réseau (Fig. 14.1) que l'on nomme EDNet, prend en entrée 7 coupes 2D consécutives extraites du volume et les utilise pour segmenter la coupe centrale exclusivement. Les coupes supplémentaires apportent des informations contextuelles sur la section centrale, améliorant la robustesse globale de la méthode. Le réseau est composé de quatre couches de sous-échantillonnage 2x (pour le chemin

d'encodage) dans lesquelles les indices des pixels d'intensités maximales sont stockés. Dans le chemin de décodage, elles sont suivies de quatre étapes de sur-échantillonnage basées sur la couche max-unpool, qui ré-utilise les indices stockés précédemment pour améliorer l'interpolation. Chaque couche de décodage a également des connexions directes depuis les couches d'encodage de même niveaux de résolutions.

14.2 Extension à la 3D

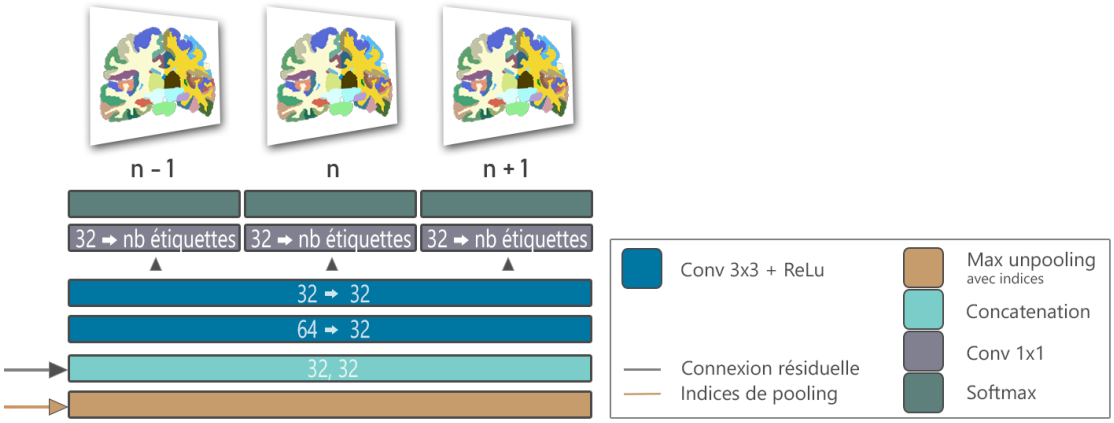


FIGURE 14.2 – Configuration du dernier bloc de convolution pour transformer EDNet en architecture 2.5D. La convolution finale est convertie en 3 convolutions parallèles, générant 3 cartes de segmentation distinctes. La sortie du réseau est modifiée pour segmenter les trois coupes successives $n-1$, n et $n+1$.

Pour tirer parti de la nature 3D des images médicales, nous souhaitons élargir les contraintes d'adjacences dans la dimension z (profondeur) pour appliquer des contraintes spécifiques à chaque orientations (section 12.1.2) ou globale (section 12.1.1). Cependant l'utilisation d'une architecture 2D tel que EDNet pose problème à cet égard car elle produit exclusivement des sorties 2D. L'utilisation de convolutions 3D résoudrait ce problème au détriment d'un réseau beaucoup plus lourd en terme de complexité de calcul, avec de fortes contraintes sur la mémoire GPU requise. Pour éviter cela, nous avons modifié l'architecture EDNet 2D de la Fig. 14.1 pour segmenter les trois coupes centrales (au lieu d'une seule coupe) issues des sept coupes d'entrée. Précisément, nous avons remplacé la convolution finale 1x1 par trois convolutions parallèles 1x1 (voir Fig. 14.2). Avec ce réseau dit "2.5D", les cartes de probabilités des 3 coupes centrales successives sont estimées et la fonction de coût globale se trouve aussi modifiée en calculant l'erreur de segmentation sur les trois sorties (au lieu d'une seule). Chaque coupe de sortie est optimisée en fonction de sa propre vérité terrain, de sorte que la fonction de perte pour la segmentation devienne la somme non pondéré de la fonction de coût originale pour chacune des coupes. On nomme EDNet 2.5D, l'architecture utilisée pour segmenter trois coupes successives à la fois. Cela permet :

— De produire une segmentation 3D, sans utiliser de convolutions avec un noyau 3D, d'où l'appellation 2.5D.

— D'utiliser un voisinage 3D pour calculer les contraintes d'adjacence isotropes ou multi-orientées.

— D'utiliser une stratégie de fusion au cours de l'inférence.

Méthode de fusion Lors de la segmentation de la coupe c_i, avec i l'indice de la coupe centrale en entrée du réseau et $s_{i,j}$ la carte de probabilités en sortie de la coupe $j \in \{i-1, i, i+1\}$, nous obtenons en fonction de l'architecture 2D ou 2.5D des configurations différentes. Dans le cas 2.5D qui nous intéresse, l'évaluation du réseau sur c_i permet l'annotation automatique de trois coupes (c_{i-1}, c_i, c_{i+1}). En faisant varier i on traite donc plusieurs fois une même coupe 2D avec des contextes d'entrée qui varient. Le fait que la fenêtre glissante produise des segmentations chevauchées donne la possibilité d'affiner s_j à travers une stratégie de vote ou de fusion. Soit $s_{i-1,i}$, $s_{i,i}$ et $s_{i+1,i}$ les trois cartes de probabilités de la coupe i obtenues en appliquant le réseau sur les coupes c_{i-1}, c_i et c_{i+1}, on prend comme valeur pour s_i la moyenne des segmentations, c'est à dire $s_i = \frac{s_{i-1,i}+s_{i,i}+s_{i+1,i}}{3}$. Avec cette stratégie, jusqu'à trois versions segmentées d'une même coupe sont déterminées, en les fusionnant, on corrige éventuellement certaines anomalies ou incertitudes de délimitation. Dans notre cas, nous choisissons de fusionner les segmentations en sommant directement les probabilités des cartes, avant application de la fonction argmax sur ces dernières. Ce post-traitement ne nécessite pas de données supplémentaires, mais oblige une augmentation du temps de calcul qui est non négligeable (facteur temporel x10, dû en parti au stockage des cartes de probabilité sur le CPU). Dans le cas de la base MICCAI12, on passe d'un temps de segmentation par image de 2.3 secondes sans post-traitement à 19.8 secondes avec fusion.

14.3 Conclusion

Dans ce chapitre nous avons proposées les architectures des RNCs encodeur-décodeur 2D et 2.5D nommées EDNet, qui intègrent notre contrainte d'adjacence et qui seront utilisées dans le protocole expérimental décrit au chapitre suivant.

Chapitre 15

Protocole Expérimental

Dans ce chapitre nous décrivons le protocole expérimental que nous avons conduit avec nos architectures EDNet 2D et 2.5D. Dans la section 15.1 les détails d'implémentation et choix des hyperparamètres des algorithmes sont présentés. La pénalisation NonAdjLoss ainsi que la stratégie d'apprentissage semi-supervisé ont été testées sur deux jeux de données de neuroimagerie [Landman, 2012, Worth, 2003], et un jeu de données corps entier [Jimenez-del-Toro et al., 2016] (section 15.2). Pour mesurer la capacité de nos méthodes à réduire les erreurs d'adjacence dans les images segmentées par un réseau de neurones, nous proposons de nouvelles métriques topologiques qui évaluent le type et le volume des adjacences erronées (Section 15.3).

15.1 Détails d'implémentation

Nous détaillons les hyperparamètres que nous avons sélectionnés au cours des expériences pour le problème de segmentation cérébrale de la base MICCAI 2012.

Apprentissage du réseau pour la segmentation L'optimisation des paramètres du réseau a été réalisée à l'aide d'une descente de gradient stochastique (SGD) avec un momentum de $0,9$. La taille du batch a été fixée à 8. La vitesse d'apprentissage a été initialisée à $0,01$ et mise à jour à l'aide de la stratégie de polynomial rate de [Chen et al., 2016] $(1 - \frac{iter}{max_{iter}})^{power}$, où $iter$ est l'indice de l'itération, max_{iter} le nombre d'itérations et $power$ un hyperparamètre qui détermine la vitesse de décroissement, ici fixé à 0.9. Le choix des paramètres pour les autres bases de données est détaillé dans le tableau 15.1, puis dans le tableau 15.3 pour l'optimisation de la NonAdjLoss.

La fonction de coût pour la segmentation est une somme pondérée de l'entropie croisée et de la fonction de perte Dice, toutes deux définies dans la section 3.4. Lors de l'optimisation

	nombre d'itérations	taille batch	vitesse d'apprentissage
MICCAI12	300	8	0.03
IBSRv2	300	8	0.01
Anatomy3	200	16	0.005

TABLE 15.1 – Descriptif des paramètres d'apprentissage pour les trois bases de données, au cours de la phase d'optimisation dédiée à la segmentation.

de la fonction de perte d'entropie croisée, nous avons constaté que le déséquilibre des structures posait problème en raison du grand nombre de classes et des variations de volume considérables, pouvant parfois mener à une divergence en début d'apprentissage. D'après le travail de [Roy et al., 2017], une pondération fréquentielle médiane $\omega(\mathbf{x})$ a été utilisée avec succès pour l'entropie croisée :

$$\mathbf{poids} = \frac{\text{median}(\mathbf{f})}{\mathbf{f}}. \tag{15.1}$$

Avec $\mathbf{f} = [f_0, ..., f_c]$ le vecteur des fréquences d'apparitions des régions. Nous avons également pondéré la fonction de perte Dice à l'aide d'un paramètre $\beta = 5$ pour équilibrer le poids des termes avec l'entropie croisée.

L'équation de la fonction de coût pour la segmentation est la suivante :

$$\mathcal{L} = \frac{1}{N}\frac{1}{|\mathbf{x}|}\sum_{(\mathbf{x},\mathbf{y})\in(\mathbf{X},\mathbf{Y})}\sum_{i\in|\mathbf{x}|}\underbrace{-\mathbf{poids}_{y_i} \times \log(\phi(\mathbf{x}_i)_{y_i})}_{\text{entropie croisée}} - \beta\underbrace{\frac{2\sum_c^C \phi(\mathbf{x}_i)_c \times onehot(\mathbf{y}_i)_c}{\sum_c^C \phi(\mathbf{x}_i)_c + onehot(\mathbf{y}_i)_c}}_{\text{soft dice}}, \tag{15.2}$$

avec $(\mathbf{x}, \mathbf{y})$ le couple image et carte d'annotations issues d'une base de données annotée manuellement $(\mathbf{X}, \mathbf{Y})$, $|\mathbf{x}|$ le nombre de pixels contenus dans l'image, $\phi(\mathbf{x})_c$ la carte de probabilité en sortie du réseau pour la classe c, $onehot(\mathbf{y})$ la carte d'annotations cible encodée sous forme one-hot (cf section 3.4.2) et N le nombre d'exemples dans la base.

Fine-tuning du réseau avec la NonAdjLoss La pénalisation NonAdjLoss a été progressivement appliquée sur un modèle pré-entraîné, ajustant la pondération λ à l'aide de l'algorithme proposé Algo. 3 (section 13.2). Les paramètres de ce dernier lors des expériences sont détaillés dans le tableau 15.2.

Cette approche nous permet d'appliquer progressivement la contrainte NonAdjLoss tout en maintenant une mesure de Dice acceptable. Les hyperparamètres d'apprentissage sont détaillés dans le tableau 15.3.

Lors du calcul de a_{ij}, la mesure d'adjacence en sortie du réseau, nous choisissons pour tous les modèles NonAdjLoss(n) un voisinage V de 1 pixel, toutefois il est possible d'élargir la taille de ce dernier afin de capter les relations de connectivité à des échelles variables.

paramètre	valeur	rôle
λ_{ratio}	$0.3 \in [0, 1]$	importance de la pénalisation par rapport à la segmentation dans la fonction de coût lors de la première initialisation de λ.
$\lambda_{increase}$	$1.3 \in [1, +\infty]$	facteur d'augmentation de λ si l'apprentissage est stable.
$\lambda_{reduction\_factor}$	$0.98 \in [0, 1]$	facteur de diminution de $\lambda_{increase}$ en cas d'instabilité.
$\lambda_{reduction}$	$0.9 \in [0, 1]$	facteur de diminution de λ en cas d'instabilité.
n_{update}	$5 \in [1, +\infty]$	nombre d'itération entre deux mises à jour de λ.
ϵ	$0.02 \in [0, +\infty]$	seuil de tolérance pour la dégradation du Dice.

TABLE 15.2 – Détail et rôle des paramètres de l'algorithme contrôlant l'évolution de la pondération λ de la NonAdjLoss

	nombre d'itérations	taille batch	vitesse d'apprentissage
MICCAI12	170	8	0.001
IBSRv2	170	8	0.001
Anatomy3	100	16	0.0005

TABLE 15.3 – Descriptif des paramètres d'apprentissage pour les trois bases de données, au cours de la phase de fine-tuning dédiée à la pénalisation NonAdjLoss.

15.1.0.1 Comparaison avec un CRF

Une comparaison avec l'approche de post-traitement par champ aléatoire conditionnel (CRF) [Krähenbühl and Koltun, 2011] a été effectuée, dans la mesure où la pénalisation proposée peut être considérée comparable dans l'objectif à atteindre, à savoir corriger des anormalités de segmentation en apprenant des relations entre les pixels. Seulement, plusieurs différences séparent ces deux approches, la plus importante étant l'intégration de l'*a priori* d'adjacence directement lors de l'apprentissage du réseau, éliminant tout calcul additionnel lors de l'inférence, à l'inverse d'un CRFs. L'inférence du CRF dense [Krähenbühl and Koltun, 2011] a été réalisée en 15 itérations, avec un terme unaire basé sur les cartes de probabilité produites par le réseau et deux termes par paire (dépendant de la position spatiale et de l'image). Les termes par paire tirent parti de $\tilde{A}$, la matrice d'adjacence binarisée, en tant que matrice d'affinité des structures. L'augmentation du nombre d'itérations à 50 a entraîné une baisse des performances (environ +2mm pour la distance moyenne de Hausdorff sur la base MICCAI12) et un temps de calcul d'environ une heure.

15.2 Bases de données

Pour mesurer l'impact de nos contributions, nous avons utilisé quatre bases d'imagerie médicale, dont trois avec des annotations manuelles par des experts.

Neuro-imagerie La contrainte NonAdjLoss et ses variantes ont été évaluées sur des segmentations de régions cérébrales à partir d'images IRM en séquence pondéré T1 des

	subjects	labels	train	validation	test
MICCAI12	35	135	10	5	20
IBSRv2	18	33	10	3	5
OASIS	406	0	284	122	0
Anatomy3	20 + 55	20	10	10	25

TABLE 15.4 – Détails des trois bases de données d'IRM cérébrale (MICCAI12, IBSRv2, OASIS) et de la base corps entier (Anatomy3) en TDM. Les colonnes indiquent le nombre d'images et la séparation des données pour chacune des étapes du protocole d'expérimentation. La base de données OASIS est issue d'une étude multi-centrique et ne comporte pas d'annotations. La base de données Anatomy3 comporte des annotations d'experts ainsi que des annotations obtenues par fusion des résultats des participants du challenge.

jeux de données MICCAI 2012 multi-atlas challenge [Landman, 2012] et IBSRv2 [Worth, 2003] (voir les exemples dans la figure 15.1 à gauche). La base OASIS [Marcus et al., 2010b] a été utilisée comme donnée d'apprentissage non annotée pour les expériences semi-supervisées (section 13.3), à l'exclusion des sujets figurant également dans MICCAI 2012. Dans IBSRv2, 6 étiquettes sur 39 ont été supprimées des données originales (ex : lésion, vaisseau sanguin, inconnu). Chaque jeu de données d'imagerie cérébrale a été divisé en sous-ensembles d'apprentissage / validation / test, comme présenté dans le tableau 15.4. Nous avons suivi le protocole expérimental officiel de séparation des données pour le base MICCAI12, mais aucun protocole officiel n'a été fourni pour le jeu de données IBSRv2.

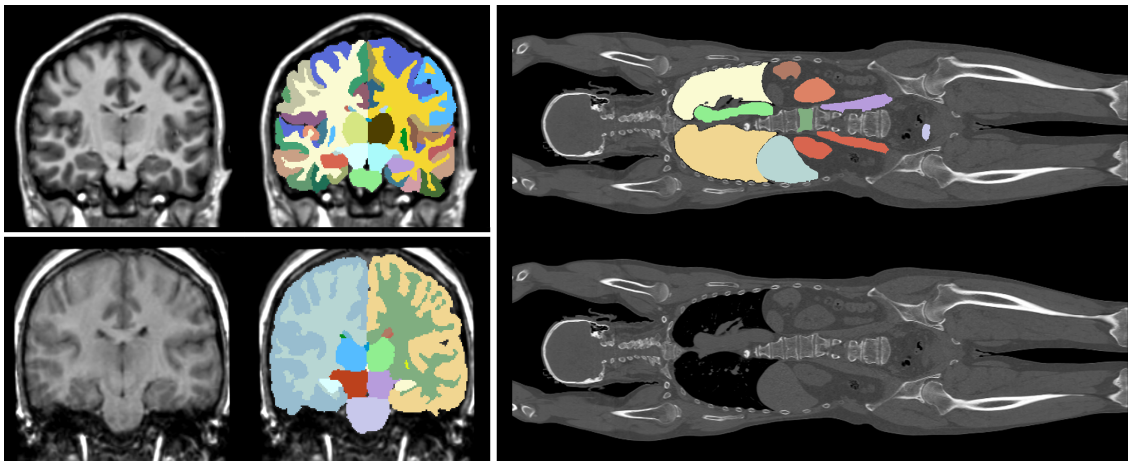


FIGURE 15.1 – Exemples d'images annotées issues des trois bases de données : MICCAI 2012 (en haut à gauche), IBSR V2 (en bas à droite) et Anatomy3 (à droite).

Afin de simplifier le problème de segmentation initiale et de tirer partie de l'invariance anatomique, toutes les images sont réorientées dans la même direction. Pour cela, toutes les images ont été alignées avec un recalage affine vers un atlas de référence dans l'espace MNI avec FSL FLIRT, puis rééchantillonnées à un espacement voxelique de 1 mm. Une correction des inhomogénéités de champ a été appliquée avec N4ITK [Tustison et al., 2010]. La moyenne et l'écart type ont été estimés pour chaque jeu de données sur la base d'apprentissage et les images correspondantes ont été centrées et réduites. La standardi-

sation a pour effet d'accélérer la convergence et d'améliorer les performances. L'extraction de la boite crânienne n'a pas été nécessaire comme étape de pré-traitement, en effet nous avons constaté que notre RNC n'est pas significativement sensible à la présence du crâne. Pendant l'entraînement, les images issues de la base annotée (D_S) ont été artificiellement augmentées à l'aide de déformations élastiques [Simard et al., 2003].

Imagerie corps entier Anatomy3 est une base de données multi-organes composée d'images 3D en TDM et d'IRM avec et sans agent de contraste, dans laquelle 20 régions anatomiques ont été annotées par des experts qualifiés (voir exemple dans la figure 15.1 à droite). Elle a été créée pour le défi Visceral [Jimenez-del-Toro et al., 2016], qui n'est plus actif à l'heure actuelle. Nous n'avons malheureusement pas pu accéder au jeu de tests utilisé pour le challenge. Cependant, un jeu "Silver Corpus" a été publié, avec des annotations obtenues par la fusion des segmentations obtenues à partir des modèles des participants. Notre répartition train / validation / test est la suivante :

— 10 images pour l'entraînement, 10 images pour la validation, avec toutes les données provenant de la base d'entraînement officielle.

— 25 images pour le test, 30 images pour la semi-supervision (sans les annotations), avec toutes les données provenant du "Silver Corpus".

À la différence de l'imagerie IRM, la tomodensitométrie indique des intensités sur l'échelle de Hounsfield, qui ont une correspondance physique. Par exemple l'air, l'eau et la boite crânienne ont respectivement une mesure de -1000 HU, 0 HU, 1900 HU. Pour cette raison, afin de ne prendre en compte que les organes humains dans l'image, toutes les images ont été seuillées entre $[-1000; 2000]$ Hounsfield Unit et l'intensité normalisée par standardisation. À des fins de calcul (limitation de la mémoire GPU), un sous-échantillonnage à la résolution 256 x 256 a été appliqué sur l'axe xy (plan d'acquisition), par rapport à la résolution d'origine en 512x512. Pendant l'entraînement, les images issues de la base annotée (D_S) ont été augmentées à l'aide de déformations élastiques [Simard et al., 2003].

15.3 Métriques d'adjacence

Pour toutes les expériences, les métriques de distance (section 3.5.2) de Hausdorff et de distance surfacique moyenne ont été quantifiées. Cependant, ces informations ne sont pas des mesures directes des défauts topologiques tels que les erreurs d'adjacence, que nous souhaitons corriger. Pour quantifier cette incohérence anatomique, nous introduisons deux nouvelles métriques qui sont évaluées sur les segmentations produites par le réseau EDNet :

$$CA^{unique}(A^I) = \frac{|O^I \cap H|}{|H|} \tag{15.3}$$

$$CA^{volume}(A^I) = \frac{\sum_{(i,j)\in(O^I\cap H)} A^I_{ij}}{vol_{contour}}, \tag{15.4}$$

avec A^I est la matrice d'adjacence issue d'une carte de segmentation en sortie du réseau pour l'image I, $\tilde{A}$ la matrice binarisée issue de la vérité terrain, $O^I = \{(i,j) \mid A^I_{ij} > 0\}$ l'ensemble des paires (i,j) ayant au moins une adjacence, $H = \{(i,j) \mid \tilde{A}_{ij} = 0\}$ l'ensemble des paires de structures sans adjacence et $vol_{non_contour}$ le nombre total de voxels qui ne sont pas des contours dans la segmentation inférée.

CA^{unique} est le pourcentage de toutes les adjacences anormales uniques qui apparaissent quelque part dans l'image. CA^{volume} est le pourcentage de voxels dans l'image qui ont une adjacence interdite, normalisé par le volume des régions. CA^{unique} mesure la proportion d'adjacences de région incorrectes, tandis que CA^{volume} donne le rapport volumétrique des erreurs d'adjacence.

Les figures 15.2 et 15.3 témoignent de la complémentarité des indicateurs proposés. La mesure CA^{volume} quantifie le volume des adjacences incorrectes (cf figure 15.3), mais n'informe pas si les erreurs concernent uniquement un nombre restreint de régions (figure 15.2 à droite) ou un nombre important (figure 15.2 à gauche). En combinant CA^{volume} et CA^{unique}, on caractérise la surface de l'erreur d'adjacence et la diversité des régions concernées.

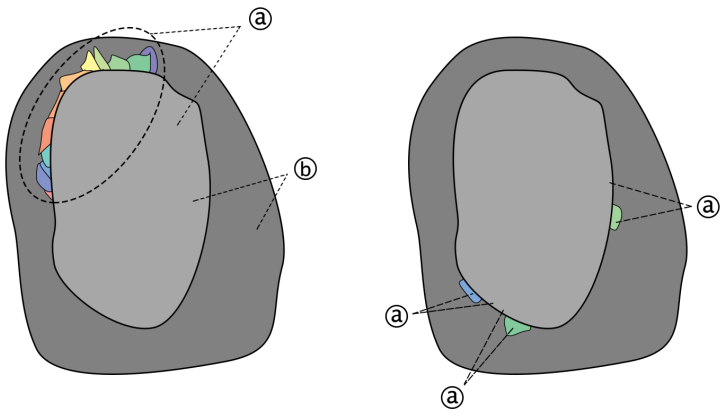


FIGURE 15.2 – Illustration de deux cartes de segmentation avec des adjacences incorrectes (a) et admises (b). La figure de gauche montre une dizaine de structures (en couleurs) ne satisfaisant pas la contrainte, en opposition à la figure de droite qui présente moins d'erreurs et de types de relations d'adjacence incorrectes. La mesure CA^{unique} indiquera une valeur plus forte pour l'exemple à gauche en raison du nombre plus élevé de paires de régions incorrectes.

15.4 Conclusion

Dans ce chapitre nous avons détaillé les données, le protocole expérimental et les métriques avec lesquels nous avons obtenu les résultats présentés au chapitre suivant.

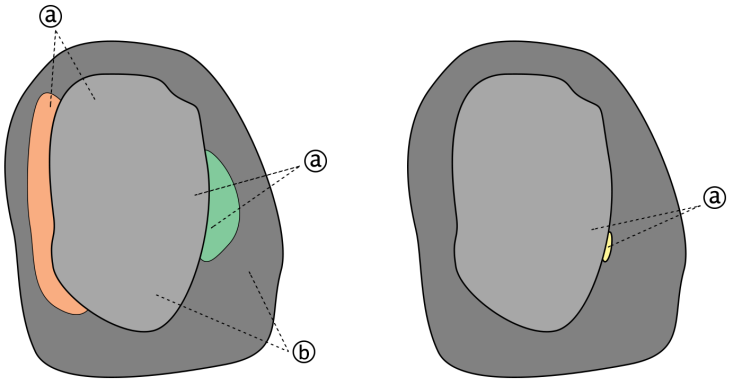


FIGURE 15.3 – Illustration de deux cartes de segmentation avec des adjacences incorrectes (a) et admises (b). La figure de gauche montre des adjacences incorrectes entre des structures dont la surface s'étend le long de la région centrale. Le volume de pixels ayant des contraintes de connectivités anormales est plus élevé que dans la figure droite, une indication qui sera quantifiable à travers le calcul de CA^{volume}.

Chapitre 16

Résultats

Ce chapitre présente et analyse les principaux résultats obtenus en ayant intégré une contrainte d'adjacence (section 13.1) dans une architecture d'un RNC de type encodeur-décodeur.

Nous évaluons l'intérêt de la contrainte NonAdjLoss 2D sur un RNC pré-entraîné (section 16.1), puis nous intégrons l'aspect multi-échelle en faisant varier le voisinage pour plusieurs mesures de non-adjacence (section 16.2). Nous évaluons ensuite dans quelle mesure l'utilisation de données non-annotées peut faire progresser la généralisation de la contrainte (section 16.3). Enfin, nous étendons l'architecture EDNet 2D en 2.5D (section 16.4) en modifiant également la NonAdjLoss pour les cas mono-orientation et multi-orientations (section 16.5). Pour terminer, on compare l'utilisation du réseau par patch avec ou sans *a priori* spatial (partie II), à un RNC 2D utilisant la contrainte NonAdjLoss (section 16.7).

16.1 Application de la non-adjacence 2D

Les résultats de l'application de la contrainte NonAdjLoss sont présentés dans le tableau 16.1, où EDNet correspond au modèle 2D de référence sans apprentissage contraint (fonctions de perte Dice et entropie croisée). NonAdjLoss(n) correspond au modèle 2D pénalisé par la contrainte NonAdjLoss avec n images non annotées issues de la base de données OASIS (ou du Silver Corpus pour Anatomy3), utilisées pour la semi-supervision.

On note des améliorations significatives de la distance de Hausdorff moyenne lorsque la contrainte NonAdjLoss est prise en compte (avec un niveau de confiance de 95% assuré par un t-test apparié), à la fois sur MICCAI12 (-40.89%), IBSRv2 (-12.19%) et Anatomy3 (-34.15%). De même, les mesures d'adjacences CA^{unique} et CA^{volume} évaluées sur 30 images de OASIS, indiquent que les connectivités anormales sont réduites de manière importante, avec des baisses de 94.7% du nombre d'adjacences uniques anormales pour

la base MICCAI12, 93% pour IBSRv2 et 88% pour Anatomy3. À titre de comparaison avec les méthodes prenants en compte un *a priori* spatial, EDNet a été post-traitée par l'approche CRF dense proposée dans [Krähenbühl and Koltun, 2011] avec un terme unaire basé sur la vraisemblance logarithmique négative issue du réseau. Dans tab. 16.1, nous notons que l'ajout de l'inférence CRF entraîne une légère amélioration des métriques de distance et de connectivité, au prix de 13 minutes de temps de calcul, contre moins d'une seconde pour les modèles EDNet et NonAdjLoss.

MICCAI12	Dice	HD (mm)	MSD (mm)	CA^{unique}	CA^{volume}
EDNet	0.740 ± 0.11	20.93 ± 9.50	1.18 ± 0.40	5.1e-2 ± 6.8e-2	1.8e-2 ± 4.9e-2
EDNet + CRF	0.739 ± 0.11*	18.86 ± 8.03*	1.17 ± 0.40	4.4e-2 ± 6.8e-2*	1.5e-2 ± 4.5e-2*
NonAdjLoss(0)	0.734 ± 0.10*	12.37 ± 4.62*	1.10 ± 0.34	2.7e-3 ± 6.6e-3*	2.6e-4 ± 9.4e-4*
IBSRv2					
EDNet	0.833 ± 0.11	15.99 ± 15.27	0.78 ± 0.37	1.0e-1 ± 8.8e-2	1.5e-3 ± 3.0e-3
NonAdjLoss(0)	0.835 ± 0.10*	14.04 ± 15.45	0.76 ± 0.34*	7.0e-3 ± 2.1e-2*	3.1e-5 ± 1.5e-4*
Anatomy3					
EDNet	0.682 ± 0.26	88.76 ± 52.30	3.88 ± 2.31	9.2e-2 ± 3.9e-2	3.9e-4 ± 6.4e-4
NonAdjLoss(0)	0.679 ± 0.26	58.44 ± 39.46*	3.38 ± 2.01	1.1e-2 ± 1.3e-2*	3.5e-5 ± 8.3e-5*

TABLE 16.1 – Effet de la prise en compte de la contrainte NonAdjLoss sur 3 bases de données et comparaison avec un post-traitement par CRF. Métriques de similarité, distances et de connectivité mesurées pour chaque modèle. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD, MSD, CA^{unique} et CA^{volume} sont moyennées sur l'ensemble de test. Le caractère * indique que la moyenne de la métrique est significativement différente de celle de EDNet, avec un seuil de confiance de 95%. Nous reportons de la façon suivante : score moyen ± écart type.

Les expériences sur le problème de segmentation multi-organes Anatomy3 présentées dans Tab. 16.1 montrent les mêmes tendances que pour les bases MICCAI12 et IBSRv2. À savoir, une nette diminution de l'ordre de 30 mm de la distance moyenne de Hausdorff, ainsi qu'une réduction de la connectivité anormale (-88%). En inspectant les mesures de Dice, les rapports indiquent une légère dégradation de 0.01, ce qui peut être attribué à la sélection des hyperparamètres et à notre critère de sélection multi-objectifs.

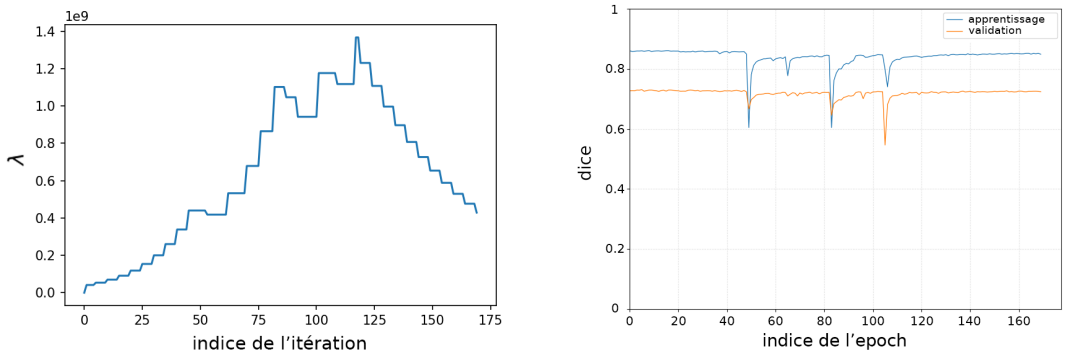


FIGURE 16.1 – À gauche, courbe d'évolution de λ lors de l'apprentissage de NonAdjLoss(0). λ est contrôlé par l'algorithme 3, sa valeur est augmentée au cours des itérations et réduite en cas d'instabilité. À droite, mesure du Dice moyen lors de l'apprentissage sur les ensembles d'entraînement et de test.

16.1.1 Contrôle de la pondération

Pour l'application de la pénalisation NonAdjLoss, le contrôle de la pondération λ est essentiel pour garantir au mieux la robustesse de la contrainte. L'algorithme 3 proposé maîtrise l'augmentation de λ en fonction de la stabilité de l'apprentissage. L'évolution de λ lors de l'optimisation de NonAdjLoss(0) pour la base MICCAI12 est présenté dans la figure 16.1 (à gauche), où l'on observe une évolution quasi-croissante, jusqu'à la diminution programmée autour de 125 itérations pour la recherche d'un possible optimum Dice/NonAdjLoss. En cas d'instabilités telle qu'une baisse du Dice mesurée sur la base de validation (visible à droite dans la figure 16.1), l'augmentation de λ est interrompue temporairement, on note plusieurs évènements de ce type à partir de l'itération 50.

16.2 Non-adjacence 2D multi-échelle

Dans le but de prendre en compte des adjacences entre structures anatomiques dans un voisinage de plus de 1 pixel, nous introduisons une méthodologie multi-échelle qui est une somme non pondérée de plusieurs pénalisation NonAdjLoss en 2D, avec des tailles de voisinage V spécifiques à chacune d'entre elles (section 12.1). Dans cette expérience, nous considérons deux modèles : Multi-scale 1, 3, 5 et Multi-scale 1, 5, 7 qui sont l'application de la NonAdjLoss pour toutes les tailles de voisinage spécifiées. Les résultats sont présentés dans le tableau 16.2.

MICCAI12	Dice	HD (mm)	MSD (mm)	CA^{unique}	CA^{volume}
EDNet	0.740 ± 0.11	20.93 ± 9.50	1.18 ± 0.40	5.1e-2 ± 6.8e-2	1.8e-2 ± 4.9e-2
NonAdjLoss(0)	0.734 ± 0.10*	12.37 ± 4.62*	1.10 ± 0.34	2.7e-3 ± 6.6e-3*	2.6e-4 ± 9.4e-4*
Multi-scale{1, 3, 5}	0.737 ± 0.10*	12.46 ± 5.23*	1.09 ± 0.36	4.4e-3 ± 1.0e-2*	5.4e-4 ± 1.5e-3*
Multi-scale{1, 5, 7}	0.734 ± 0.10*	12.01 ± 4.69*	1.09 ± 0.34	2.6e-3 ± 5.4e-3*	3.0e-4 ± 8.4e-4*

TABLE 16.2 – Comparaison de la prise en compte de la contrainte d'adjacence à plusieurs échelles. Métriques de similarité, distances et de connectivité mesurées pour chaque modèle sur la base MICCAI12. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD, MSD, CA^{unique} et CA^{volume} sont moyennées sur l'ensemble de test. Le caractère * indique que la moyenne de la métrique est significativement différente de celle de EDNet, avec un seuil de confiance de 95%. Nous reportons de la façon suivante : score moyen ± écart type.

Bien que la prise en compte de distances variables entre les organes ait du sens sur le plan anatomique, l'approche multi-échelle ne démontre pas sur la base MICCAI12 d'amélioration significative des performances par rapport à l'utilisation de la contrainte NonAdjLoss avec un voisinage fixe, aussi bien pour la similarité que pour les distances d'erreur (cf tableau 16.2). Ces résultats ne remettent pas en cause l'efficacité de la contrainte proposée pour corriger les connectivités anormales, mais plutôt l'intérêt d'une approche multi-échelle pour la segmentation de structures cérébrales. D'un point de vue expérimental, on note toutefois une meilleure stabilité du processus d'optimisation, qui est probablement renforcé par les informations renvoyées par chacune des échelles. Il est aussi important de noter que

les tailles de voisinage sont ici sélectionnées arbitrairement, sans prendre en compte les spécificités de la base MICCAI12. Pour être plus rigoureux nous aurions pu analyser à partir de la base d'entraînement la distance moyenne qui sépare les régions et utiliser les échelles les plus représentatives.

16.3 Semi-supervision

La semi-supervision, à savoir l'utilisation d'images annotées et non-annotées lors de l'apprentissage du modèle, fait une utilisation intelligente de données n'ayant pas été délimitées, ce qui se révèle utile pour améliorer la généralisation de la contrainte d'adjacence structurelle. Cela est particulièrement vrai dans le cas où l'accès aux données annotées est très limité, mais que des données d'imagerie brute sont à disposition. Dans la Fig. 16.2 (à gauche) est illustré pour chaque structure, le total des adjacences incorrectes de tous les sujets (sur une échelle logarithmique), pour les modèles EDNet, NonAdjLoss(0) et NonAdjLoss(50) de MICCAI12.

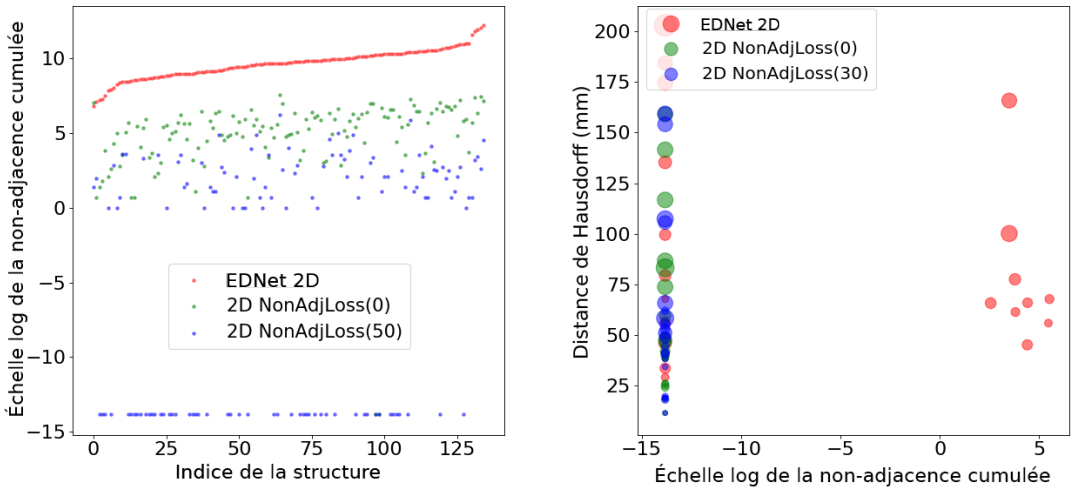


FIGURE 16.2 – À gauche, illustration des erreurs d'adjacence pour chaque régions anatomiques sur 30 image de la base test OASIS, pour les modèles entraînés sur MICCAI12. Le total des adjacences d'erreur est passé à l'échelle log et les régions sans erreur sont égales à -14. Les régions sont triées en fonction de la non-adjacence du modèle EDNet. À droite, influence de la connectivité sur la distance de Hausdorff pour la base Anatomy3, le diamètre des points est proportionnel à leur écart-type.

L'approche semi-supervisée (bleu) est nettement plus fiable que la référence ou NonAdjLoss, fournissant une segmentation sans anomalie (selon nos critères de non-adjacence) pour un grand nombre de régions anatomiques de cette base de données. L'effet sur la base Anatomy3 (Fig. 16.2 à droite) est plus restreint de par le nombre réduit d'adjacences anormales dans le modèle de référence.

On observe des améliorations claires (tableau 16.3) pour les métriques de connectivité (CA^{unique}, CA^{volume}) avec une baisse de 82.5% du nombre de connections anormales

MICCAI12	Dice	HD (mm)	MSD (mm)	CA^{unique}	CA^{volume}
EDNet	0.740 ± 0.11	20.93 ± 9.50	1.18 ± 0.40	5.1e-2 ± 6.8e-2	1.8e-2 ± 4.9e-2
NonAdjLoss(0)	0.734 ± 0.10*	12.37 ± 4.62*	1.10 ± 0.34	2.7e-3 ± 6.6e-3*	2.6e-4 ± 9.4e-4*
NonAdjLoss(20)	0.739 ± 0.10*	11.19 ± 4.40*	1.06 ± 0.34	5.8e-4 ± 1.4-e3*	2.8e-5 ± 9.6e-5*
NonAdjLoss(50)	0.741 ± 0.10	10.97 ± 4.37*	1.04 ± 0.33	3.9e-4 ± 9.9e-4*	1.4e-5 ± 4.8e-5*
NonAdjLoss(100)	0.743 ± 0.10	11.31 ± 4.69*	1.04 ± 0.33	4.7e-4 ± 1.5e-3*	1.9e-5 ± 6.8e-5*
IBSRv2					
EDNet	0.833 ± 0.11	15.99 ± 15.27	0.78 ± 0.37	1.0e-1 ± 8.8e-2	1.5e-3 ± 3.0e-3
NonAdjLoss(0)	0.835 ± 0.10*	14.04 ± 15.45	0.76 ± 0.34*	7.0e-3 ± 2.1e-2*	3.1e-5 ± 1.5e-4*
NonAdjLoss(20)	0.834 ± 0.10	12.75 ± 13.26	0.77 ± 0.34*	1.2e-3 ± 2.2e-3*	3.4e-7 ± 8.7e-7*
NonAdjLoss(50)	0.832 ± 0.10	11.92 ± 12.65*	0.77 ± 0.37	1.6e-3 ± 4.6e-3*	1.8e-6 ± 8.1e-6*
Anatomy3					
EDNet	0.682 ± 0.26	88.76 ± 52.30	3.88 ± 2.31	9.2e-2 ± 3.9e-2	3.9e-4 ± 6.4e-4
NonAdjLoss(0)	0.679 ± 0.26	58.44 ± 39.46*	3.38 ± 2.01	1.1e-2 ± 1.3e-2*	3.5e-5 ± 8.3e-5*

TABLE 16.3 – Effet de l'augmentation du nombre d'images non-annotées lors de l'apprentissage du réseau EDNet avec la contrainte NonAdjLoss. Métriques de similarité, distances et de connectivité mesurées pour chaque modèle. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD, MSD, CA^{unique} et CA^{volume} sont moyennées sur l'ensemble de test. Le caractère * indique que la moyenne de la métrique est significativement différente de celle de EDNet, avec un seuil de confiance de 95%. Nous reportons de la façon suivante : score moyen ± écart type.

uniques pour MICCAI12, par rapport à la contrainte NonAdjLoss sans semi-supervision. Cependant, pour les métriques de surface (HD , MSD), aucune amélioration nette n'est observée, toutefois les moyennes respectives restent globalement stables. On voit qu'avec les modèles NonAdjLoss(n), les mesures Dice sont généralement similaires à EDNet, quelques fois meilleur ou avec des détériorations. Ce changement de tendance dans les indicateurs de performance peut s'expliquer par la nature des contraintes imposées par NonAdjLoss. La suppression des incohérences corrige généralement des petits groupes de pixels éloignés de leur véritable emplacement. Comme ces erreurs sont faibles en nombre, l'impact sur le Dice est limité (surtout après moyennage des cartes de probabilités), mais si la distance qui les sépare de leur véritable emplacement est grande, l'impact sur la distance de Hausdorff sera d'autant plus important. Les deux exemples de segmentation dans la figure 16.4 illustre ce phénomène où l'on observe des non-adjacences de faible volume (au centre) corrigées par l'approche semi-supervisée (à droite).

Les matrices d'adjacences obtenues en fusionnant toutes les adjacences observées en segmentant les ensembles de test de MICCAI12, IBSRv2 et Anatomy3 sont présentées dans la Fig. 16.3. Pour chacune des bases de données utilisées (une base par ligne), on observe des tendances similaires dans la réduction des adjacences uniques, à savoir une première diminution significative entre EDNet (colonne de gauche) et NonAdjLoss(0) (colonne du milieu), puis une baisse plus faible avec NonAdjLoss(30) (colonne de droite), qui reste non négligeable étant donné qu'elle ne nécessite pas d'annotation supplémentaire. La taille du voisinage V (cf Eq. 12.1) de l'adjacence est fixé à $3 \times 3 \times 3$. Aucun pré-traitement n'a été requis sur la matrice d'adjacence, après plusieurs tentatives de seuillage du nombre d'adjacences dans le but d'éliminer les transitions erronées introduites par des erreurs

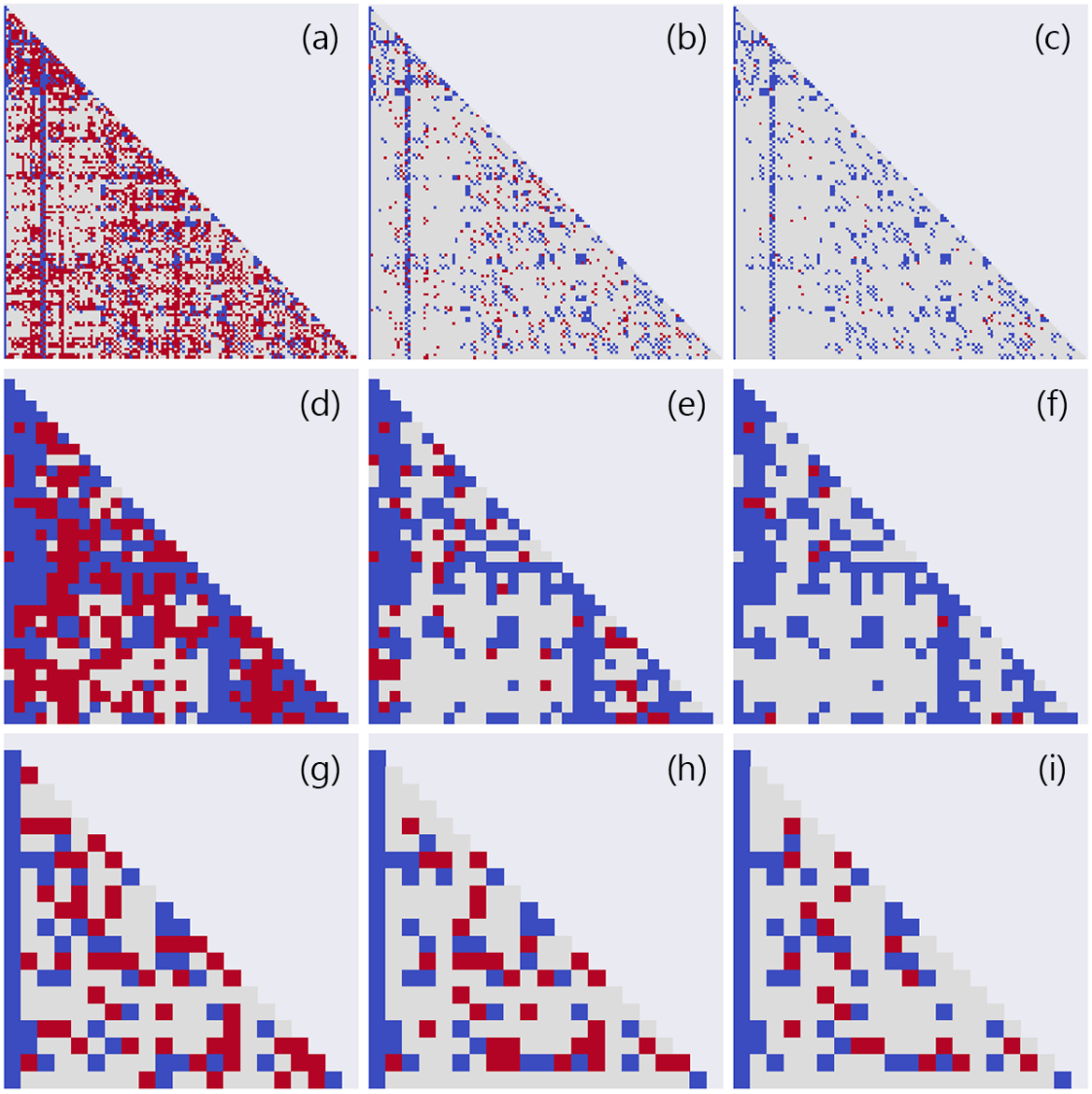


FIGURE 16.3 – Effet de la prise en compte de la contrainte NonAdjLoss sur les matrices d'adjacences. Matrices d'adjacences binaires extraites sur les bases de test MICCAI12 (a, b, c), IBSRv2 (d, e, f), Anatomy3 (g, h, i) pour les modèles EDNet (a, d, g), NonAdjLoss(0) (b, e, h), NonAdjLoss(30) (c, f, i). Les points rouges indiquent la présence d'au moins une adjacence anormale pour les paires de régions correspondantes.

d'annotation, aucune amélioration n'a été constatée. Dans les deux bases de neuroimagerie, les modèles de référence produisent un grand nombre de transitions interdites entre classes (points rouges, cf figure 16.3), tandis que les mêmes RNC entraînés avec la NonAdjLoss génèrent beaucoup moins d'erreurs. Dans la base de données Anatomy3, on note les mêmes conclusions que précédemment. Ces tendances sont confirmées par un t-test apparié avec un niveau de confiance de 95%, où la moyenne de chaque modèle est comparée à la moyenne de EDNet afin de tester la présence d'une différence significative qui confirmerait l'utilité du modèle. Les erreurs de segmentation qui sont éloignées spatialement de leur vraies structures sont corrigées comme attendu, tandis que les adjacences autorisées (points bleus) sont préservées.

Au regard de ces résultats, on peut conclure que l'apprentissage semi-supervisé a de l'intérêt pour améliorer la généralisation de la contrainte en utilisant des données non-annotées, jamais observées auparavant, tout en maintenant les performances sur les métriques de segmentation et de distance surfacique.

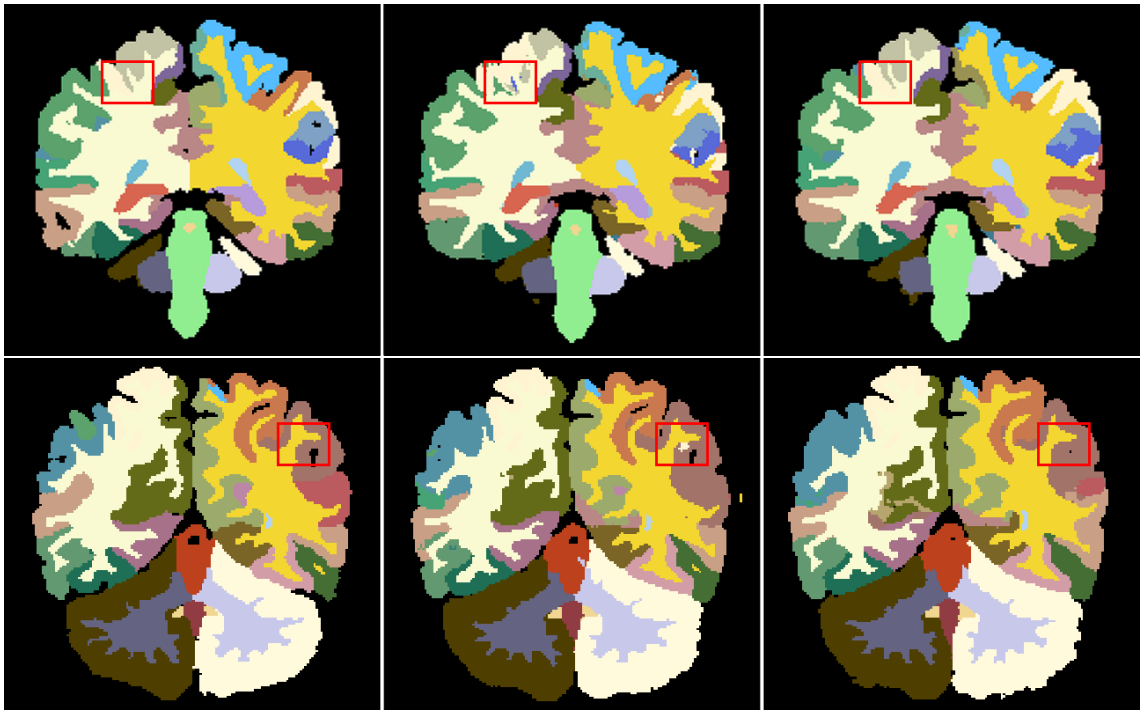


FIGURE 16.4 – Carte de segmentation de deux patients issus de la base de test de MICCAI12, de gauche à droite : vérité terrain, EDNet, NonAdjLoss(50). Les boites rouges mettent en valeur les incohérences anatomiques corrigées.

MICCAI12	Dice	HD (mm)	MSD (mm)
EDNet 2.5D	0.733 ± 0.11	19.77 ± 9.52	1.239 ± 0.397
EDNet 2.5D + fusion	0.738 ± 0.11	16.06 ± 7.11	1.196 ± 0.390
NonAdjLoss(0) + fusion	0.736 ± 0.10	12.10 ± 4.75	1.148 ± 0.382
NonAdjLoss(50) + fusion	0.744 ± 0.10	**10.19 ± 3.73**	1.055 ± 0.336
IBSRv2			
EDNet 2.5D	0.832 ± 0.11	14.48 ± 16.00	0.792 ± 0.341
EDNet 2.5D + fusion	0.834 ± 0.11	12.60 ± 14.60	0.781 ± 0.346
NonAdjLoss(0) + fusion	0.837 ± 0.10	**9.71 ± 10.38**	0.755 ± 0.331
NonAdjLoss(50) + fusion	0.835 ± 0.10	10.94 ± 13.96	0.765 ± 0.321

TABLE 16.4 – Comparaison de l'utilisation de l'architecture 2.5D et de la fusion des cartes de probabilités. Métriques de similarité et distances mesurées pour chaque modèle. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD et MSD sont moyennées sur l'ensemble de test. Nous reportons de la façon suivante : score moyen ± écart type.

16.4 Adjacence 3D isotrope avec architecture 2.5D

Afin d'étudier si d'avantage d'informations de connectivité 3D issues des images du cerveau peuvent être exploitées avec un coût raisonnable, nous avons entraînes le RNC EDNet avec l'architecture 2.5D proposée dans la section 14.2. Durant l'inférence, nous avons appliqué une stratégie de post-traitement basée sur la fusion, qui consiste à sommer les cartes qui se chevauchent tout en faisant glisser la fenêtre de segmentation sur tout le volume. Le tableau 16.4 montre que ce post-traitement réduit les erreurs aberrantes, ce qui entraîne une diminution de la distance de Hausdorff. La combinaison du modèle 2.5D, de la pénalisation NonAdjLoss avec la semi-supervision et du post-traitement donne les meilleurs résultats lors de nos expériences pour les bases MICCAI12 et IBSRv2, sur les critères de similarité et de distance. La figure 16.5 montre pour les bases de données MICCAI12 et IBSRv2, la distance moyenne de Hausdorff en fonction de chaque structure anatomique, comparant le modèle EDNet à NonAdjLoss(0) et 2.5D NonAdjLoss(50) avec fusion. On observe que ce dernier présente des erreurs plus faibles que les autres pour presque toutes les régions anatomiques dans la base MICCA12. Cela est dû majoritairement à l'utilisation de la semi-supervision et du post-processing par fusion. Le même effet de moindre amplitude est relevé pour les modèles entraînés sur la base IBSRv2. Ces observations sont validées par les matrices d'adjacences binaires (figure 16.6), où l'on constate une décroissance du nombre de connections anormales uniques.

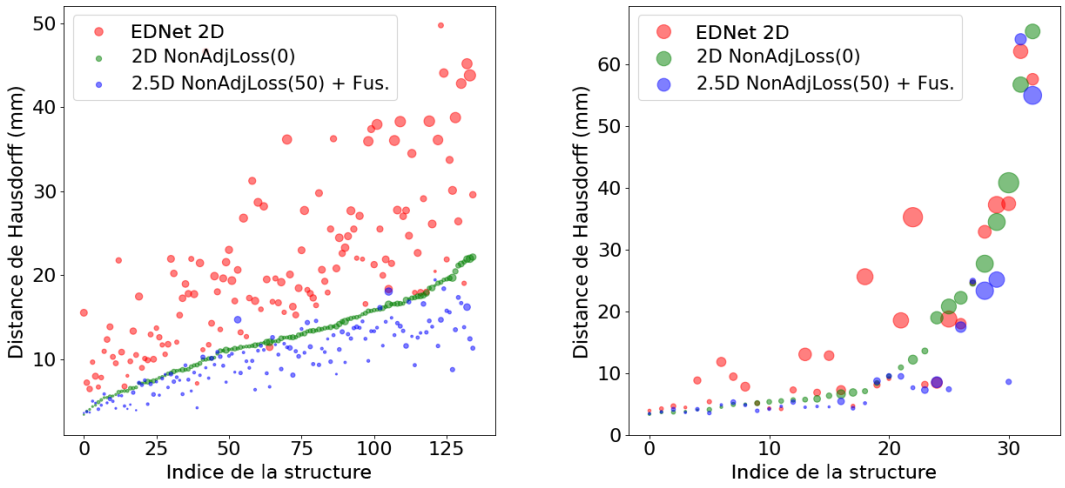


FIGURE 16.5 – Illustrations de l'influence des modèles proposés sur la distance de Hausdorff pour MICCAI12 et IBSRv2. Le total des adjacences d'erreur est passé à l'échelle log et les régions sans erreurs sont égales à -14. Pour les distances de Hausdorff, le diamètre des points est proportionnel à leur écart-type. Les régions sont triées en fonction des variables en ordonnée.

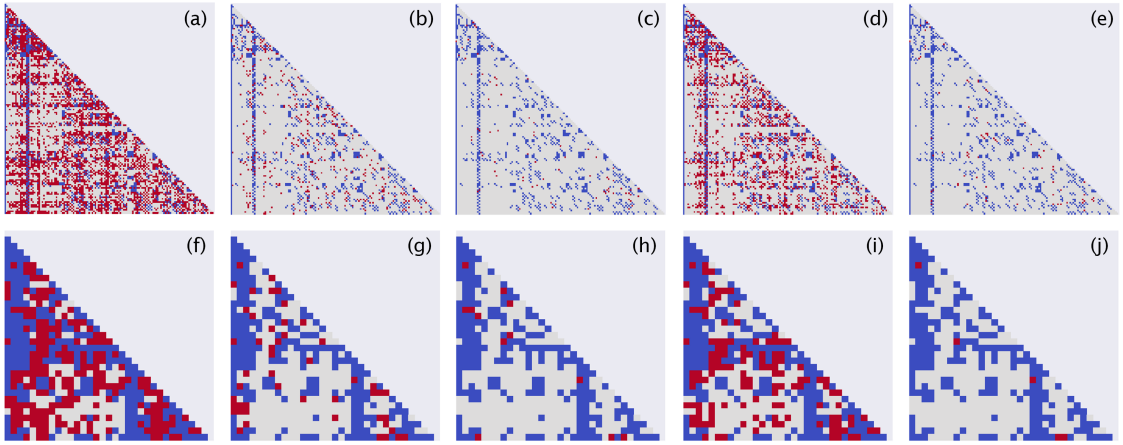


FIGURE 16.6 – Influence de la contrainte sur les architectures 2D et 2.5D avec et sans semi-supervision. Matrice d'adjacence binaires produites à partir de réseaux entraînés sur les bases MICCAI 2012 (a, b, c, d, e) et IBSRv2 (f, g, h, i, j). Le bleu représente les adjacences autorisées et celles interdites en rouge. Les modèles sont les suivants (de gauche à droite) : 2D sans NonAdjLoss (a, f) ; 2D avec NonAdjLoss (b, g) ; 2D avec NonAdjLoss et semi-supervision (c, h) ; 2.5D avec fusion (d, i) ; 2.5D avec fusion et semi-supervision (e, j).

16.5 Adjacence 3D multi-orientation avec architecture 2.5D

La pénalisation NonAdjLoss spatialement orientée (multi-orientation cf section 12.1.2) fournit des contraintes anatomiques plus fines, cependant pour optimiser un modèle utilisant les six orientations 3D, il est nécessaire de disposer de cartes de segmentation 3D complètes. Pour cela, nous avons entraîné le réseau 2.5D proposé, avec une NonAdjLoss à 6 orientations appliquée aux 3 coupes produites en sortie. Pour étudier si l'adjacence non-orientée et orientée portent des informations discriminantes, plus précisément est-ce que certaines adjacences sont uniquement représentées à travers des orientations particulières, nous comparons les deux types de matrices. La figure 16.7 montre les matrices de dissimilarités entre les matrices non-orientées et orientées pour les bases MICCAI12 et IBSRv2, elles sont construites en sommant le nombre de différences entre chacune des matrices orientées et la version non-orientée. On constate dans cette dernière un nombre concluant de paires de régions qui sont spécifiques à des orientations, ce qui encourage l'utilisation de ce cas particulier de pénalisation.

Le tableau 16.5 montre que, bien que la NonAdjLoss orientée améliore les scores de non-adjacence CA sur les deux ensembles de données, elle dégrade légèrement les scores Dice, HD et MSD. Pour l'apprentissage semi-supervisé de la fonction de coût multi-orientée, la configuration expérimentale n'est pas la même que pour les expériences NonAdjLoss(50), en raison de contraintes de mémoire GPU car nous évaluons 6 fonctions de perte au lieu d'une. Nous ne pouvions pas stocker toutes les images et la taille du batch a été réduite. Ces dégradations relativement faibles pourraient également être dues à une sélection

MICCAI12	Dice	HD (mm)	MSD (mm)	CA^{unique}	CA^{volume}
EDNet 2D	0.740 ± 0.11	20.93 ± 9.50	1.18 ± 0.40	5.1e-2 ± 6.8e-2	1.8e-2 ± 4.9e-2
EDNet 2.5D + fusion	0.738 ± 0.11	16.06 ± 7.11	1.20 ± 0.39	1.9e-2 ± 3.1e-2	8.0e-3 ± 2.6e-2
2.5D + NAL(0) + Fus.	0.736 ± 0.10	12.10 ± 4.74	1.15 ± 0.38	4.0e-3 ± 1.1e-2	1.3e-3 ± 5.9e-3
2.5D + NAL(50) + Fus.	0.744 ± 0.10	**10.19 ± 3.73**	**1.05 ± 0.34**	3.2e-4 ± 1.4e-3	4.4e-5 ± 2.2e-4
2.5D + NAL(50) + Fus. + M-O	0.734 ± 0.10	10.27 ± 4.02	1.10 ± 0.34	**1.7e-4 ± 5.4e-4**	**1.2e-5 ± 4.4e-5**
IBSRv2					
EDNet 2D	0.833 ± 0.11	15.99 ± 15.27	0.78 ± 0.37	1.0e-1 ± 8.8e-2	1.5e-3 ± 3.0e-3
EDNet 2.5D + fusion	0.834 ± 0.11	12.60 ± 14.60	0.78 ± 0.34	5.6e-2 ± 5.5e-2	1.5e-3 ± 2.5e-2
2.5D + NAL(0) + Fus.	0.837 ± 0.10	**9.71 ± 10.39**	**0.75 ± 0.33**	5.3e-3 ± 1.2e-2	6.2e-5 ± 2.1e-4
2.5D + NAL(50) + Fus.	0.835 ± 0.10	10.94 ± 13.96	0.76 ± 0.32	5.3e-4 ± 2.8e-3	1.3e-6 ± 7.0e-6
2.5D + NAL(50) + Fus. + M-O	0.836 ± 0.10	9.99 ± 11.45	0.76 ± 0.35	**4.2e-4 ± 2.3e-3**	**1.3e-6 ± 7.0e-6**

TABLE 16.5 – Comparaison de l'architecture EDNet 2D, 2.5D, de la stratégie de fusion et de la multi-orientation. NAL représente la contrainte NonAdjLoss. Métriques de similarité, distances et de connectivité mesurées pour chaque modèle. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD, MSD, CA^{unique} et CA^{volume} sont moyennées sur l'ensemble de test. Nous reportons de la façon suivante : score moyen ± écart type.

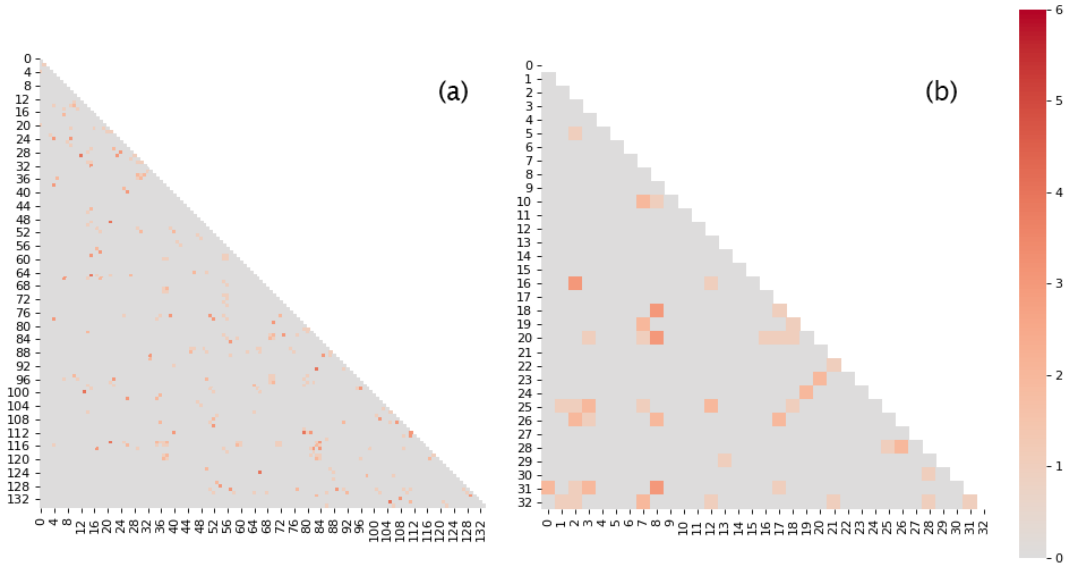


FIGURE 16.7 – Illustration de l'influence de l'orientation dans l'adjacence par rapport à la version non-orientée. Matrices de dissimilarités entre l'adjacence binaire non-orientée et les adjacences binaires orientées pour les jeux de données MICCAI12 (a) et IBSRv2 (b). Une dissimilarité entre la matrice non-orientée et une des matrices orientées indique qu'une adjacence à changé d'état (activation/désactivation). Plus le nombre de dissimilarités par rapport à la version non-orientée augmente, plus la valeur de la case tend vers le rouge, indiquant que l'adjacence est spécifique à l'orientation.

d'hyperparamètres sous-optimale, telle que la vitesse d'apprentissage ou l'initialisation des paramètres du réseau.

A plus long terme, nous pensons que la fonction de perte orientée sera une technique utile pour les applications médicales qui nécessitent une segmentation anatomique rigoureuse. La figure 16.8 montre la relation entre la distance de Hausdorff moyenne et le nombre de connexions incorrectes (échelle log) dans les bases MICCAI12 et IBSRv2. Elle suggère que pour la plupart des régions, la réduction des erreurs de connectivité diminue également

les distances de Hausdorff dans une certaine mesure. En effet, le modèle 2.5D orienté avec fusion NonAdjLoss(50) corrige un nombre important de régions, jusqu'à toutes les erreurs de contiguïté. Cependant, même pour ces régions la distance de Hausdorff n'est jamais nulle.

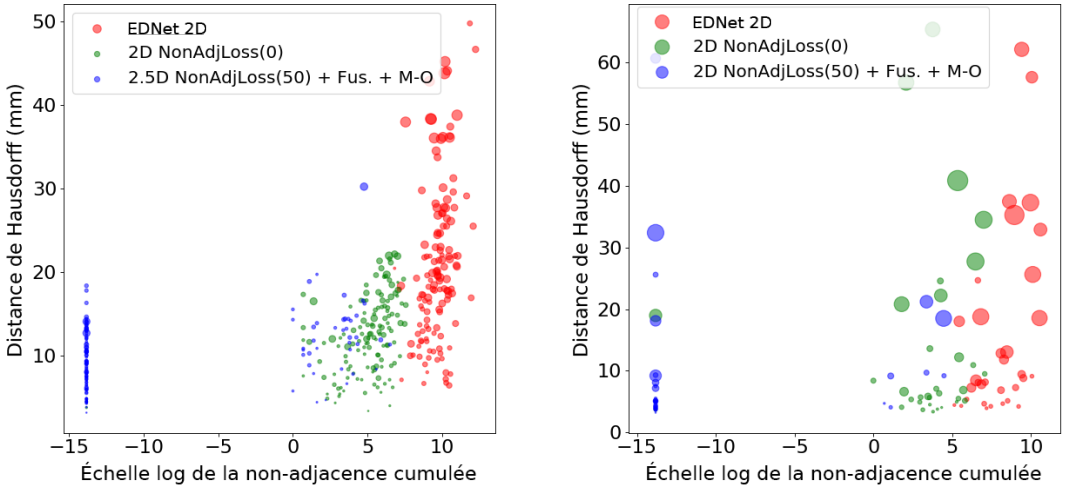


FIGURE 16.8 – Illustrations de l'influence de la non-adjacence sur la distance de Hausdorff pour MICCAI12 et IBSRv2. Le total des adjacences d'erreur est passé à l'échelle log et les régions sans erreurs sont égales à -14. Pour les distances de Hausdorff, le diamètre des points est proportionnel à leur écart-type.

16.6 Effet de la contrainte sur les cas problématiques

Pour comprendre l'impact de la contrainte NonAdjLoss sur le réalisme des segmentations produites par les modèles appris avec ou sans pénalisation, nous étudions dans cette section l'évolution de la distance de Hausdorff sur les données segmentées de la base de test de MICCAI12, individuellement pour les patients et les structures.

La figure 16.9 représente par des diagrammes en moustache la variation de la distance de Hausdorff, pour chacun des patients et en considérant toutes les structures sans distinction. On observe dans cette figure, que deux patients (1119 et 1128) se détachent particulièrement du groupe, avec une variance et une médiane significativement éloignées de la moyenne. C'est sur ces deux patients problématiques que l'on observe la plus forte amélioration, en ramenant les performances à un niveau inférieure. Même si l'impact sur les patients non-problématiques semblent plus limité en raison de l'échelle du graphique, on constate tout de même une amélioration constante pour tous les cas, dès l'utilisation de la contrainte NonAdjLoss, pouvant conduire à une nouvelle baisse, à l'aide de l'apprentissage semi-supervisé de la contrainte.

Au niveau des structures cérébrales, on observe dans la figure 16.10 la variation de la mesure de Hausdorff pour les 15 régions ayant les plus mauvaises performances sur les seg-

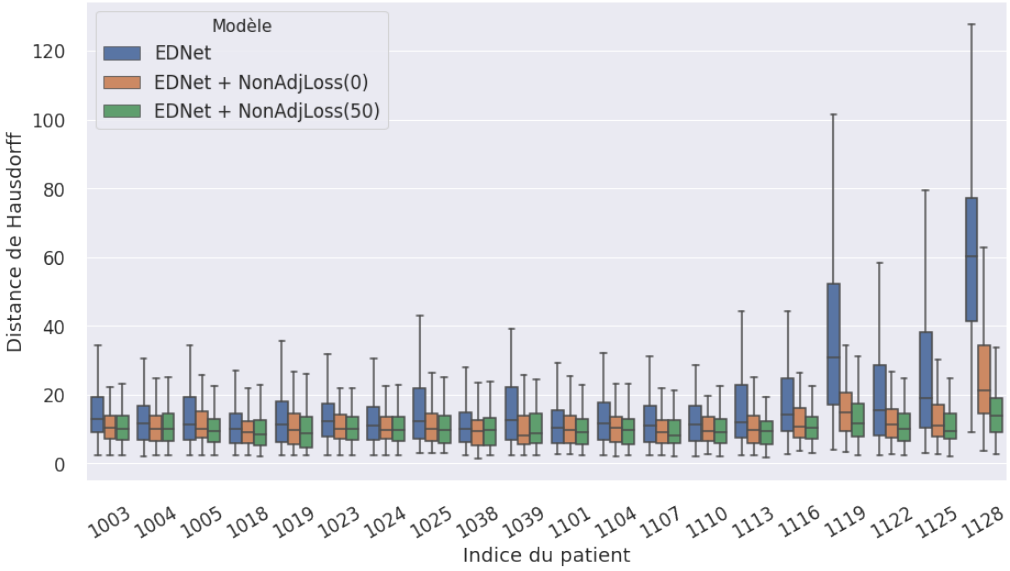


FIGURE 16.9 – Illustration de la dispersion de la distance de Hausdorff, toutes structures confondues pour chaque patients de la base de test MICCAI12. On compare les performances du réseau EDNet, avec l'ajout de la contrainte NonAdjLoss et l'utilisation de la semi-supervision.

mentations du modèle EDNet. Après l'ajout de la contrainte NonAdjLoss, on remarque une baisse de la médiane ou de la variance pour toutes ces régions, une tendance qui s'accroît généralement avec l'apprentissage semi-supervisé NonAdjLoss(50) sur la base OASIS.

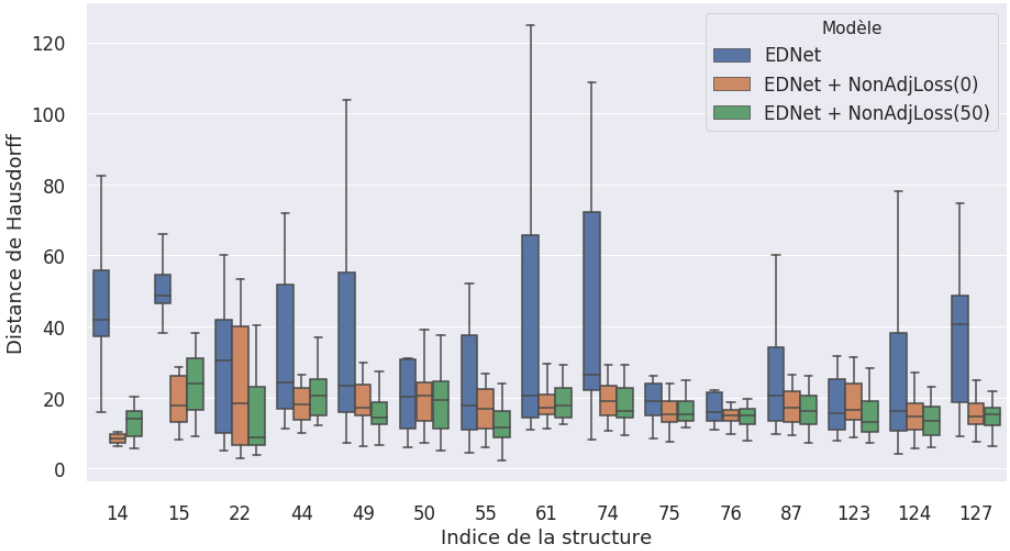


FIGURE 16.10 – Illustration de la dispersion de la distance de Hausdorff, pour toutes les segmentations des patients de la base de test MICCAI12, en fonction des 15 structures les plus mal délimitées. On compare les performances du réseau EDNet, avec l'ajout de la contrainte NonAdjLoss et l'utilisation de la semi-supervision.

L'introduction de connaissances de position spatiale dans un réseau par patch (partie II) suit le même objectif que celui de ce travail, à savoir l'utilisation d'*a priori* issu des données pour améliorer le réalisme des segmentations produites par le modèle. En ce sens,

il est logique de comparer les performances de ces deux approches pour comprendre leurs forces et faiblesses.

16.7 Comparaison RNC par patch et entièrement convolutif

Dans la section 13.2, nous avons présenté le processus d'apprentissage de la contrainte NonAdjLoss, qui se divise en deux étapes : le pré-apprentissage classique à partir des cartes d'annotations, puis le fine-tuning avec la pénalisation proposée. Pour évaluer les performances de l'architecture EDNet par rapport à une approche par patch, nous reportons les métriques de Dice, Hausdorff et distance surfacique moyenne pour les deux principaux modèles présentés (tableau 16.6). À savoir PatchNet le réseau par patch multi-résolution et PatchNet Full, la version améliorée comportant l'intégration de (tableau 16.7) descripteurs spatiaux et volumiques. L'architecture encodeur-décodeur proposée dans cette partie est aussi évalué dans sa version 2.5D (section 14.2) qui segmente 3 coupes en sortie (au lieu d'une seule pour la 2D). Cette dernière permet également un post-processing en fusionnant les probabilités de coupes qui se chevauchent (section 14.2).

MICCAI12	Dice	HD (mm)	MSD (mm)
PatchNet [Ganaye et al., 2018b]	0.694 ± 0.17	40.26 ± 40.12	1.74 ± 2.14
PatchNet Full	0.748 ± 0.14	9.66 ± 5.46	1.00 ± 0.59
EDNet [Ganaye et al., 2019]	0.740 ± 0.11	20.93 ± 9.50	1.18 ± 0.40
EDNet 2.5D	0.733 ± 0.11	19.77 ± 9.52	1.24 ± 0.40
EDNet 2.5D Fusion	0.738 ± 0.11	16.06 ± 7.11	1.20 ± 0.39
EDNet + NonAdjLoss(50)	0.741 ± 0.10	10.97 ± 4.37	1.04 ± 0.33
EDNet 2.5D + NonAdjLoss(50) + Fus.	0.744 ± 0.10	10.19 ± 3.73	1.05 ± 0.34

TABLE 16.6 – Comparaison des deux approches d'intégration d'informations dans un RNC proposées dans cette thèse. Métriques de similarité et distances mesurées pour chaque modèle sur la base de données MICCAI12. HD signifie distance de Hausdorff, MSD distance surfacique moyenne, toutes les deux en millimètres. Les mesures Dice, HD et MSD sont moyennées sur l'ensemble de test. Nous reportons de la façon suivante : score moyen ± écart type.

C'est l'ajout de la position du patch dans le réseau qui réduit la distance de Hausdorff de façon plus significative (tableau 16.6), avec le contre-coût d'un temps d'inférence de 9 minutes (tableau 16.7. À l'inverse, les approches encodeur-décodeur EDNet montre un potentiel d'amélioration important : performances initiales satisfaisantes, mais surtout le temps d'inférence est de moins d'une seconde. Le coût algorithmique de l'approche par patch (temps d'inférence ×100 par rapport à EDNet, cf tableau 16.7), ainsi que les résultats du tableau 16.6 démontrent que le réseau PatchNet Full possède des performances très proches au modèle EDNet 2.5D avec NonAdjLoss et fusion pour la base MICCAI 2012. Ce qui fait de EDNet 2.5D avec contrainte NonAdjLoss, l'approche la plus performante globalement si l'on considère la qualité de segmentation et la rapidité d'exécution, par rapport à PatchNet Full.

MICCAI12	Nb paramètres	temps inférence/image (min)
PatchNet [Ganaye et al., 2018b]	1 249 415	3.08
PatchNet Full	2 847 794	9.09
EDNet [Ganaye et al., 2019]	2 172 096	0.03
EDNet 2.5D	2 180 736	0.04
EDNet 2.5D Fusion	2 180 736	0.33

TABLE 16.7 – Comparaison de la complexité des modèles pour le nombre de paramètres à optimiser et le temps d'inférence pour segmenter une image.

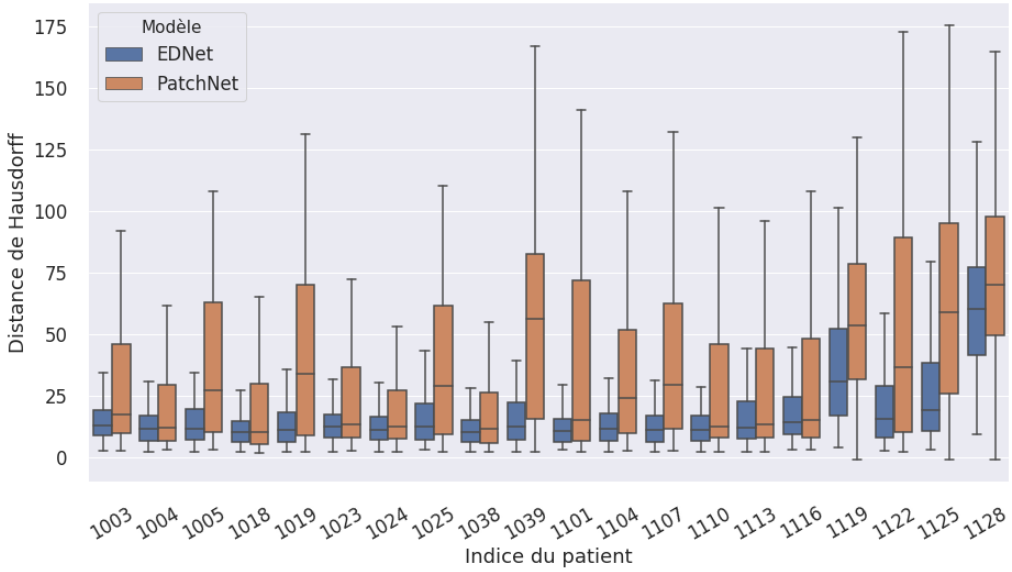


FIGURE 16.11 – Représentation de la variation de la distance de Hausdorff pour les patients segmentés de la base MICCAI12, en fonction de l'approche entièrement convolutive (EDNet) et d'un modèle par patch (PatchNet).

À l'échelle plus fine du patient, on observe dans la figure 16.11 la différence de performance pour la distance de Hausdorff entre les architectures par patch et FCN. On constate d'après cette figure que le réseau entièrement convolutif EDNet possède de meilleurs résultats, sans l'utilisation de contrainte spécifique. On peux attribuer cette différence importante à l'architecture FCN qui exploite totalement l'image d'entrée, à l'augmentation réaliste de données ou encore à la fonction de coût basée sur le Dice. Le réseau EDNet produit donc naturellement des segmentations plus proches du résultat attendu que l'approche par patch PatchNet.

Pour approfondir la comparaison des deux approches de segmentations proposées dans ce manuscrit, nous détaillons dans les figures 16.12 et 16.13 la variation de la distance de Hausdorff pour les images de la base test de MICCAI12, segmentées par les modèles EDNet, EDNet + NonAdjLoss(50) et PatchNet Full. On constate tout d'abord dans la figure 16.12 que les améliorations apportées par les deux approches d'intégration de l'*a priori* sont proches en terme de variance et de médiane. C'est une observation rassurante car si le réseau EDNet appris sous contrainte, ainsi que le réseau PatchNet, possèdent des

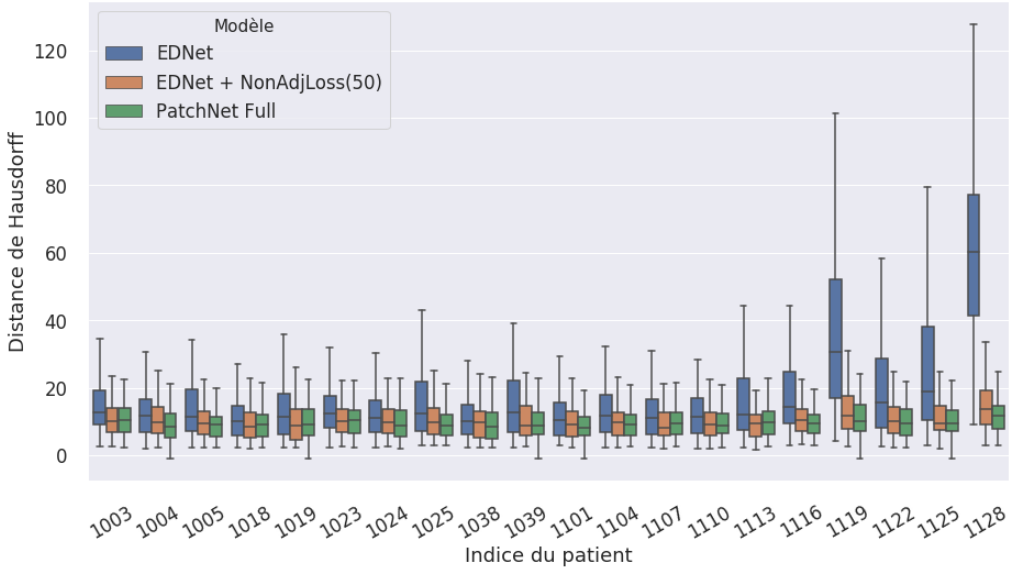


FIGURE 16.12 – Comparaison de la variation de la distance de Hausdorff pour les patients de la base MICCAI12 segmentés avec les modèles suivants : EDNet (architecture encodeur-décodeur), EDNet + NonAdjLoss(50) (réseau entraîné sous contrainte NonAdjLoss avec apprentissage semi-supervisé), PatchNet Full (réseau par patch intégrant plusieurs sources d'informations).

performances similaires en terme de qualité de segmentation, le réseau EDNet est beaucoup plus rapide pour effectuer le calcul des cartes de segmentation, ce qui démontre un gain considérable qui peut faire la différence dans un contexte d'utilisation clinique.

Au niveau des structures cérébrales, on remarque dans la figure 16.13 que le réseau par patch corrige en général mieux les erreurs que les autres modèles, toutefois les performances du modèle EDNet + NonAdjLoss(50) sont très proches. Pour un nombre important de ces régions, les médianes sont quasi-similaires à EDNet avec NonAdjLoss(50), ce qui rassure sur l'utilité de la contrainte NonAdjLoss. À noter que la figure 16.13 présente uniquement les structures les plus mal segmentées par les modèles PatchNet et EDNet.

Ces comparaisons de modèles entraînés sur des architectures différentes et avec une contrainte d'adjacence des structures anatomiques, nous a permis de mettre en évidence l'utilité de ces deux approches. Le modèle PatchNet Full semble être le vainqueur au sens strict, toutefois nous constatons que le réseau EDNet est très proche sur les mesures de Dice et de distances surfaciques, néanmoins avec un temps d'exécution largement réduit, ce qui en fait un favori pour une utilisation pratique.

16.8 Conclusion

Dans ce chapitre nous avons présenté les résultats obtenues sur plusieurs bases d'images lorsque l'on intègre la contrainte d'adjacence NonAdjLoss dans notre architecture EDNet. L'application de la contrainte de pénalisation proposée sur 3 jeux de données annotés démontre systématiquement une diminution du nombre d'adjacences interdites, d'après

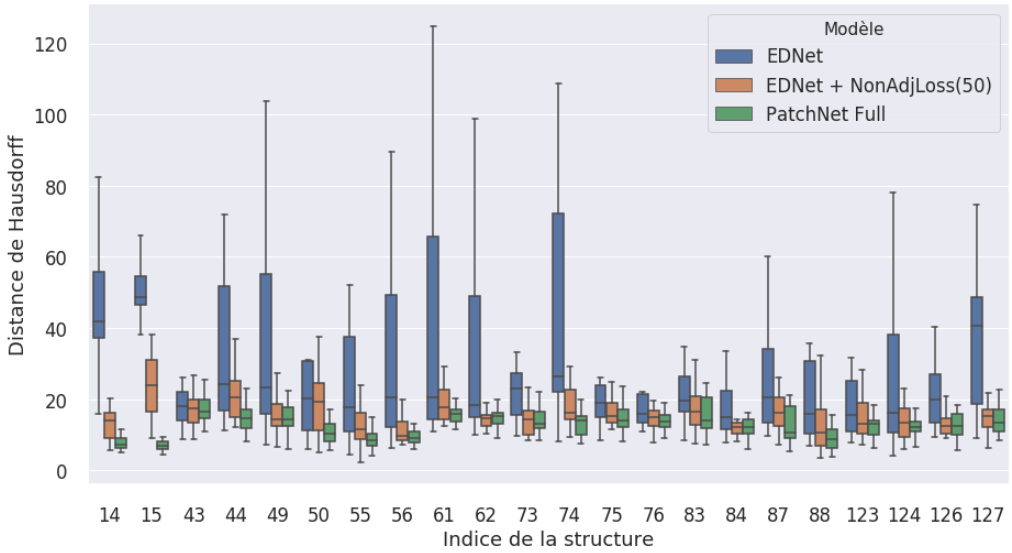


FIGURE 16.13 – Comparaison de la variation de la distance de Hausdorff pour les structures cérébrales des patients de la base MICCAI12 segmentés avec les modèles suivants : EDNet (architecture encodeur-décodeur), EDNet + NonAdjLoss(50) (réseau entraîné sous contrainte NonAdjLoss avec apprentissage semi-supervisé), PatchNet Full (réseau par patch intégrant plusieurs sources d'informations). Les structures étudiées possèdent les distances de Hausdorff les plus élevées pour les expériences PatchNet et EDNet.

les deux métriques de non-adjacence CA^{unique} et CA^{volume}. Cette amélioration s'observe visuellement par l'élimination d'une majeure partie des erreurs qui sont des aberrations pour l'oeil d'un expert. L'utilisation de la NonAdjLoss en 2D résulte globalement en une amélioration de la distance de Hausdorff, qui s'explique par la suppression des points dont l'adjacence est incorrecte. Les mesures de Dice restent majoritairement équivalentes aux niveaux initiaux, malgré quelques légères variations positives ou négatives. Les expériences pour démontrer l'utilité de contraintes avec des voisinages variables n'ont pas apporté de preuves significatives, toutefois pour d'autres applications de segmentation, l'idée pourrait se montrer utile. L'adaptation de l'architecture 2D en 2.5D, permet l'utilisation d'une méthode de post-traitement par fusion, qui réduit de quelques millimètres la distance de Hausdorff, pour les bases de données cérébrales. Cette modification donne également la possibilité de considérer les adjacences en fonction de chaque orientations, pour être plus proche des contraintes anatomiques réelles. Cette fonctionnalité donne les performances maximales pour les mesures de connectivité (CA^{unique} et CA^{volume}), sans affecter positivement le Dice ou la distance de Hausdorff.

Chapitre 17

Conclusion

Nous avons introduit NonAdjLoss, une fonction de coût en vue de supprimer les adjacences interdites de régions dans les segmentations anatomiques. Elle quantifie un indicateur anatomique qui était jusqu'alors évalué implicitement par les experts pour détecter des incohérences anatomiques. De plus la formulation proposée peut être intégrée à n'importe quel type de réseau de neurones avec une segmentation 2D ou 3D en sortie. Pour cela nous avons formulé une méthode pour extraire la connaissance d'adjacence structurelle à partir des données, que nous avons ensuite adaptée sous forme dérivable pour l'intégrer dans un RNC. Seule la procédure d'apprentissage du réseau est modifiée : l'architecture du réseau sous-jacent reste inchangée et il n'y a aucun coût supplémentaire lors de l'inférence. Nous avons également proposé une stratégie d'apprentissage semi-supervisé, qui tire partie de la plus forte accessibilité à des données non-annotées. Enfin, nous avons proposé l'intégration de cette contrainte 2D dans une architecture de réseau 2.5D, avec une méthode de post-traitement par fusion et une reformulation de la contrainte NonAdjLoss dans le cas où l'on dispose de règles d'adjacence spécifiques à l'orientation dans l'image.

D'après les expériences, on observe que même si la méthode a eu un effet limité sur le score de qualité de segmentation Dice, elle a nettement amélioré les mesures de distance de Hausdorff, de distance surfacique moyenne et de connectivité pour toutes les bases de données. L'utilisation de la semi-supervision permise par cette approche est utile dans le cas où la base de données est limitée en annotation, mais également pour améliorer la capacité de généralisation du modèle par rapport à la contrainte proposée, avec des images jamais observées dans les bases délimitées.

En conclusion, cette contrainte d'adjacence devrait être particulièrement utile pour les problèmes complexes de segmentation anatomique tels que la segmentation des régions corticales, car l'augmentation du nombre de régions anatomiques élargit également le nombre

de contraintes actives. La capacité de la méthode à gérer des données partiellement non annotées au cours de l'apprentissage est un autre avantage majeur, car elle donne la possibilité d'entraîner des modèles sur de plus grands ensembles de données.

Ce travail à donné lieu à une présentation orale lors de la conférence MICCAI 2018 [Ganaye et al., 2018a] en session plénière, puis à une publication dans les proceedings. Une version étendue des travaux a été acceptée pour publication dans la revue spéciale MICCAI 2018 du journal Medical Image Analysis (MIA) en Octobre 2019 [Ganaye et al., 2019].

IV Conclusion générale

Cette thèse a porté sur un enjeu important en imagerie médicale à savoir l'utilisation de modèles de segmentation automatiques respectueux des connaissances anatomiques. C'est une problématique vaste à traiter en raison de la diversité et du nombre de propriétés anatomiques que l'on peut observer dans les données médicales, mais aussi en raison de la difficulté à les modéliser sous la forme d'une contrainte dérivable lors de l'apprentissage d'un modèle supervisé de type RNC.

Dans ce manuscrit, nous avons exploré deux méthodologies afin d'intégrer des connaissances liées aux structures anatomiques, lors de l'apprentissage par réseau de neurones.

Dans la partie II, nous avons proposé une architecture par patch multi-résolution, dans laquelle nous avons intégré plusieurs autres sources d'informations, dont une image de distances encodant la position spatiale du patch dans l'image d'entrée et un atlas probabiliste extrait à partir des données d'apprentissage. Avec ces nouvelles connaissances, nous observons une amélioration de la qualité globale de la segmentation, ainsi qu'une forte réduction de la distance de Hausdorff moyenne, indiquant que les segmentations produites par le nouveau modèle sont plus réalistes par rapport à la vérité terrain. Toutefois, les approches par patch sont notablement lentes en raison du caractère itératif de la segmentation, où chacun des pixels est classifié indépendamment.

Dans la partie III, nous avons étudié les relations d'adjacence inter-structures afin d'extraire un *a priori* anatomique à partir des annotations des images. Pour intégrer cette connaissance dans une architecture FCN 2D (EDNet), une fonction de coût nommée NonAdjLoss a été formulée pour mesurer les adjacences structurelles, permettant ainsi de formaliser une contrainte dérivable de non-adjacence entre les régions anatomiques. Nous avons proposé une version 2.5D de l'architecture EDNet, pour laquelle la contrainte NonAdjLoss a été étendue pour considérer des règles d'adjacence spécifiques à l'orientation dans l'image. Au cours des expériences sur trois bases de données d'imagerie médicale (cerveau et corps entier), nous avons constaté que la pénalisation NonAdjLoss réduit efficacement les erreurs d'adjacence en sortie du modèle et que la généralisation sur de nouvelles données est bonne. La contrainte NonAdjLoss appliquée lors de l'apprentissage des paramètres permet une réduction importante de la distance de Hausdorff pour toutes les bases de données, tout en préservant la qualité globale de segmentation.

À travers nos deux principales contributions [Ganaye et al., 2018b, Ganaye et al., 2019], nous avions le même objectif méthodologique, à savoir augmenter le réalisme des approches de segmentation par réseaux de neurones, en utilisant des connaissances extraites des données qui n'étaient jusqu'alors pas prises en compte pour la prédiction. Si l'on compare les résultats finaux des travaux présentés en partie II et III (section 16.7), on obtient des performances similaires en terme de Dice (0.748 contre 0.744) et distance de Hausdorff (9.66mm contre 10.19). Même si le réseau par patch est le supérieur au sens strict de ces critères, c'est de notre point de vue la deuxième approche qui offre les perspectives les plus intéressantes. En effet d'un point de vue pratique, le réseau FCN 2D proposé avec la

contrainte NonAdjLoss requière environ une seconde pour effectuer la segmentation, contre plusieurs minutes pour le réseau par patch. Au delà de cet aspect matériel qui pourrait être amélioré avec des solutions d'ingénierie, les approches par patch ne peuvent pas intégrer des contraintes de haut-niveau telles que la non-adjacence structurelle, ce sont par conséquent les architectures FCNs que nous favorisons pour les réflexions futures. Nous espérons que la contrainte NonAdjLoss pourra trouver une utilité dans des applications de segmentation d'image médicale, où des règles fortes d'agencement structurel existent.

Dans le sens des travaux entrepris dans cette thèse, nous considérons la modélisation de propriétés anatomiques et son intégration dans un FCN, comme une piste de recherche sérieuse à poursuivre. Ces propriétés peuvent s'exprimer sous la forme de règles géométriques, dans la relation ou l'organisation inter-structure, dans la finalité de traduire le plus fidèlement ce qui est observable dans les données. Le challenge se pose alors de capter des informations sémantiques automatiquement et de les intégrer dans la fonction de coût d'un FCN.

Bibliographie

[Aljabar et al., 2007] Aljabar, P., Heckemann, R., Hammers, A., Hajnal, J. V., and Rueckert, D. (2007). Classifier selection strategies for label fusion using large atlas databases. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 523–531. Springer.

[Aljabar et al., 2009] Aljabar, P., Heckemann, R. A., Hammers, A., Hajnal, J. V., and Rueckert, D. (2009). Multi-atlas based segmentation of brain images : atlas selection and its effect on accuracy. *Neuroimage*, 46(3) :726–738.

[Anbeek et al., 2013] Anbeek, P., Išgum, I., van Kooij, B. J. M., Mol, C. P., Kersbergen, K. J., Groenendaal, F., Viergever, M. A., de Vries, L. S., and Benders, M. J. N. L. (2013). Automatic segmentation of eight tissue classes in neonatal brain mri. *PLOS ONE*, 8(12).

[Anbeek et al., 2005] Anbeek, P., Vincken, K. L., van Bochove, G. S., van Osch, M. J., and van der Grond, J. (2005). Probabilistic segmentation of brain tissue in mr imaging. *NeuroImage*, 27(4) :795 – 804.

[Artaechevarria et al., 2009] Artaechevarria, X., Munoz-Barrutia, A., and Ortiz-de Solórzano, C. (2009). Combination strategies in multi-atlas image segmentation : application to brain mr data. *IEEE transactions on medical imaging*, 28(8) :1266–1277.

[Ashburner, 2007] Ashburner, J. (2007). A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1) :95–113.

[Asman and Landman, 2013] Asman, A. J. and Landman, B. A. (2013). Non-local statistical label fusion for multi-atlas segmentation. *Medical image analysis*, 17(2) :194–208.

[Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Bai et al., 2013] Bai, W., Shi, W., O'regan, D. P., Tong, T., Wang, H., Jamil-Copley, S., Peters, N. S., and Rueckert, D. (2013). A probabilistic patch-based label fusion model

for multi-atlas segmentation with registration refinement : application to cardiac mr images. *IEEE transactions on medical imaging*, 32(7) :1302–1315.

[Baumgartner et al., 2019] Baumgartner, C. F., Tezcan, K. C., Chaitanya, K., Hötker, A. M., Muehlematter, U. J., Schawkat, K., Becker, A. S., Donati, O., and Konukoglu, E. (2019). PHiSeg : Capturing Uncertainty in Medical Image Segmentation. *arXiv e-prints*, page arXiv :1906.04045.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf : Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

[Belaroussi et al., 2006] Belaroussi, B., Milles, J., Carme, S., Zhu, Y. M., and Benoit-Cattin, H. (2006). Intensity non-uniformity correction in mri : Existing methods and their validation. *Medical Image Analysis*, 10(2) :234 – 246.

[BenTaieb and Hamarneh, 2016] BenTaieb, A. and Hamarneh, G. (2016). Topology aware fully convolutional networks for histology gland segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 460–468. Springer.

[Bottou, 2010] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer.

[Chen et al., 2016] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915.

[Chen et al., 2014] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv :1412.7062*.

[Chen et al., 2017a] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017a). Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4) :834–848.

[Chen et al., 2017b] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017b). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv :1706.05587*.

[Chiu et al., 2015] Chiu, S. J., Allingham, M. J., Mettu, P. S., Cousins, S. W., Izatt, J. A., and Farsiu, S. (2015). Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema. *Biomed. Opt. Express*, 6(4) :1172–1194.

[Collewet et al., 2004] Collewet, G., Strzelecki, M., and Mariette, F. (2004). Influence of mri acquisition protocols and image intensity normalization methods on texture classification. *Magnetic Resonance Imaging*, 22(1) :81 – 91.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5) :603–619.

[Cook and Koles, 2006] Cook, M. J. D. and Koles, Z. J. (2006). A high-resolution anisotropic finite-volume head model for eeg source analysis. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4536–4539.

[Cordier et al., 2016] Cordier, N., Delingette, H., and Ayache, N. (2016). A patch-based approach for the segmentation of pathologies : Application to glioma labelling. *IEEE Transactions on Medical Imaging*, 35(4) :1066–1076.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3) :273–297.

[Coupé et al., 2011] Coupé, P., Manjón, J. V., Fonov, V., Pruessner, J., Robles, M., and Collins, D. L. (2011). Patch-based segmentation using expert priors : Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2) :940 – 954.

[de Brebisson and Montana, 2015] de Brebisson, A. and Montana, G. (2015). Deep Neural Networks for Anatomical Brain Segmentation. *arXiv :1502.02445 [cs, stat]*. arXiv : 1502.02445.

[Ferree et al., 2000] Ferree, T. C., Eriksen, K. J., and Tucker, D. M. (2000). Regional head tissue conductivity estimation for improved eeg analysis. *IEEE Transactions on Biomedical Engineering*, 47(12) :1584–1592.

[Fisher et al., 2008] Fisher, E., Lee, J.-C., Nakamura, K., and Rudick, R. A. (2008). Gray matter atrophy in multiple sclerosis : a longitudinal study. *Annals of Neurology : Official Journal of the American Neurological Association and the Child Neurology Society*, 64(3) :255–265.

[Friedman, 1997] Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1) :55–77.

[Friston et al., 1995] Friston, K. J., Ashburner, J., Frith, C. D., Poline, J.-B., Heather, J. D., and Frackowiak, R. S. (1995). Spatial registration and normalization of images. *Human brain mapping*, 3(3) :165–189.

[Fuchs et al., 2007] Fuchs, M., Wagner, M., and Kastner, J. (2007). Development of volume conductor and source models to localize epileptic foci. *Journal of Clinical Neurophysiology*, 24(2) :101–119.

[Ganaye et al., 2018a] Ganaye, P.-A., Sdika, M., and Benoit-Cattin, H. (2018a). Semi-supervised learning for segmentation under semantic constraint. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 595–602, Cham. Springer International Publishing.

[Ganaye et al., 2018b] Ganaye, P.-A., Sdika, M., and Benoit-Cattin, H. (2018b). Towards integrating spatial localization in convolutional neural networks for brain image segmentation. In *Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on*, pages 621–625. IEEE.

[Ganaye et al., 2019] Ganaye, P.-A., Sdika, M., Triggs, B., and Benoit-Cattin, H. (2019). Removing segmentation inconsistencies with semi-supervised non-adjacency constraint. *Medical Image Analysis*, 58 :101551.

[Ghafoorian et al., 2017a] Ghafoorian, M., Karssemeijer, N., Heskes, T., Bergkamp, M., Wissink, J., Obels, J., Keizer, K., de Leeuw, F.-E., Ginneken, B., Marchiori, E., and Platel, B. (2017a). Deep multi-scale location-aware 3d convolutional neural networks for automated detection of lacunes of presumed vascular origin. *NeuroImage : Clinical*, 14(Supplement C) :391 – 399.

[Ghafoorian et al., 2017b] Ghafoorian, M., Karssemeijer, N., Heskes, T., Bergkamp, M., Wissink, J., Obels, J., Keizer, K., Leeuw, F.-E. d., Ginneken, B., Marchiori, E., and Platel, B. (2017b). Deep multi-scale location-aware 3d convolutional neural networks for automated detection of lacunes of presumed vascular origin. *NeuroImage : Clinical*, 14 :391–399.

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Goodfellow et al., 2013] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. *arXiv preprint arXiv :1302.4389*.

[Han et al., 2015] Han, S., Mao, H., and Dally, W. J. (2015). Deep compression : Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv :1510.00149*.

[Havaei et al., 2017] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M., and Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35 :18 – 31.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

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

[Heckemann et al., 2006] Heckemann, R. A., Hajnal, J. V., Aljabar, P., Rueckert, D., and Hammers, A. (2006). Automatic anatomical brain mri segmentation combining label propagation and decision fusion. *NeuroImage*, 33(1) :115 – 126.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Huang et al., 2013] Huang, Y., Dmochowski, J. P., Su, Y., Datta, A., Rorden, C., and Parra, L. C. (2013). Automated MRI segmentation for individualized modeling of current flow in the human head. *Journal of Neural Engineering*, 10(6) :066004.

[Jenkinson et al., 2002] Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *Neuroimage*, 17(2) :825–841.

[Jimenez-del-Toro et al., 2016] Jimenez-del-Toro, O., Müller, H., Krenn, M., Gruenberg, K., Taha, A. A., Winterstein, M., Eggel, I., Foncubierta-Rodríguez, A., Goksel, O., Jakab, A., Kontokotsios, G., Langs, G., Menze, B. H., Salas Fernandez, T., Schaer, R., Walleyo, A., Weber, M., Dicente Cid, Y., Gass, T., Heinrich, M., Jia, F., Kahl, F., Kechichian, R., Mai, D., Spanier, A. B., Vincent, G., Wang, C., Wyeth, D., and Hanbury, A. (2016). Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms : Visceral anatomy benchmarks. *IEEE Transactions on Medical Imaging*, 35(11) :2459–2475.

[Kervadec et al., 2019] Kervadec, H., Dolz, J., Tang, M., Granger, E., Boykov, Y., and Ayed, I. B. (2019). Constrained-cnn losses for weakly supervised segmentation. *Medical Image Analysis*, 54 :88 – 99.

[Klein et al., 2017] Klein, A., Ghosh, S. S., Bao, F. S., Giard, J., Häme, Y., Stavsky, E., Lee, N., Rossa, B., Reuter, M., Chaibub Neto, E., and Keshavan, A. (2017). Mindboggling morphometry of human brains. *PLOS Computational Biology*, 13(2) :1–40.

[Krähenbühl and Koltun, 2011] Krähenbühl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kuczyński et al., 2010] Kuczyński, K., Stegierski, R., and Siczek, M. (2010). Brain atrophy progress detection in mr iimages. *Journal of Medical Informatics and Technologies*, 16.

[Kwong et al., 1992] Kwong, K. K., Belliveau, J. W., Chesler, D. A., Goldberg, I. E., Weisskoff, R. M., Poncelet, B. P., Kennedy, D. N., Hoppel, B. E., Cohen, M. S., and Turner, R.

(1992). Dynamic magnetic resonance imaging of human brain activity during primary sensory stimulation. *Proceedings of the National Academy of Sciences*, 89(12) :5675–5679.

[Landman, 2012] Landman, B. (2012). Miccai 2012 workshop on multi-atlas labeling. In *MICCAI Grand Challenge and Workshop on Multi-Atlas Labeling*.

[LeCun and Bengio, 1998] LeCun, Y. and Bengio, Y. (1998). The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA.

[Lee et al., 2011] Lee, N., Laine, A. F., and Klein, A. (2011). Towards a deep learning approach to brain parcellation. In *2011 IEEE International Symposium on Biomedical Imaging : From Nano to Macro*, pages 321–324.

[Li et al., 2016] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2016). Hyperband : A novel bandit-based approach to hyperparameter optimization. *arXiv preprint arXiv :1603.06560*.

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

[Liu et al., 2018] Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., and Murphy, K. (2018). Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34.

[Long et al., 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110.

[Maas et al.,] Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models.

[Madabhushi and Udupa, 2006] Madabhushi, A. and Udupa, J. K. (2006). New methods of mr image intensity standardization via generalized scale. *Medical Physics*, 33(9) :3426–3434.

[Marcus et al., 2010a] Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2010a). Open access series of imaging studies : Longitudinal mri data in nondemented and demented older adults. *Journal of Cognitive Neuroscience*, 22(12) :2677–2684. PMID : 19929323.

[Marcus et al., 2010b] Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C., and Buckner, R. L. (2010b). Open access series of imaging studies : longitudinal mri data in nondemented and demented older adults. *Journal of cognitive neuroscience*, 22(12) :2677–2684.

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

[Moeskops et al., 2016] Moeskops, P., Viergever, M. A., Mendrik, A. M., Vries, L. S. d., Benders, M. J. N. L., and Išgum, I. (2016). Automatic Segmentation of MR Brain Images With a Convolutional Neural Network. *IEEE Transactions on Medical Imaging*, 35(5) :1252–1261.

[Moritz et al., 2018] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., et al. (2018). Ray : A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pages 561–577.

[Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical optimization, chapter 17*. Springer Science & Business Media.

[Noh et al., 2015] Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528.

[Nyul et al., 2000] Nyul, L. G., Udupa, J. K., and Xuan Zhang (2000). New variants of a method of mri scale standardization. *IEEE Transactions on Medical Imaging*, 19(2) :143–150.

[Oktay et al., 2018] Oktay, O., Ferrante, E., Kamnitsas, K., Heinrich, M., Bai, W., Caballero, J., Cook, S. A., de Marvao, A., Dawes, T., O'Regan, D. P., Kainz, B., Glocker, B., and Rueckert, D. (2018). Anatomically constrained neural networks (acnns) : Application to cardiac image enhancement and segmentation. *IEEE TMI*, 37(2) :384–395.

[Painchaud et al., 2019] Painchaud, N., Skandarani, Y., Judge, T., Bernard, O., Lalande, A., and Jodoin, P.-M. (2019). Cardiac MRI Segmentation with Strong Anatomical Guarantees. *arXiv e-prints*, page arXiv :1907.02865.

[Phan et al., 2017] Phan, T. V., Smeets, D., Talcott, J., and Vandermosten, M. (2017). Processing of structural neuroimaging data in young children : Bridging the gap between current practice and state-of-the-art methods. *Developmental Cognitive Neuroscience*, 33.

[Powell et al., 2008] Powell, S., Magnotta, V. A., Johnson, H., Jammalamadaka, V. K., Pierson, R., and Andreasen, N. C. (2008). Registration and machine learning-based automated segmentation of subcortical and cerebellar brain structures. *NeuroImage*, 39(1) :238 – 247.

[Ravishankar et al., 2017] Ravishankar, H., Venkataramani, R., Thiruvenkadam, S., Sudhakar, P., and Vaidya, V. (2017). Learning and incorporating shape models for semantic segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 203–211. Springer.

[Rohlfing et al., 2004] Rohlfing, T., Brandt, R., Menzel, R., and Maurer Jr, C. R. (2004). Evaluation of atlas selection strategies for atlas-based image segmentation with application to confocal microscopy images of bee brains. *NeuroImage*, 21(4) :1428–1442.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

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

[Rousseau et al., 2011] Rousseau, F., Habas, P. A., and Studholme, C. (2011). A supervised patch-based approach for human brain labeling. *IEEE transactions on medical imaging*, 30(10) :1852–1862.

[Roy et al., 2017] Roy, A. G., Conjeti, S., Sheet, D., Katouzian, A., Navab, N., and Wachinger, C. (2017). Error corrective boosting for learning fully convolutional networks with limited data. In *MICCAI*, pages 231–239. Springer.

[Rumelhart et al., 1995] Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation : The basic theory. *Backpropagation : Theory, architectures and applications*, pages 1–34.

[Sciuto et al., 2019] Sciuto, C., Yu, K., Jaggi, M., Musat, C., and Salzmann, M. (2019). Evaluating the Search Phase of Neural Architecture Search. *arXiv e-prints*, page arXiv :1902.08142.

[Sdika, 2008] Sdika, M. (2008). A fast nonrigid image registration with constraints on the Jacobian using large scale constrained optimization. *IEEE Transactions on Medical Imaging*, 27 :271–81.

[Sdika, 2010] Sdika, M. (2010). Combining atlas based segmentation and intensity classification with nearest neighbor transform and accuracy weighted vote. *Med Image Anal*, 14(2) :219–26. 1361-8423 (Electronic) 1361-8415 (Linking) Journal Article Research Support, Non-U.S. Gov't.

[Sdika, 2013] Sdika, M. (2013). A Sharp Sufficient Condition for B-Spline Vector Field Invertibility. Application to Diffeomorphic Registration and Interslice Interpolation. *SIAM Journal on Imaging Sciences*, 6(4) :2236–2257.

[Sdika, 2015] Sdika, M. (2015). Enhancing atlas based segmentation with multiclass linear classifiers. *Medical physics*, 42 :7169.

[Sdika and Pelletier, 2009] Sdika, M. and Pelletier, D. (2009). Nonrigid registration of multiple sclerosis brain images using lesion inpainting for morphometry or lesion mapping. *Human Brain Mapping*, 30 :1060–7.

[Simard et al., 2003] Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963.

[Simkó et al., 2019] Simkó, A., Löfstedt, T., Garpebring, A., Nyholm, T., and Jonsson, J. (2019). A generalized network for {mri} intensity normalization. In *International Conference on Medical Imaging with Deep Learning – Extended Abstract Track*, London, United Kingdom.

[Simon et al., 1999] Simon, J., Jacobs, L., Campion, M., Rudick, R., Cookfair, D., Herndon, R., Richert, J., Salazar, A., Fischer, J., Goodkin, D., et al. (1999). A longitudinal study of brain atrophy in relapsing multiple sclerosis. *Neurology*, 53(1) :139–139.

[Sled et al., 1998] Sled, J. G., Zijdenbos, A. P., and Evans, A. C. (1998). A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE Transactions on Medical Imaging*, 17(1) :87–97.

[Snoek et al., 2012] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

[Sotiras et al., 2013] Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration : A survey. *IEEE Transactions on Medical Imaging*, 32(7) :1153–1190.

[Srivastava et al., 2014] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout : a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1) :1929–1958.

[Sudre et al., 2017] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In Cardoso, M. J., Arbel, T., Carneiro, G., Syeda-Mahmood, T., Tavares, J. M. R., Moradi, M., Bradley, A., Greenspan, H., Papa, J. P., Madabhushi, A., Nascimento, J. C., Cardoso, J. S., Belagiannis, V., and Lu, Z., editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Cham. Springer International Publishing.

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Szeliski and Coughlan, 1997] Szeliski, R. and Coughlan, J. (1997). Spline-based image registration. *International Journal of Computer Vision*, 22(3) :199–218.

[Tanenbaum et al., 2017] Tanenbaum, L., Tsiouris, A., Johnson, A., Naidich, T., DeLano, M., Melhem, E., Quarterman, P., Parameswaran, S., Shankaranarayanan, A., Goyen, M., and Field, A. (2017). Synthetic mri for clinical neuroimaging : Results of the magnetic resonance image compilation (magic) prospective, multicenter, multireader trial. *American Journal of Neuroradiology*, 38(6) :1103–1110.

[Tustison et al., 2010] Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., and Gee, J. C. (2010). N4itk : improved n3 bias correction. *IEEE TMI*, 29(6) :1310–1320.

[Vercauteren et al., 2009] Vercauteren, T., Pennec, X., Perchant, A., and Ayache, N. (2009). Diffeomorphic demons : Efficient non-parametric image registration. *NeuroImage*, 45(1) :S61–S72.

[Vrooman et al., 2007] Vrooman, H. A., Cocosco, C. A., van der Lijn, F., Stokking, R., Ikram, M. A., Vernooij, M. W., Breteler, M. M., and Niessen, W. J. (2007). Multi-spectral brain tissue segmentation using automatically trained k-nearest-neighbor classification. *NeuroImage*, 37(1) :71 – 81.

[Wang et al., 2012] Wang, H., Suh, J. W., Das, S. R., Pluta, J. B., Craige, C., and Yushkevich, P. A. (2012). Multi-atlas segmentation with joint label fusion. *IEEE transactions on pattern analysis and machine intelligence*, 35(3) :611–623.

[Wang and Yushkevich, 2013] Wang, H. and Yushkevich, P. A. (2013). Multi-atlas segmentation with joint label fusion and corrective learning—an open source implementation. *Frontiers in neuroinformatics*, 7.

[Wolters et al., 2006] Wolters, C., Anwander, A., Tricoche, X., Weinstein, D., Koch, M., and MacLeod, R. (2006). Influence of tissue conductivity anisotropy on eeg/meg field and return current computation in a realistic head model : A simulation and visualization study using high-resolution finite element modeling. *NeuroImage*, 30(3) :813 – 826.

[Worth, 2003] Worth, A. (2003). *Internet Brain Segmentation Repository*.

[Wu et al., 2007] Wu, M., Rosano, C., Lopez-Garcia, P., Carter, C. S., and Aizenstein, H. J. (2007). Optimum template selection for atlas-based segmentation. *NeuroImage*, 34(4) :1612–1618.

[Xie et al., 2019] Xie, S., Kirillov, A., Girshick, R., and He, K. (2019). Exploring Randomly Wired Neural Networks for Image Recognition. *arXiv e-prints*, page arXiv :1904.01569.

[Yang et al., 2017] Yang, T.-J., Chen, Y.-H., and Sze, V. (2017). Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5687–5695.

[Yu and Koltun, 2015] Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv :1511.07122*.

[Yu-Feng et al., 2007] Yu-Feng, Z., Yong, H., Chao-Zhe, Z., Qing-Jiu, C., Man-Qiu, S., Meng, L., Li-Xia, T., Tian-Zi, J., and Yu-Feng, W. (2007). Altered baseline brain activity in children with adhd revealed by resting-state functional mri. *Brain and Development*, 29(2) :83–91.

[Zoph and Le, 2016] Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv :1611.01578*.

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : Ganaye
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 26/11/2019

Prénoms : Pierre-Antoine, Abel, Hilaire

TITRE : A priori et apprentissage profond pour la segmentation en imagerie cérébrale

NATURE : Doctorat

Numéro d'ordre :

Ecole doctorale : ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE (EEA)

Spécialité : Traitement du signal et de l'image

RESUME :
L'imagerie médicale est un domaine vaste guidé par les avancées en instrumentation, en techniques d'acquisition et en traitement d'images. Les progrès réalisés dans ces grandes disciplines concourent tous à l'amélioration de la compréhension de phénomènes physiologiques comme pathologiques.

En parallèle, l'accès à des bases de données d'imagerie plus large, associé au développement de la puissance de calcul, a favorisé le développement de méthodologies par apprentissage machine pour le traitement automatique des images dont les approches basées sur des réseaux de neurones profonds. Parmi les applications où les réseaux de neurones profonds apportent des solutions, on trouve la segmentation d'images qui consiste à localiser et délimiter dans une image les régions avec des propriétés spécifiques qui seront associées à une même structure. Malgré de nombreux travaux récents en segmentation d'images par réseaux de neurones, l'apprentissage des paramètres d'un réseau de neurones reste guidé par des mesures de performances quantitatives n'incluant pas la connaissance de haut niveau de l'anatomie.

L'objectif de cette thèse est de développer des méthodes permettant d'intégrer des a priori dans des réseaux de neurones profonds, en ciblant la segmentation de structures cérébrales en imagerie IRM. Notre première contribution propose une stratégie d'intégration de la position spatiale du patch à classifier, pour améliorer le pouvoir discriminant du modèle de segmentation. Ce premier travail corrige considérablement les erreurs de segmentation étant très éloignées de la réalité anatomique, en améliorant également la qualité globale des résultats. Notre deuxième contribution est ciblée sur une méthodologie pour contraindre les relations d'adjacence entre les structures anatomiques, et ce directement lors de l'apprentissage des paramètres du réseau, dans le but de renforcer le réalisme des segmentations produites. Nos expériences permettent de conclure que la contrainte proposée corrige les adjacences non-admises, améliorant ainsi la consistance anatomique des segmentations produites par le réseau de neurones.

MOTS-CLÉS : segmentation d'images médicales, apprentissage profond, a priori, contrainte anatomique

Laboratoire (s) de recherche : Centre de Recherche en Acquisition et Traitement de l'Image pour la Santé (CREATIS)

Directeur de thèse: Hugues Benoit-Cattin

Président de jury :

Composition du jury :

Petitjean, Caroline	Maître de conférences HDR	Université de Rouen	Rapporteur
Thome, Nicolas	Professeur des Universités	CNAM	Rapporteur
Garcia, Christophe	Professeur des Universités	INSA Lyon	Examinateur
Jodoin, Pierre-Marc	Professeur des Universités	Université de Sherbrooke	Examinateur
Benoit-Cattin, Hugues	Professeur des Universités	INSA Lyon	Directeur de thèse