



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra



Rapport de Stage

Master 2 Informatique option SDR spécialité recherche année 2016-2017

Présenté par

LISA CORBAT

Sujet du stage recherche :

Segmentation d'Images Médicales par Deep Learning

-
Étude et Implémentation

Tuteur :

M. Julien HENRIET

Remerciements

Je tiens à remercier tout particulièrement M. Julien HENRIET ainsi que M. Jean-Cristophe LAPAYRE pour le soutien apporté tout au long de ce stage, ainsi que leur confiance qui m'a été accordée dans ce grand projet.

Je tiens également à remercier M. Florent MARIE, doctorant au laboratoire FEMTO-ST et M.Thibault DELAVELLE, stagiaire sur le projet SAIAD, pour l'aide et les conseils apportés au bon avancement de ce stage.

Je remercie aussi M. Maxime MARTIN de m'avoir aidé dans diverses configurations et installations sur mon ordinateur.

Je remercie également tous les doctorants du laboratoire pour la bonne ambiance et leur soutien moral.

Je remercie l'Interreg pour avoir porté le projet SAIAD sans lequel mon stage n'aurait été possible.

Pour finir, je remercie également le soutien du Laboratoire d'excellence Labex ACTION (contrat ANR-11-LABX-0001-01).

Table des matières

Nomenclature	11
Introduction	13
Contexte du travail : le projet SAIAD	13
Objectif de ces travaux de master recherche	15
1 Les réseaux de neurones	17
1.1 Perceptron	17
1.2 Le perceptron multicouche	19
1.3 Le Deep Learning	20
1.3.1 Le principe	20
1.3.2 Le processus d'apprentissage	20
Conclusion	24
2 Les réseaux de neurones convolutifs	25
2.1 Le principe	25
2.2 Les différentes couches	27
2.2.1 La couche de Convolution	27
2.2.2 La couche de pooling	30
2.2.3 La couche de correction	31
2.2.4 La couche <i>Fully Connected</i>	32
2.2.5 La couche de perte	32
2.3 Exemple d'architecture de classification d'images basée sur les CNN	32
2.3.1 Le principe du dropout	33
Conclusion	34
3 Les segmentations avec Deep Learning	35
3.1 Les différentes méthodes de segmentation	35
3.1.1 <i>Fully Convolutional Networks</i>	35
3.1.2 <i>Deconvolution Network</i> et <i>Decoupled Network</i>	38
3.1.3 SegNet	40
3.1.4 U-Net	41
3.1.5 Autres méthodes	42

3.2 Comparatif	43
Conclusion	45
4 Implémentations et résultats	47
4.1 Méthode <i>Fully Convolutional Networks</i>	47
4.1.1 Normalisation des images	47
4.1.2 Entraînement	48
4.1.3 Résultats avec FCN-32s/FCN-16s/FCN-8s	50
4.1.4 Entraînement sur base non pré-traitée	50
4.1.5 Entraînement sur base pré-traitée	54
4.1.6 Comparaison des résultats	60
Conclusion	62
Conclusion et perspectives	63
Annexes	69
Annexe 1 : Le neurone biologique	69

Table des figures

1	Schéma théorique du rein (issue de www.cancer.be)	13
2	Image scanner et sa segmentation	15
3	Segmentation 3D	15
1.1	Perceptron	17
1.2	Les différentes fonctions d'activation	19
1.3	Perceptron multicouche	20
1.4	Le principe général de l'apprentissage	21
1.5	Principe de rétropropagation du gradient	23
2.1	Un CNN (<i>Convolution Neural Network</i>)	25
2.2	Schéma du fonctionnement du Neocognitron, issue de [1]	26
2.3	Le principe de la convolution	27
2.4	Le principe du <i>padding</i>	28
2.5	Une convolution et une déconvolution	28
2.6	Convolution transposée avec padding = 0 et stride > 1	29
2.7	Convolution transposée avec padding > 0 et stride > 1	30
2.8	Le principe du max pooling	30
2.9	Le principe du unpooling	31
2.10	L'architecture VGG16	33
2.11	Dropout, issue de [2]	33
3.1	<i>Fully Convolutional Network</i> , issue de [3]	35
3.2	FCN-32s, FCN-16s et FCN-8s	36
3.3	DeconvNet, issue de [4]	38
3.4	DecoupledNet, issue de [5]	39
3.5	SegNet, issue de [6]	40
3.6	U-Net, issue de [7]	42
4.1	Labellisation d'une segmentation	48
4.2	Le surapprentissage durant l'entraînement	48
4.3	Principe de la <i>k-fold cross validation</i>	49
4.4	Résultats des différents FCN	51

4.5	Courbes de l'erreur en fonction des itérations pendant l'entraînement sur les images non pré-traitées	52
4.6	Exemple du pré-traitement appliqué	54
4.7	Courbes de l'erreur (ou coût) en fonction des itérations pendant l'entraînement des images pré-traitées et non pré-traitées	55
4.8	Résultats des segmentations sur les images d'entraînement	57
4.9	Résultats des segmentations sur les images d'entraînement (suite)	58
4.10	Résultats des segmentations sur les images de test	59
4.11	Comparaison d'un résultat par Deep Learning avec un résultat par Croissance de région	60
4.12	Comparaison d'un résultat par Deep Learning avec un résultat par Watershed	61
4.13	Neurone biologique	69

Liste des tableaux

3.1	Comparaison des différents réseaux	43
3.2	Comparaison des résultats obtenus entre les différents réseaux	44
4.1	Résultats de la validation croisée sur les 3 sous-ensembles	50
4.2	Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images entraînées	52
4.3	Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images non entraînées	53
4.4	Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images pré-traitées et entraînées	55
4.5	Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images pré-traitées et non entraînées	56

Nomenclature

[BN] Batch Normalization

[CamVid] Cambridge-driving labeled Video Database

[CFI] Centre de Formation Informatique

[CHRU] Centre Hospitalier Régional Universitaire

[CNN] Convolutional Neural Network

[CRF] Conditional Random Field

[DeconvNet] Deconvolution Network

[DecoupledNet] Decoupled Network

[ENet] Efficient neural NETwork

[EPFL] École Polytechnique Fédérale de Lausanne

[FC] Fully Connected

[FCN] Fully Convolutional Networks

[ILSVRC] ImageNet Large Scale Visual Recognition Competition

[ReLU] Rectified Linear Unit

[SAIAD] Segmentation Automatique de reins tumoraux chez l'enfant par Intelligence Artificielle Distribuée

[VGG] Visual Geometry Group

Introduction

Contexte du travail : le projet SAIAD¹

Le néphroblastome est la tumeur du rein la plus fréquente chez l'enfant et survient généralement entre 1 et 5 ans. Il mène dans la plupart des cas à une ablation totale. Cependant, l'idéal serait de réaliser, lorsqu'il est possible, une néphrectomie partielle en épargnant ainsi la partie saine et fonctionnelle du rein touché.

Pour pouvoir déterminer le type d'opération à réaliser, le chirurgien doit être en mesure de visualiser l'emplacement précis de la tumeur grâce à des segmentations d'images scanners, seulement quelques heures avant l'opération. Néanmoins, ces segmentations étant manuelles et chronophages (il faut plusieurs heures pour obtenir les informations nécessaires à une décision) le chirurgien n'a généralement pas d'autres choix que de recourir à une néphrectomie totale du rein touché.

Voici, sur la figure 1, le schéma d'un rein qui permet d'identifier les différents éléments qui seront importants à segmenter dans les images.

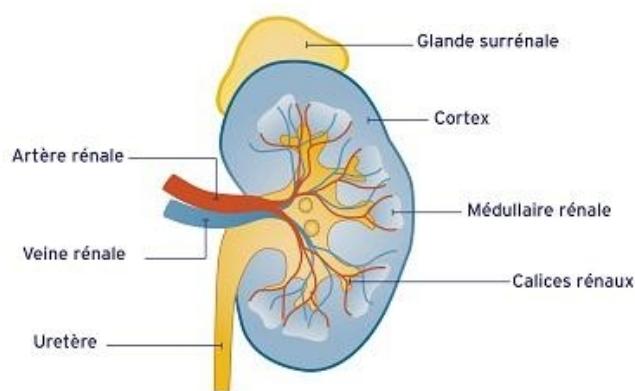


FIGURE 1 – Schéma théorique du rein (issue de www.cancer.be)

Mon travail s'intègre dans le projet SAIAD : un projet Interreg (financé par l'Europe) Franco-

1. Segmentation Automatique de reins tumoraux chez l'enfant par Intelligence Artificielle Distribuée

Suisse, d'une durée de 36 mois. L'institut FEMTO-ST est le leader de ce projet ayant comme partenaires le CHRU de Besançon, l'entreprise française Covalia IDO-In, l'École Polytechnique Fédérale de Lausanne (EPFL) ainsi que l'entreprise suisse CFI.

Chaque partenaire a un rôle bien défini dans ce projet :

- Le CHRU de Besançon devra fournir les segmentations d'images scanners de reins tumoraux,
- L'entreprise Covalia IDO-In devra créer une plateforme sécurisée de stockage et de visualisation des données des patients,
- L'EPFL sera en charge de l'état de l'art des techniques de segmentation manuelle,
- L'entreprise CFI sur la reconstruction automatique en 3D des segmentations, ainsi que sur la législation Suisse,
- Et enfin notre département de FEMTO-ST sera en charge de l'état de l'art et de l'implémentation des techniques des segmentations automatiques et distribuées.

Nous sommes trois à travailler sur l'état de l'art des segmentations automatiques. M. Florent MARIE, doctorant de Femto-ST, travaille sur l'ensemble des techniques de segmentation. M. Thibault DELAVELLE, stagiaire du master IPAC de Nancy (Interaction Perception Apprentissage Connaissance), travaille sur le raisonnement à partir de cas et enfin mes travaux portent sur le Deep Learning et son utilisation dans le cadre de la segmentation.

Plusieurs verrous scientifiques se présentent dans ce projet :

Actuellement, les segmentations de tumeurs rénales chez les enfants atteints de néphroblastome sont très longues à réaliser. Les enfants ne possèdent que très peu de tissus adipeux et sur les images scanner, les nuances de gris sont très proches si bien que la tumeur peut se confondre par exemple avec des muscles.

De plus, le projet a pour but de pouvoir segmenter les tumeurs de *tous* les enfants atteints de néphroblastome, ce qui couvre des enfants d'âges relativement différents : de 2 à 11ans. La taille des reins peut donc fortement varier d'un patient à un autre en fonction de l'âge. La figure 2 montre une coupe des reins ainsi que sa segmentation manuelle. Nous devons être capables de reproduire ces segmentations coupe par coupe pour reproduire la segmentation en 3D (cf figure 3).

Enfin la tumeur peut être de n'importe quelle forme (allongée, difforme, ...) et être placée à n'importe quel endroit sur le rein (collée au rein, plus invasive, le recouvrant,...).

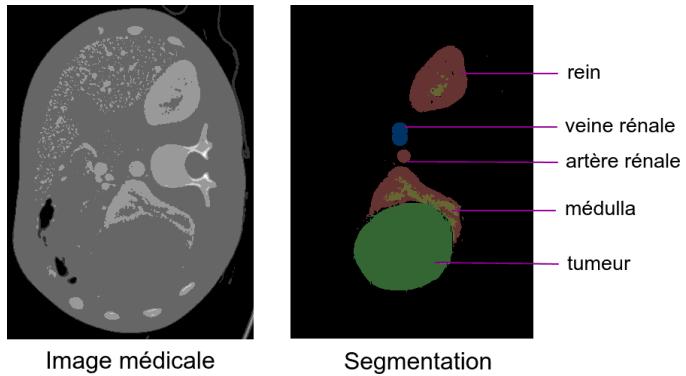


FIGURE 2 – Image scanner et sa segmentation

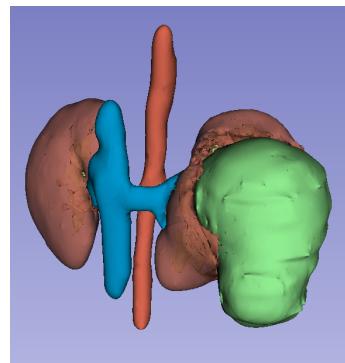


FIGURE 3 – Segmentation 3D

Le défis est de réaliser une segmentation automatique précise reproductible indépendamment de l'âge de l'enfant, de la taille de la tumeur ainsi que sa localisation aléatoirement sur un des deux reins.

Objectif de ces travaux de master recherche

Ces recherches doivent permettre au projet d'avoir une première approche sur une méthode particulière d'intelligence artificielle : le Deep Learning, technique obtenant des résultats de plus en plus prometteurs.

Les trois premiers chapitres de ce mémoire représentent le travail d'état de l'art que nous avons effectué.

Le premier chapitre expose les travaux issus de la littérature sur les réseaux de neurones. Nous présentons les perceptrons, et plus particulièrement le principe du Deep Learning, son apprentissage, ainsi que ses avantages et inconvénients.

Puis, le chapitre suivant est consacré à l'utilisation du Deep Learning dans des réseaux dits convolutifs, permettant entre autres la classification d'image. Nous verrons les différents principes de chaque étape du réseau ainsi qu'un exemple d'architecture basé sur ces réseaux convolutifs. Et nous terminons ce chapitre sur une discussion de l'utilisation de ces réseaux dans notre projet.

Enfin, nous présentons dans le chapitre suivant, les différentes techniques actuelles permettant de réaliser des segmentations automatiques ainsi que notre intérêt pour ces nouvelles méthodes.

Le dernier chapitre permet de décrire nos implémentations ainsi que les résultats d'une méthode de segmentation adaptée à la problématique du projet SAIAD, suivis d'une discussion sur ces résultats.

Chapitre 1

Les réseaux de neurones

1.1 Perceptron

Les réseaux de neurones artificiels ont pour objectif d'imiter la stimulation des neurones du cerveau humain, ils sont donc inspirés des neurones biologiques (cf Annexe 1). Le perceptron est le réseau de neurones artificiel le plus simple, inventé en 1957 par Franck Rosenblatt [8], représenté comme un système à N entrées et une sortie.

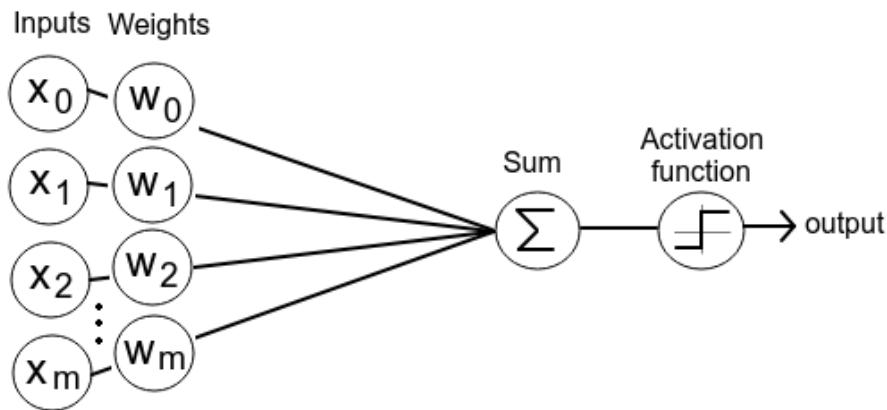


FIGURE 1.1 – Perceptron

Comme sur la figure 1.1, chaque entrée possède un poids w . La somme des entrées pondérées Σ est réalisée (cf équation 1) avant d'envoyer le résultat dans une fonction d'activation (cf équation 2).

Équation 1

Somme des entrées pondérées : $\sum_{i=1}^m x^i w_i$

La fonction d'activation ne renvoie généralement que deux valeurs, 1 si la somme des entrées pondérées est supérieure à un certain seuil, 0 sinon.

Équation 2

Fonction d'activation : $f(s) = f(\sum_{i=1}^m x^i w_i + b)$

On ajoute à la fonction d'activation un biais b qui correspond au seuil d'activation de la fonction. Cette fonction applique une non-linéarité au résultat, essentiel au fonctionnement du neurone, car la plupart des données en entrée sont non-linéaires.

Équation 3

Différents types de fonction d'activation peuvent être appliqués (cf figure 1.2) :

- La fonction Threshold : $\sigma(x) = \begin{cases} 1 & \text{if } \sum > threshold \\ 0 & \text{else} \end{cases}$
- La fonction Sigmoid : $\sigma(x) = \frac{1}{1+e^{-x}}$
- La fonction Tanh : $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- La fonction ReLU : $\sigma(x) = \max(0, x)$

Principalement, la fonction *ReLU* (cf équation 3) est utilisée comme fonction d'activation, car elle se trouve être la plus performante en particulier dans les réseaux profonds (qui seront au cœur de notre recherche et que nous présentons dans la section suivante). L'utilisation des autres fonctions peut engendrer des problèmes d'apprentissage, notamment le problème du *vanishing gradient*.

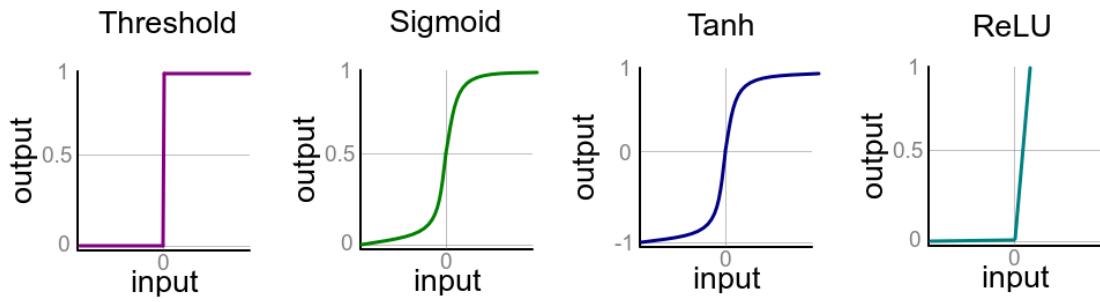


FIGURE 1.2 – Les différentes fonctions d'activation

Le problème du *vanishing gradient* est une difficulté rencontrée lors de l'apprentissage dans des réseaux de neurones profonds lors de la rétropropagation. Le gradient diminue exponentiellement à mesure de l'avancement dans le réseau et les dernières couches se retrouvent avec un gradient extrêmement faible. La sortie du réseau n'est alors que très peu modifiée et il ne peut pas s'entraîner efficacement.

Ceci est dû à la plupart des fonctions d'activation (Threshold, Sigmoid, Tanh) dans lesquelles les valeurs sont "écrasées" dans un intervalle relativement faible (0 ou 1 pour Threshold, [0,1] pour Sigmoid et [-1,1] pour Tanh) et dans lequel même une grande modification des paramètres aura finalement un très faible impact sur la sortie.

Pour éviter ce problème, nous pouvons utiliser la fonction ReLU qui ne restreint pas les valeurs positives dans un ensemble fini et supprime simplement les valeurs négatives.

1.2 Le perceptron multicouche

Le perceptron multicouche a été inventé dans les années 1980. Il s'agit d'une succession de couches contenant plusieurs neurones, chaque neurone d'une couche étant relié à la totalité des neurones des couches adjacentes (cf figure 1.3).

Un perceptron multicouche possède une couche d'entrée, une couche de sortie ainsi qu'une ou plusieurs couches cachées. Chaque résultat des fonctions d'activations des neurones à la couche $N - 1$ sera alors transmis à tous les neurones de la couche N . Chaque neurone de la couche N effectue la somme de toutes leurs entrées pondérées, transmet ce résultat à la fonction d'activation et envoie la sortie à tous les neurones de la couche $N + 1$, et ainsi de suite jusqu'à la dernière couche.

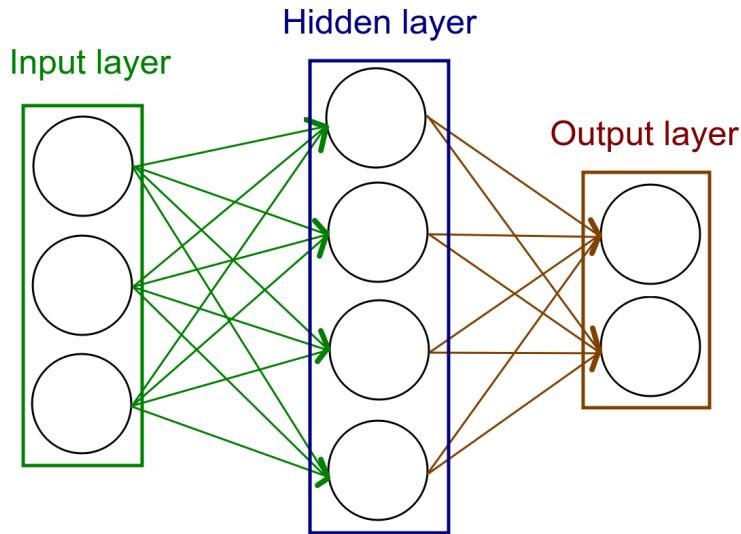


FIGURE 1.3 – Perceptron multicouche

1.3 Le Deep Learning

1.3.1 Le principe

Le concept de Deep Learning ou apprentissage profond ne se concrétise qu'au début des années 2010. Il s'agit d'une manière particulière de réaliser du Machine Learning.

Le Machine Learning, ou apprentissage automatique est l'étude d'algorithmes permettant d'apprendre et d'évoluer seul en étudiant des exemples, c'est-à-dire en s'entraînant. Le Deep Learning consiste à réaliser un réseau de neurones profond, sur le même modèle que les perceptrons multicouches, mais avec de nombreuses couches cachées (plus de deux couches cachées). Ce réseau est alors capable d'extraire des abstractions complexes de haut niveau des données lui étant présentées.

Il s'agit d'un type d'apprentissage supervisé, c'est-à-dire que le réseau a besoin d'une base d'apprentissage pour s'entraîner et ainsi trouver une solution pour obtenir le résultat attendu.

1.3.2 Le processus d'apprentissage

Le principe général de l'apprentissage

Les réseaux de neurones ont besoin dans un premier temps d'une phase d'apprentissage durant laquelle les poids et les biais des neurones sont ajustés jusqu'à ce que la sortie prédite soit

similaire à la sortie désirée.

Comme le montre la figure 1.4, les entrées sont transmises au système, c'est-à-dire au réseau de neurones, permettant ainsi de calculer une valeur de sortie. La sortie prédictive sera alors comparée à la sortie désirée grâce à une fonction de coût. En fonction de la différence entre la sortie calculée et la sortie attendue, les paramètres seront ajustés avant de repasser dans le réseau pour un nouveau calcul, et ainsi de suite jusqu'à ce que la valeur calculée et la valeur désirée soient similaires.

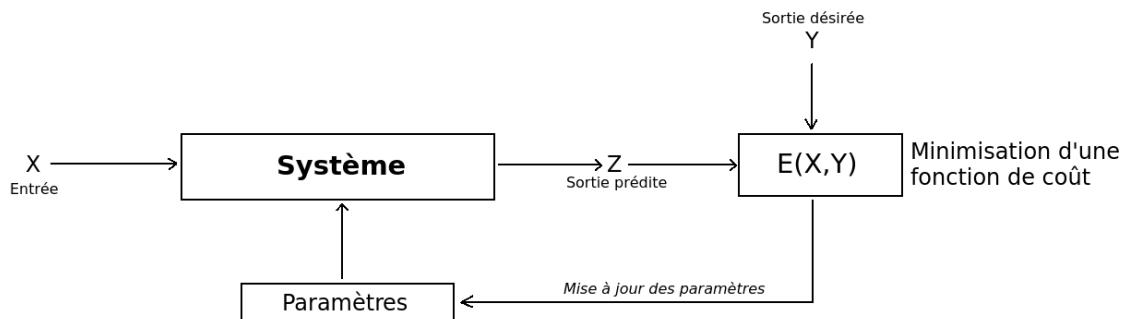


FIGURE 1.4 – Le principe général de l'apprentissage

La technique de rétropropagation du gradient (ou backpropagation) est utilisée afin d'ajuster correctement les paramètres lors de la phase d'apprentissage. Elle s'appuie sur la technique de descente du gradient.

La descente du gradient

Le gradient est un vecteur de pente représentant la variation d'une fonction par rapport à la variation de ses paramètres. Il existe un algorithme d'optimisation du gradient qui permet de converger de manière itérative vers une configuration optimisée. Il se peut toutefois que la configuration optimale puisse être un minimum local et non un minimum global.

Au début de l'entraînement, les poids sont initialisés aléatoirement. À chaque fin d'itération, l'algorithme calcule un indicateur de sa performance en calculant une fonction de coût. Il existe différentes fonctions de coût, mais l'une des plus utilisées est la fonction *multinomial logistic loss* (cf équation 4).

Équation 4

Soit E la fonction de coût, N le nombre d'images à segmenter, p'_n la sortie calculée pour l'image n et l_n la sortie désirée pour la classe n , E est définie :

$$E = \frac{-1}{N} \sum_{n=1}^N \log(p'_n, l_n)$$

Les paramètres sont ensuite mis à jour en utilisant la technique de descente du gradient (cf équation 5).

Équation 5

Soit $\Delta w_{i,j}$ le déplacement à appliquer au poids reliant les neurones i et j , α le pas d'apprentissage et $\frac{\delta E}{\delta w_{i,j}}$ la dérivée de la fonction coût par rapport au poids de i à j , $\Delta w_{i,j}$ est défini :

$$\Delta w_{i,j} = -\alpha \frac{\delta E}{\delta w_{i,j}}$$

À chaque itération, le poids $w_{i,j}$ est incrémenté de $\Delta w_{i,j}$. Tous les paramètres à modifier sont alors ajustés en fonction de leur gradient. À noter qu'un gradient faible aura un léger impact sur la modification des paramètres alors qu'un gradient important aura un plus grand impact sur le changement des valeurs, au risque même de dépasser le minimum local ou global.

La rétropropagation du gradient

Dans les réseaux de neurones multicouches, le principe de descente du gradient ne suffit plus, car il faut désormais pouvoir se déplacer dans les différentes couches du réseau pour ajuster les paramètres. La technique de rétropropagation du gradient est alors appliquée dans le cas du Deep Learning, technique prenant de l'engouement en 1986 grâce à David Rumelhart et al. [9].

Comme l'indique la figure 1.5, le principe est le suivant :

Tout d'abord, le système initialise les paramètres de chaque couche de façon aléatoire à travers l'ensemble du réseau. Une fois l'initialisation effectuée, l'indicateur de performance avec la

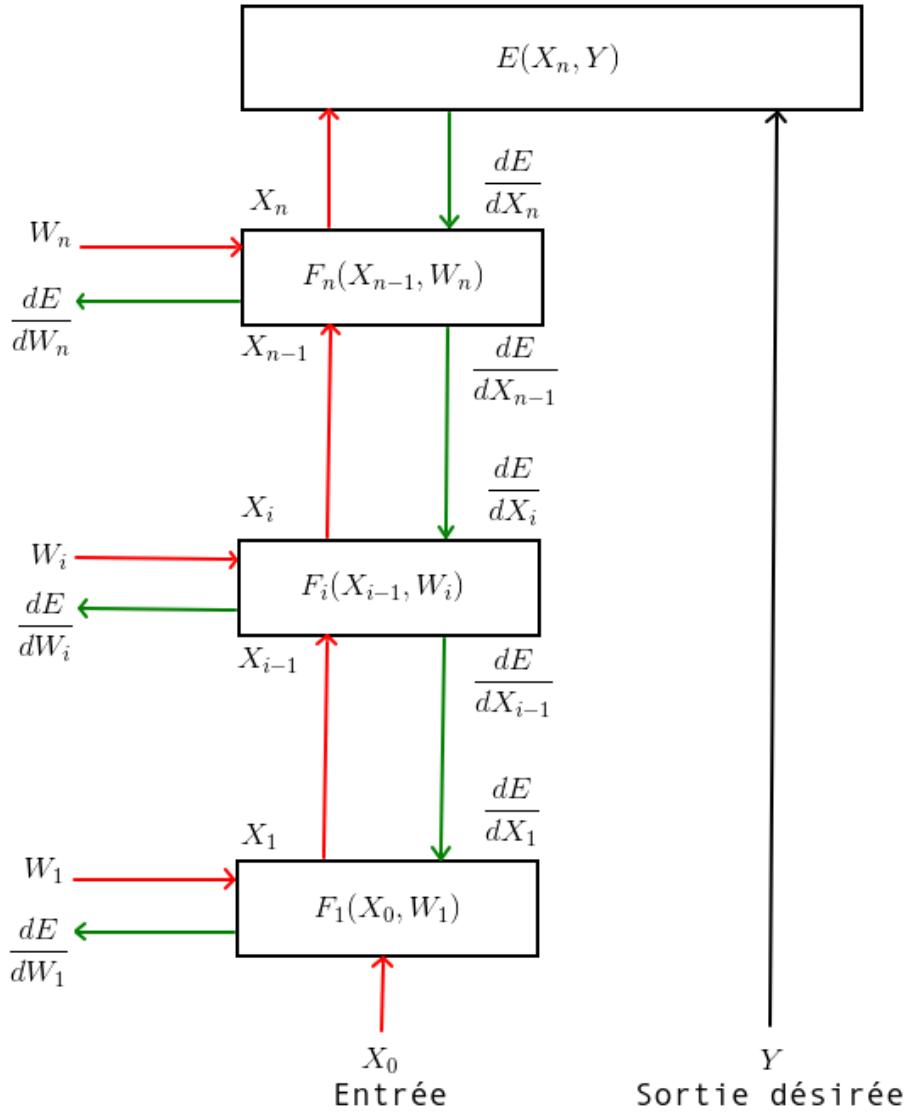


FIGURE 1.5 – Principe de rétropropagation du gradient

fonction de coût est calculé comme pour la descente du gradient. Le système remonte ensuite dans le réseau en calculant la dérivée de la fonction de coût par rapport à la valeur du neurone trouvé précédemment pour passer à la couche précédente ($\frac{\delta E}{\delta X_n}$ avec X la sortie du neurone de la couche n). Une fois aux couches précédentes, le système calcule le gradient de chaque poids et biais et ajuste leur valeur, également de la même façon que pour la technique de descente du gradient. Arrivé à l'entrée du réseau, il itère alors à nouveau pour obtenir une sortie prédictive plus affinée grâce aux paramètres précédemment ajustés, le but étant d'itérer jusqu'à ce que l'indicateur de performance soit satisfaisant.

Conclusion

Le Deep Learning est une méthode de plus en plus utilisée depuis les années 2010 et les recherches sur ce domaine ne cessent d'augmenter. Pour cause, l'un des principaux avantages du Deep Learning est l'analyse et l'apprentissage de quantités massives de données. De nombreuses grandes entreprises comme Google, Facebook ou Microsoft l'utilisent en particulier pour le Big Data et plus particulière pour le Data Mining.

Le Deep Learning est un véritable atout dans le traitement d'image permettant ainsi la classification d'image, la détection d'objets et même de la segmentation. Il peut donc être capable d'apprendre par lui-même la réalisation de segmentations précises avec un bon entraînement.

Il s'agit donc d'une technique très intéressante et qui semble particulièrement adaptée pour le projet SAIAD. Bien entraîné, notre réseau peut apprendre à détecter et segmenter la tumeur, même si elle se confond avec les muscles de l'enfant, car le réseau apprendra de lui-même la légère différence entre la tumeur et les muscles. Quel que soit l'avancement de la tumeur, sa localisation (rein gauche ou droit) et sa forme, le réseau sera également capable d'apprendre de toute ces différences et les principales problématiques du projet peuvent alors être écartées.

Toutefois, le Deep Learning a besoin d'une base de cas assez conséquente et hétérogène pour l'apprentissage, car chaque cas de tumeur rénale chez un patient est différent. Cela peut être problématique pour la reproductibilité de la segmentation, car la tumeur possède une forme aléatoire et peut déplacer et déformer les organes.

Un autre des principaux inconvénients se trouve dans le nombre limité de segmentations de reins tumoraux dont nous disposons dans le projet. Il est en effet difficile d'obtenir une centaine de cas pour des raisons à la fois médico-légales et de temps. L'apprentissage peut également être très long, dû au nombre important de données et le Deep Learning a besoin de carte graphique performante ainsi que de beaucoup de ressource.

Chapitre 2

Les réseaux de neurones convolutifs

2.1 Le principe

Un réseau de neurones convolutif ou CNN (*Convolution Neural Network*) est une méthode d'apprentissage automatique basée sur le Deep Learning. Le problème est divisé en sous-parties et chaque partie sera étudiée pour en déterminer les caractéristiques. On utilise les CNN pour le traitement d'image, le traitement de vidéo ainsi que le traitement du langage.

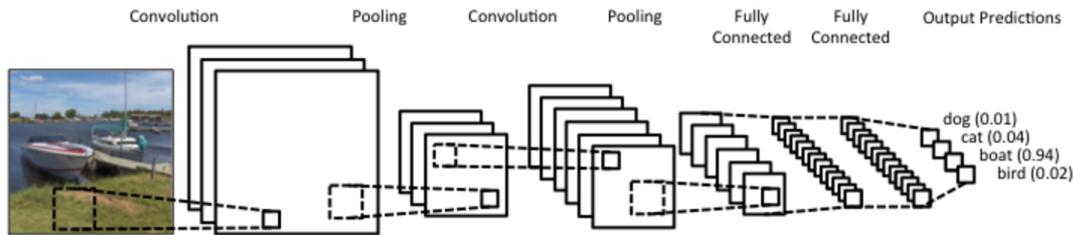


FIGURE 2.1 – Un CNN (*Convolution Neural Network*)

Le Neocognitron est le précurseur des CNN et a été introduit en 1980 par Kunihiko Fukushima et al. [1], mais la référence standard des CNN vient de Yann LeCun et al. en 1998 [10]. Les auteurs proposent alors un nouveau réseau prénommé LeNet-5 permettant de reconnaître les chiffres et est appliqué à la reconnaissance de nombres manuscrits sur les chèques.

Avec le Neocognitron le système se focalise sur toutes les formes de l'image pour permettre la reconnaissance de celle-ci. Chacune de ses couches est divisée en deux, les sous-couches simples Us ainsi que les sous-couches complexes Uc. La sous-couche simple acquiert de l'information de la sous-couche complexe précédente, tandis que la sous-couche complexe généralise l'information de la sous-couche simple. Chaque cellule de la sous-couche simple référence une des formes de l'image qui sera généralisée dans une des cellules de la sous-couche complexe (cf figure 2.2).

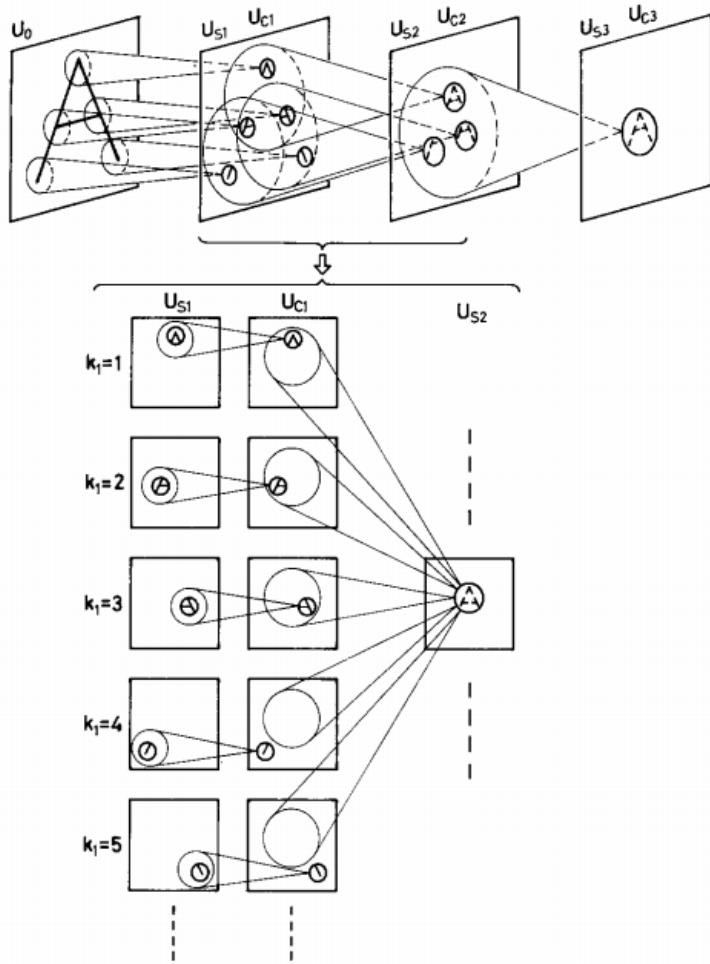


FIGURE 2.2 – Schéma du fonctionnement du Neocognitron, issue de [1]

Le CNN d'aujourd'hui possède différents types de couches qui permettent de modifier et de moduler l'information afin de parvenir au résultat final. Les différentes couches présentées par la suite sont plus généralement utilisées dans le cas de traitement d'image, comme la classification d'image, dans la détection et la segmentation d'objet.

La figure 2.1 représente un CNN pour la classification d'images. Différentes couches sont utilisées à la suite pour modifier et rétrécir l'image, jusqu'à l'obtention d'un pourcentage d'appartenance aux classes du réseau.

2.2 Les différentes couches

2.2.1 La couche de Convolution

Le principe des couches de convolution est d'appliquer un filtre à l'image dans le but de la modifier.

La convolution

Dans le cas d'une convolution, le filtre (ou le noyau) est appliqué à l'image en multipliant chacun de ses pixels par celui du filtre correspondant. Puis ces multiplications sont additionnées afin d'obtenir un des pixels finaux de l'image résultante. Le filtre est alors déplacé sur toute l'image afin de calculer tous les pixels en sortie (cf figure 2.3).

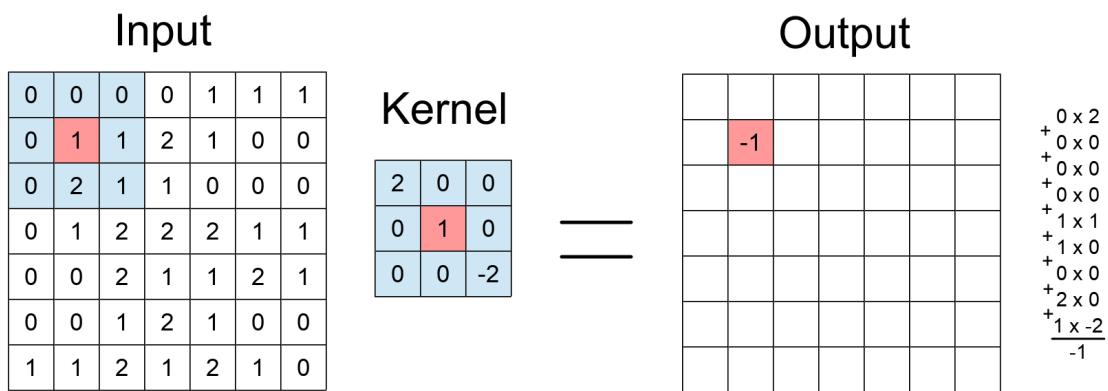


FIGURE 2.3 – Le principe de la convolution

Les filtres sont le plus généralement de taille 2x2, 3x3 ou 5x5, car des filtres supérieurs apportent une modification beaucoup trop importante de l'image.

Comme le montre la figure 2.3, la taille de l'image en sortie est réduite par rapport à l'image en entrée, parce que le filtre ne peut s'appliquer aux bordures de l'image.

L'opération nommée *padding* (cf figure 2.4) consistant à ajouter des pixels de valeurs 0 tout autour de l'image peut être réalisée, permettant ainsi au filtre de réaliser une convolution sur l'ensemble de l'image et ainsi conserver les mêmes dimensions.

La notion nommée *stride* est également appliquée. Il s'agit du pas appliqué à notre filtre pour son déplacement sur l'image. Le pas est aussi à prendre en compte sur la taille de l'image en sortie, car un grand pas aura pour effet de diminuer la taille et à l'inverse un pas très faible l'augmentera.

0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0
0	0	1	1	2	1	0	0	0
0	0	2	1	1	0	0	0	0
0	0	1	2	2	2	1	1	0
0	0	0	2	1	1	2	1	0
0	0	0	1	2	1	0	0	0
0	1	1	2	1	2	1	0	0
0	0	0	0	0	0	0	0	0

FIGURE 2.4 – Le principe du *padding*

La déconvolution

La déconvolution (ou appelée également *convolution transposée*) est un procédé qui consiste à inverser les effets de la convolution. Le principe est d'appliquer un filtre à l'image de sortie résultante de la convolution pour obtenir une image similaire à l'image d'entrée avant la convolution. Le filtre de la déconvolution doit être de la même taille que le filtre de la convolution.

Si la dimension de l'image en sortie de la convolution est différente de la dimension de l'image d'origine, il est possible alors d'ajuster le *padding* et le *stride* lors de la déconvolution pour retrouver la dimension initiale.

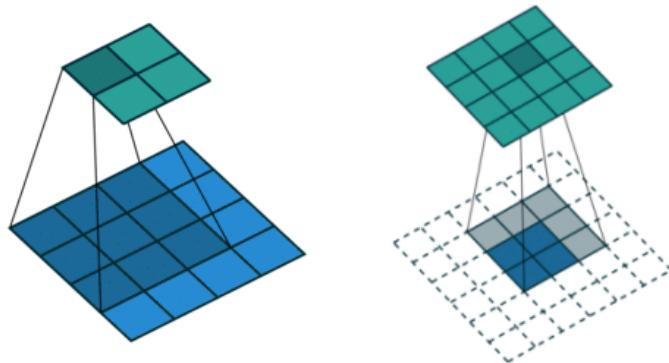


FIGURE 2.5 – Une convolution et une déconvolution

La figure 2.5 présente tout d'abord une convolution d'une image 4×4 avec un filtre 3×3 ce qui donne une image 2×2 . La déconvolution de cette convolution peut être un ajout d'un *padding* de 2 à l'image 2×2 , ainsi qu'un *stride* de 1.

Cette déconvolution est valable dans le cas où le *stride* lors de la convolution est égale à 1. Il existe également d'autres méthodes pour la déconvolution dans lesquelles la convolution possède un *stride* supérieur à 1 et le pas appliqué est tellement faible que des pixels de valeurs 0 sont ajoutés entre les pixels de l'image.

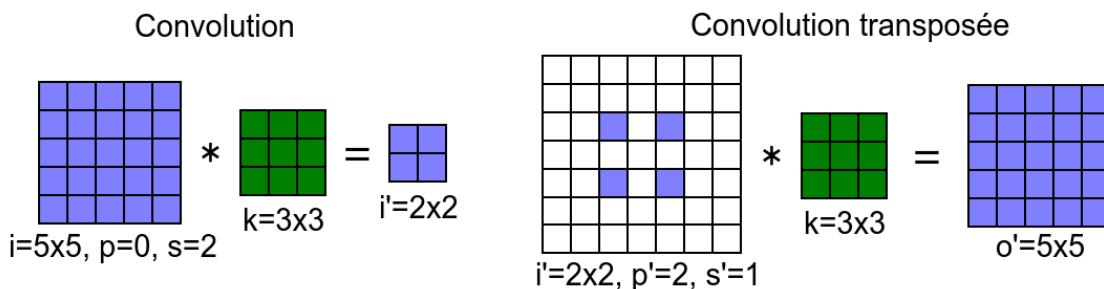


FIGURE 2.6 – Convolution transposée avec padding = 0 et stride > 1

Voici deux exemples courants qui permettent de bien comprendre les principes de convolution transposée :

- Premier cas (cf figure 2.6) : la convolution est effectuée sans padding et avec un stride supérieur à 1. Dans ce cas, les paramètres de la déconvolution sont calculés à l'aide de la formule 6 suivante :

Équation 6

Soit k et k' le noyau de la convolution et de la déconvolution,
 p et p' le padding de la déconvolution,
 s et s' les strides respectivement de la convolution et de la déconvolution,
 o' la taille de l'image après la déconvolution,
 i' la taille de l'image avant la déconvolution, alors :

$$k' = k, \quad p' = k - 1, \quad s' = 1 \quad \text{et} \quad o' = s(i' - 1) + k$$

- Deuxième cas (cf figure 2.7) : la convolution est effectuée avec un padding et un stride supérieur à 1. Dans ce cas, les paramètres de la déconvolution sont calculés à l'aide de la formule 7 suivante :

Équation 7

$$k' = k, \quad p' = k - p - 1, \quad s' = 1 \quad \text{et} \quad o' = s(i' - 1) + k - 2p$$

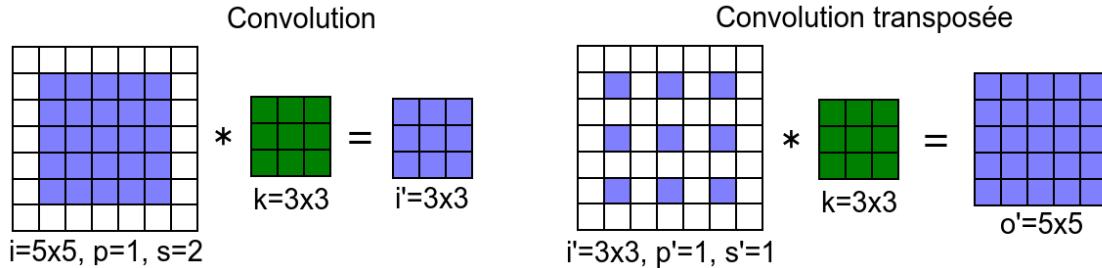


FIGURE 2.7 – Convolution transposée avec padding > 0 et stride > 1

2.2.2 La couche de pooling

Le pooling

Le pooling est un type de sous-échantillonnage de l'image. L'image d'entrée est découpée en plusieurs régions et chaque région sera désignée par une seule valeur. Il existe différents types de pooling, le max pooling qui consiste à choisir le pixel de plus grande valeur de chaque région (cf figure 2.8), ou encore l'Average pooling qui calcule la moyenne de tous les pixels de chaque région pour en déterminer le pixel final.

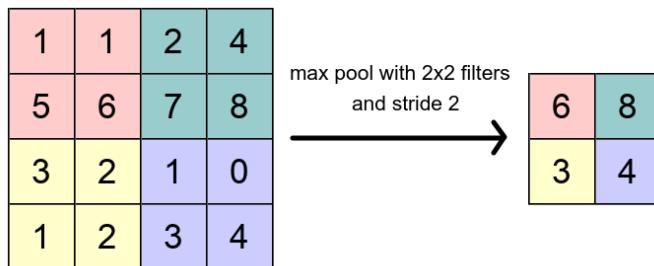


FIGURE 2.8 – Le principe du max pooling

La couche de pooling est très importante dans un réseau, car elle permet de réduire la quantité de paramètres due à la diminution de la taille de l'image, et donc de réduire le nombre de calculs dans le réseau.

Elle crée également une forme d'invariance par translation, c'est-à-dire que l'information visuelle sera traitée de la même façon où qu'elle soit.

À noter que la valeur du pas du pooling doit toujours être la même que la taille de son filtre pour que chaque valeur ne soit que dans une seule région. Il peut, dans ce cas, y avoir une perte de données si la hauteur et la largeur de l'image ne sont pas divisibles par la taille du filtre.

Le unpooling

L'opération de unpooling est l'inverse du pooling. L'étape du pooling n'est pas inversible, mais la transposée du pooling peut être approximée en sauvegardant les positions des pixels désignés par chaque région pendant le pooling, et en utilisant ces positions pour reconstruire l'image lors du unpooling (cf figure 2.9). Les pixels sont repositionnés au bon endroit grâce aux positions sauvegardées, et chaque région est remplie par des pixels de valeur zéro.

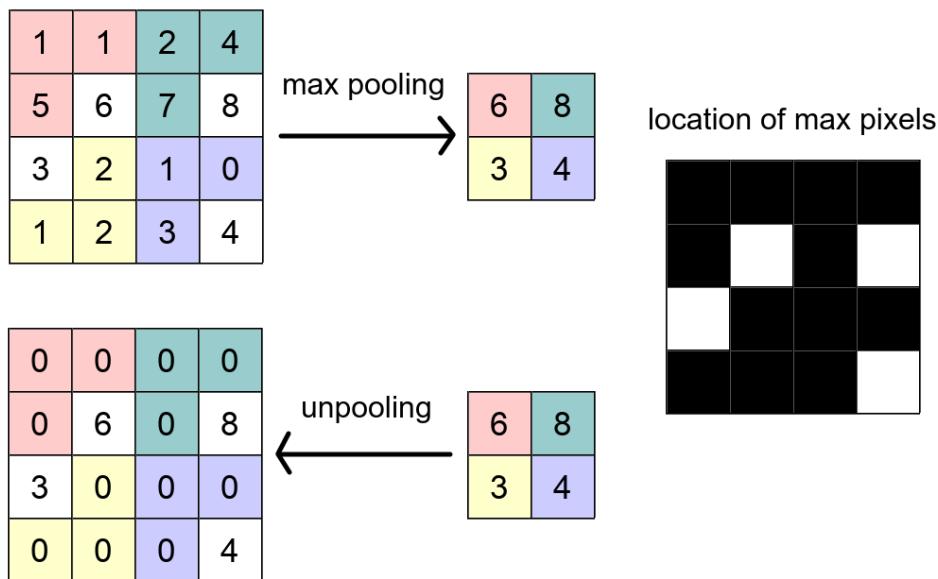


FIGURE 2.9 – Le principe du unpooling

2.2.3 La couche de correction

Il est possible d'améliorer l'efficacité du traitement en ajoutant après chaque couche de convolution une couche de correction sur l'image, dans laquelle la valeur de chaque pixel sera modifiée. La couche de correction ajoute une non-linéarité au réseau, indispensable, car les convolutions

et les déconvolutions sont des opérations linéaires. La fonction ReLU est bien entendu la fonction la plus utilisée, étant la plus performante [11]. Elle permet également de remplacer toutes les valeurs négatives d'une image par zéro et d'ajouter cette non-linéarité.

2.2.4 La couche *Fully Connected*

Les neurones des couches *Fully Connected* où FC sont connectés à la totalité des neurones des couches adjacentes. L'image n'est plus représentée sous la forme d'une matrice, mais d'un vecteur, dans lequel chaque valeur correspond à un neurone.

Dans le cas d'une classification d'image, le but des couches FC est de réduire le vecteur jusqu'à l'obtention d'un vecteur de la taille du nombre de classes que le réseau peut classifier.

2.2.5 La couche de perte

La couche de perte est normalement la dernière couche dans le réseau pour la classification d'image. Elle applique une fonction SoftMax (cf équation 8) à chaque valeur du vecteur dans le but de calculer le pourcentage d'appartenance à une classe.

Équation 8

$$\text{Fonction SoftMax : } \sigma(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

2.3 Exemple d'architecture de classification d'images basée sur les CNN

L'architecture VGG16 [12], basée sur les CNN, est l'une des architectures les plus utilisées pour la classification d'images. Cette architecture a gagné la deuxième place du concours ILSVRC 2014¹ derrière GoogLeNet [13], avec néanmoins un nombre de couches relativement plus faible que ce dernier : ce qui est remarquable compte tenu de la qualité des résultats obtenus.

Elle est composée de 13 couches de convolution, 5 couches de pooling ainsi que de 3 couches FC et une couche SoftMax pour le calcul d'appartenance aux classes.

1. ILSVRC est un concours de traitement d'images dans la détection d'objet et la classification d'images, se déroulant chaque année et fournissant une base de données d'images à utiliser pour le concours.

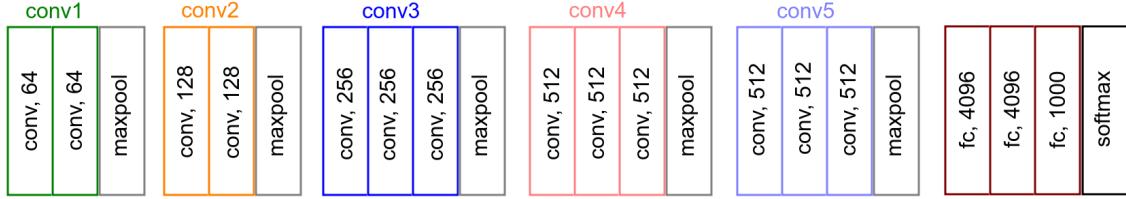


FIGURE 2.10 – L’architecture VGG16

Les blocs conv1 et conv2 de la figure 2.10 ne contiennent que deux couches de convolution par rapport aux blocs conv3, conv4 et conv5 qui en contiennent 3. Les numéros dans les couches de convolutions sont les nombres d’images résultantes en sortie des couches de convolution, où chaque image est issue d’un filtre différent. On trouve également des couches ReLU après chaque couche de convolution et 1000 valeurs sont produites à la sortie de la dernière couche FC, car l’architecture est conçue pour la base d’images du concours ILSVRC contenant 1000 classes.

Le nombre de neurones dans les couches est important et le réseau peut alors avoir un apprentissage lent. Pour pallier cet inconvénient, des couches de Dropout ont été ajoutées entre les couches FC.

2.3.1 Le principe du dropout

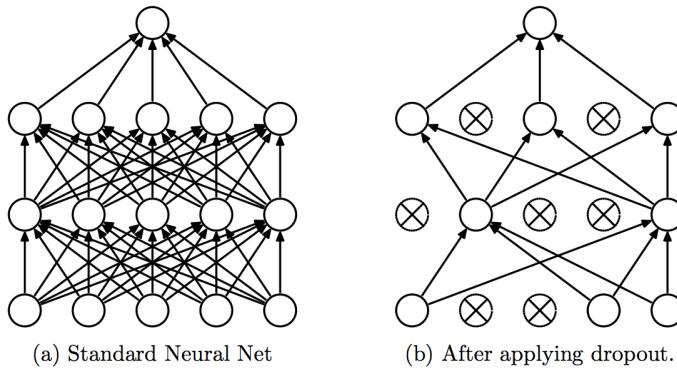


FIGURE 2.11 – Dropout, issue de [2]

L’idée du dropout [2] est de débrancher des neurones aléatoirement lors de l’apprentissage, à chaque itération, pour régulariser le nombre de neurones dans le réseau et augmenter la performance des réseaux lors de l’apprentissage (cf figure 2.11).

Conclusion

Les CNN sont donc très utilisés en classification d'images et montrent des résultats intéressants. Avec un tel réseau appliqué à notre projet, nous pouvons facilement développer un système permettant de reconnaître l'existence d'une tumeur ainsi que d'organes déformés et déplacés par celle-ci sur les images médicales.

Cependant, nous voulons savoir où se trouvent précisément la tumeur ainsi que les différents autres éléments présents sur l'image grâce à des segmentations. Heureusement, les CNN servent souvent de base à d'autres réseaux de neurones permettant d'effectuer des choses plus poussées, comme la segmentation d'objet.

Chapitre 3

Les segmentations avec Deep Learning

3.1 Les différentes méthodes de segmentation

3.1.1 *Fully Convolutional Networks*

Les premiers réseaux de neurones spécifiques pour la segmentation d'image apparaissent en 2015 avec la méthode FCN ou *Fully Convolutional Networks* [3]. Ce nouveau réseau de neurones profond est un CNN normal, dans lequel les couches FC ont été remplacées par des couches de convolution de taille 1x1. La dernière couche est une couche de sur-échantillonnage, dans le cas du FCN une déconvolution, dans laquelle l'image en résultant est de la même taille que l'image d'entrée (cf figure 3.1).

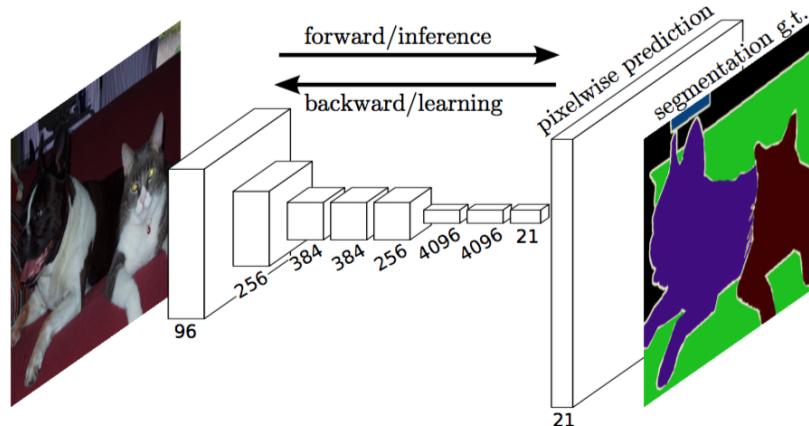


FIGURE 3.1 – *Fully Convolutional Network*, issue de [3]

Ce concept de segmentation a été réalisé sur trois types d'architectures très efficaces pour la classification d'objet : AlexNet [14], VGG16 et GoogLeNet [13]. Il s'est avéré que l'implémentation du FCN sur l'architecture VGG16 a rendu les meilleurs résultats avec la base de cas

PASCAL VOC¹ 2011 et 2012 [15], bien que l’entraînement soit plus long, possédant plus de paramètres que les deux autres architectures. Néanmoins, c’est vers cette architecture que les auteurs ainsi que toutes les autres utilisations de FCN se basent par la suite.

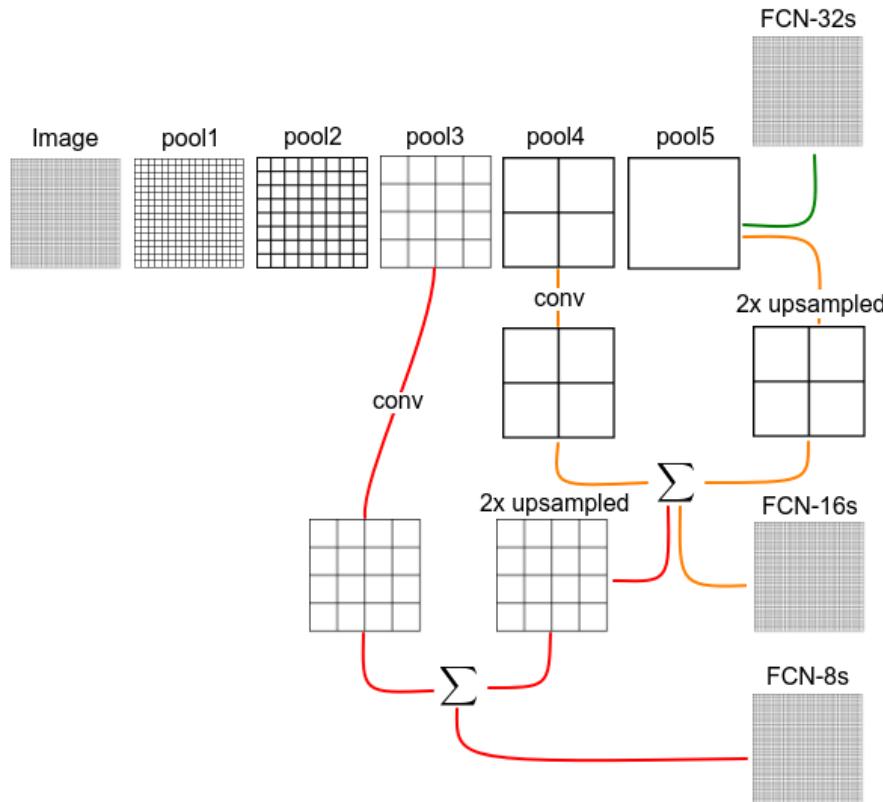


FIGURE 3.2 – FCN-32s, FCN-16s et FCN-8s

La déconvolution effectuée dans FCN est une déconvolution conforme à celle de la figure 2.6 avec un stride de 32 et un noyau de 64, d'où son nom de FCN-32s. Il est possible de rendre les segmentations encore plus précises grâce à FCN-16s et FCN-8s. Pour ces réseaux, des sauts sont alors ajoutés et ils combinent la couche de prédiction finale avec des couches inférieures plus fines. Ces couches inférieures génèrent des segmentations plus nettes, mais moins précises. La combinaison de ces couches avec la couche finale permet alors de réaliser des prédictions nettes et précises.

La figure 3.2 nous montre comment obtenir les trois types de réseaux FCN. Pour obtenir FCN-16s, le réseau FCN-32s est modifié en divisant la sortie en deux au niveau du pool4 (pooling

1. PASCAL VOC (Visual Object Classes Challenge) est un concours de traitement d’images dans plusieurs domaines (classification, détection, segmentation) fournissant une base de données d’images open source à utiliser pour le concours.

numéro 4 dans le réseau). Il est ajouté une convolution de taille 1x1 à l'une des sorties du pool4 et une déconvolution de 2 (un stride de 2) à la sortie du pool5. Les deux résultats sont alors additionnés et est ajoutée au résultat une déconvolution avec un stride de 16 et un noyau de 32. Ce réseau permet de nous donner de meilleurs résultats, mais il peut être encore amélioré en FCN-8s où en plus du FCN-16s, le réseau est séparé également en deux à la sortie du pool3. Une convolution 1x1 est alors ajoutée à l'une des sorties du pool3 et une déconvolution de 2 est ajoutée à la sortie de l'addition du FCN-16s. L'addition de ces deux images et la réalisation d'une déconvolution avec un stride de 8 et un noyau de 16 nous donnent notre réseau FCN-8s.

Une base d'apprentissage est nécessaire pour l'entraînement du réseau. Dans le cas des segmentations, il est nécessaire de fournir les images d'origines ainsi que les images segmentées correspondantes pour que le réseau puisse apprendre des résultats attendus.

FCN dans le milieu médical

Après le succès de FCN, Austin Ray de l'Université de Stanford a mené un projet sur l'utilisation de cette méthode pour la segmentation de tumeur pulmonaire [16]. Possédant des segmentations sur 107 patients, l'auteur a diminué le réseau en enlevant les dernières couches de convolution pour obtenir un réseau moins lourd et a obtenu de bons résultats en 7h d'entraînement en utilisant l'indicateur de statistique Dice avec une moyenne de 0.86 entre les segmentations réelles et les segmentations prédites.

Toutefois, les tumeurs étant toutes très diverses et variées, son logiciel de segmentation n'est pas réellement fiable pour tous les cas de figure.

L'indicateur statistique Dice (cf équation 9), compris entre 0 et 1, est utilisé principalement dans le domaine médical pour mesurer le degré de similarité de deux échantillons

Équation 9

$$S = \frac{2 \times |X \cap Y|}{|X| + |Y|}$$

avec S la valeur du Dice, X la segmentation réelle et Y la segmentation calculée.

Un Dice est alors calculé pour chaque classe du réseau, et la moyenne de ces Dices permet d'obtenir un Dice global.

3.1.2 Deconvolution Network et Decoupled Network

Deconvolution Network

En 2015, après la création de FCN, une autre méthode est apparue : DeconvNet (*Deconvolution Network*) ou "l'extrême segmentation" [4]. Son principe est de réaliser un réseau de neurones convolutif basé sur l'architecture VGG16 et d'y ajouter un réseau de neurones déconvolutif, où chaque convolution correspond à une déconvolution et où chaque pooling à un unpooling. Ce type de réseau permet de "revenir en arrière" sur les modifications effectuées et le réseau de neurones déconvolutif peut être vu comme le miroir du réseau convolutif (cf figure 3.3).

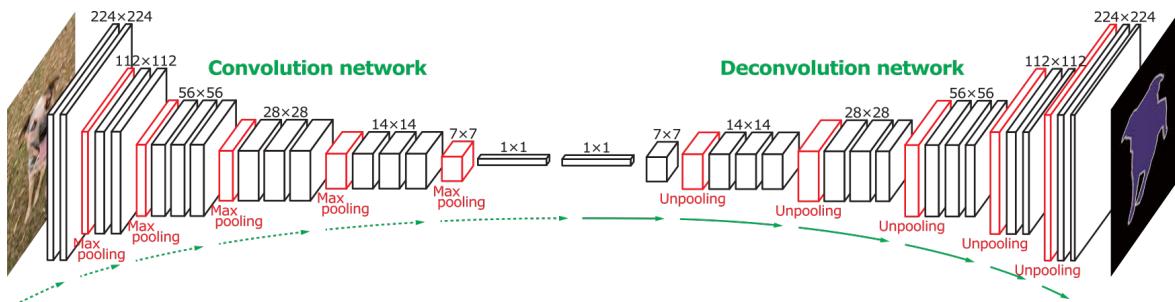


FIGURE 3.3 – DeconvNet, issue de [4]

Des couches ReLU sont placées après chaque convolution et déconvolution, mais le réseau possède également à la suite des ReLU des couches de *Batch Normalization* [17] ou BN, qui permettent de réduire le problème du *internal-covariate-shift*. Dans les réseaux de neurones profonds, une petite perturbation dans les couches initiales entraîne un changement important dans les couches ultérieures. Les valeurs des couches suivantes peuvent être totalement aberrantes et leur gradient essaie de compenser ces valeurs. Les BN permettent alors de normaliser les valeurs et l'apprentissage est optimisé et plus rapide.

Pour obtenir de meilleur résultat, le réseau EDeconvNet est également créé. Il combine les résultats de DeconvNet et FCN-8s en moyennant leur image de sortie pour obtenir la segmentation finale. Il peut être réalisé à la sortie du réseau un *Conditional Random Field*² ou CRF [18] sur l'image calculée pour obtenir une segmentation encore plus précise.

L'entraînement de DeconvNet diffère d'un entraînement classique et il s'effectue en deux étapes :

La première est l'entraînement du réseau sur des exemples simples, pour entraîner grossièrement le réseau, et la deuxième est l'affinement de celui-ci en envoyant les images dont les

2. Les CRF sont des modèles statistiques utilisés pour la reconnaissance de forme.

segmentations sont plus complexes. Testé sur la base de données PASCAL VOC 2012, le réseau a obtenu de bons résultats, dépassant ceux de FCN.

Decoupled Network

En continuation des travaux sur DeconvNet, les mêmes auteurs ont réalisé le réseau *Decoupled Network* [5] ou DecoupledNet en 2015. Il est découpé en trois parties : un réseau VGG16 pour la classification d'image, un réseau déconvolutif, le même que celui de DeconvNet pour la segmentation et les couches intermédiaires appelées *bridges* ou ponts, permettant de relier le réseau de classification à celui de segmentation (cf figure 3.4).

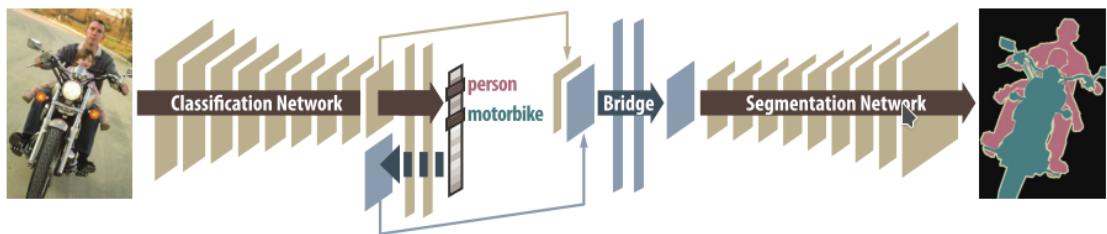


FIGURE 3.4 – DecoupledNet, issue de [5]

Les différentes classes de l'image d'entrée sont identifiées par le réseau de classification et la segmentation est ensuite effectuée sur chacun des objets identifiés sur l'image. Les couches intermédiaires fournissent les informations spécifiques dont a besoin le réseau de segmentation, c'est-à-dire le tableau de localisation de chaque classe détectée.

Plus précisément, les couches intermédiaires récupèrent les tableaux des objets en se basant des résultats de la couche du pool5 du réseau de classification, qui conserve l'information spatiale des objets. Une fois arrivé à la fin du réseau de classification, il est réalisé une *saliency map*³ en effectuant une rétropropagation spécifique [19] jusqu'au pool5. Les tableaux obtenus sont alors combinés aux tableaux résultant du pool5 et les tableaux de localisation de chaque classe sont envoyés dans le réseau de segmentation.

Le fait de réaliser séparément plusieurs segmentations sur les classes trouvées réduit l'espace de recherche sur l'image d'origine ainsi que le nombre de paramètres dans le réseau de segmentation.

3. Une saliency map permet de simplifier et modifier la représentation d'une image en quelque chose de plus significatif comme un pixel à un niveau de gris élevé. La qualité de chaque pixel apparaît alors dans la carte de manière évidente.

L’entraînement de DecoupledNet est séparé entre le réseau de classification et celui de segmentation. Le réseau de classification est entraîné séparément avec de nombreuses images riches en classes. Le réseau de segmentation ainsi que les ponts sont ensuite entraînés ensemble avec le réseau pré-entraîné de la classification, sur un nombre correct d’images. DecoupledNet permet d’obtenir un entraînement plus efficace avec une base de données comprenant peu d’images, par rapport aux réseaux vus précédemment. Cela peut alors être intéressant pour le projet SAIAD, car nous ne disposons pas d’une base d’apprentissage conséquente.

3.1.3 SegNet

SegNet [6], un autre réseau proposé en 2015 est utilisé pour la segmentation multiple de scènes en temps réel. Il est plus ou moins semblable à DeconvNet, utilisant également des couches BN, mais les couches FC sont enlevées de l’architecture VGG16 (cf figure 3.5) ce qui rend le réseau plus petit et plus rapide à entraîner (passant de 134M de paramètres pour le DeconvNet à 14.7M). Son réseau déconvolutif diffère également de celui de DeconvNet, car il ne pratique pas de déconvolution, remplaçant ces couches par des couches de convolution simple.

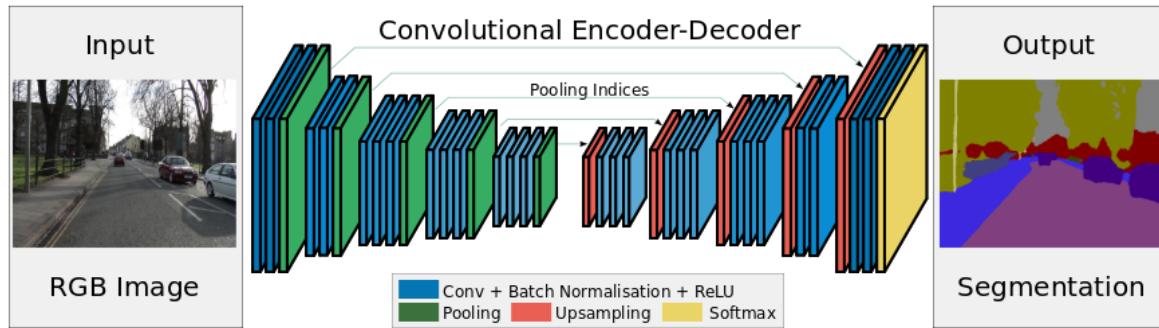


FIGURE 3.5 – SegNet, issue de [6]

La base d’images de SegNet, CamVid⁴ [20], n’est pas très volumineuse (367 images pour l’entraînement et 233 pour le test), mais elle est suffisante pour obtenir des résultats corrects sur la détection de 11 classes différentes. Pour l’entraînement, l’ensemble de la base d’images est mélangé et des mini-lots de 12 images sont choisis pour entraîner le réseau graduellement sur quelques images à la fois. Les résultats sont concluants pour les classes comprenant des objets de tailles importantes et en premier plan, mais le réseau rencontre des difficultés pour les objets plus petits et complexes en arrière-plan, car comme le réseau a été réduit pour être adapté à la segmentation en temps réel, les segmentations sont alors moins précises.

4. CamVid est une base de vidéos de rue du point de vue d’un automobiliste.

SegNet dans le milieu médical

Des recherches ont été menées sur l'utilisation de la méthode SegNet dans le milieu médical [21]. Alexander Kalinovsky et al. ont réalisé des segmentations d'images pulmonaires en reprenant le réseau SegNet, testé sur une base de cas comprenant 354 images médicales, dont 107 atteints de tuberculoses, de différents pays ainsi que différents scanners pour permettre un résultat plus généraliste.

Les Dice obtenus se situent entre 92.6% et 97.4% ce qui laisse entrevoir de très bons résultats même si la segmentation pulmonaire peut s'avérer être moins complexe que la segmentation de reins, étant donné que les poumons sont facilement délimitables sur les images médicales avec des couleurs assez distinctes par rapport au reste des éléments sur l'image.

3.1.4 U-Net

U-Net est un réseau créé en 2015 spécialement conçu pour la segmentation de cellules microscopiques [7]. Le système consiste en la capture du contexte général des premières couches et leur concaténation avec des couches plus lointaines qui permettent une localisation plus précise, comme pourrait le réaliser FCN-16s et FCN-8s, mais d'une manière différente. Il s'agit également du premier réseau n'utilisant pas l'architecture VGG16 pour son réseau convolutif, en supprimant certaines convolutions.

Le système est composé d'un réseau de neurones convolutif et déconvolutif, dans lequel chaque sur-échantillonnage du réseau déconvolutif est effectué à partir de déconvolution et non de unpooling. Après chaque déconvolution, le réseau concatène les images obtenues de la convolution correspondante au réseau convolutif avec celles obtenues après la déconvolution et réalise un rognage sur ces informations (cf figure 3.6). L'image finale est cependant plus petite que l'image d'entrée due à la déconvolution appliquée.

Le réseau ainsi entraîné sur 30 images de microscopie électronique a remporté le *ISBI cell tracking challenge 2015* qui est un concours sur la segmentation et le suivi des cellules.

U-Net dans la segmentation de foie

Ce réseau a beaucoup inspiré Patrick Ferdinand Christ et al. dans la création d'un nouveau système permettant de segmenter le foie et ses lésions en 2016 [22].

La segmentation s'effectue en plusieurs étapes. Tout d'abord, le réseau U-Net est appliqué aux images médicales pour la segmentation du foie, puis un autre réseau U-Net est appliqué

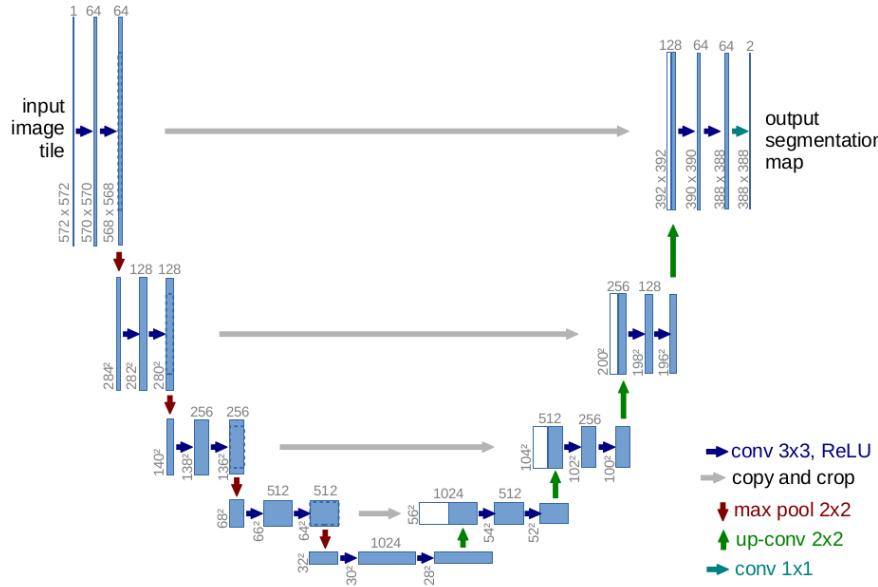


FIGURE 3.6 – U-Net, issue de [7]

pour segmenter ces lésions. Il est également ajouté en fin de traitement un CRF qui permet la construction en 3D de la segmentation (*3D CRF*).

Le réseau a été entraîné avec la méthode de validation croisée⁵ sur des images provenant de la base 3DIRCAD qui est une base de données médicale de segmentation, dans laquelle ont été compilées plus de 22000 coupes. Les résultats sont encourageants avec un Dice de 93%

3.1.5 Autres méthodes

La communauté scientifique est très active dans ce domaine, aussi de nombreuses autres méthodes existent et encore d'autres font leur apparition chaque année. C'est notamment le cas de la deuxième version de la méthode DeepLab [23] en 2016, améliorant son efficacité par rapport à la première version de 2015 [24], les deux méthodes se basant principalement sur les CRF.

ENet (Efficient neural NETwork) a été proposée en 2016 [25] et comme SegNet le but est de réaliser des segmentations en temps réel. Son réseau diffère des réseaux étudiés jusqu'ici, le réseau étant pensé pour réaliser les segmentations les plus rapides. Entraîné avec CamVid,

5. La validation croisée est un processus qui permet d'entraîner et tester un réseau en décomposant le jeu de données en sous-parties. Elle permet de tester une approche avec un jeu de données limité.

ENet obtient des résultats similaires à SegNet en termes de segmentation, tout en étant pratiquement 20 fois plus rapide que SegNet, ce qui en fait le réseau de neurones le plus rapide pour réaliser les segmentations.

3.2 Comparatif

Le tableau 3.1 nous permet de visualiser et comparer les différents réseaux de neurones pour la segmentation et le tableau 3.2 nous donne les résultats obtenus par ces différents réseaux. Toutes les méthodes, sauf FCN, utilisent un réseau déconvolutif qui est vu comme le miroir de leur réseau convolutif, car ce principe permet d'obtenir de meilleure segmentation. Le type de sur échantillonnage (déconvolution ou unpooling) varie en fonction des réseaux, mais n'a pas l'air d'avoir un réel impact sur les résultats.

Méthode	Année	Architecture VGG16	Effet "Miroir"	Deconvolution	Unpooling	FC (ou en convolution)	BN
FCN	2015	Oui	Non	Oui	Non	Oui	Non
DeconvNet	2015	Oui	Oui	Oui	Oui	Oui	Oui
DecoupledNet	2015	Oui	Oui	Oui	Oui	Oui	Oui
U-Net	2015	Non	Oui	Oui	Non	Oui	Non
SegNet	2015	Oui	Oui	Non	Oui	Non	Oui
Enet	2016	Non	Oui	Oui	Oui	Non	Oui
DeepLab	2015(v1) 2016(v2)	Oui	Oui	Oui	Non	Oui	Oui

TABLE 3.1 – Comparaison des différents réseaux

La plupart des méthodes se basent sur l'architecture VGG16, car il s'agit d'une architecture simple et efficace. Les deux méthodes n'utilisant pas VGG16 (U-Net et ENet) utilisent tout de même une architecture similaire. Enlever les couches FC réduit considérablement le nombre de paramètres dans le réseau, rendant ainsi la segmentation plus rapide. Cependant, la plupart des réseaux gardent leurs couches FC, car elles apportent des informations supplémentaires pour une segmentation plus précise.

Tous les résultats des segmentations sont calculés avec la mesure IU (cf équation 10). L'IU

Méthode	Langage	Matériels	Résultats
FCN	Caffe	NVIDIA TESLA k40c	62.2% mean IU PASCAL VOC 2012
DeconvNet	Caffe	NVIDIA GTX Titan X GPU	72.5% mean IU PASCAL VOC 2012
DecoupledNet	Caffe	NVIDIA GTX Titan X GPU	66.6% mean IU PASCAL VOC 2012
U-Net	Caffe	NVIDIA GTX Titan GPU	92.0% mean IU Phc-U373 77.5% mean IU DIC-HeLa
DeepLab	Caffe	NVIDIA GTX Titan X GPU	79.7% mean IU PASCAL VOC 2012
SegNet	Caffe	NVIDIA GTX Titan GPU	60.1% mean IU CamVid
Enet	Torch7	NVIDIA Titan TX1, NVIDIA GTX Titan GPU	51.3% mean IU CamVid

TABLE 3.2 – Comparaison des résultats obtenus entre les différents réseaux

ou IOU (*Intersection Over Union*) est utilisée comme mesure de précision pour comparer les segmentations réelles des segmentations calculées. *mean IU* est simplement la moyenne de l'IU de toutes les classes.

Le calcul de l'IU est le suivant :

Équation 10

$$IU_i = \frac{n_{ii}}{n_{ii} + n_{ji} + n_{ij}}$$

avec IU_i le IU de la classe i ,

n_{ii} le nombre de pixels correctement classifiés de la classe i ,

n_{ji} le nombre de pixels non classifiés, mais qui aurait dû l'être de la classe i ,

n_{ij} le nombre de pixels classifiés à i , mais que ne devrait pas l'être.

Les différentes méthodes ont été testées sur différentes bases de données : PASCAL VOC 2012, Phc-U373, CamVid et DIC-HeLa.

Toutes les méthodes sauf ENet sont implémentées en Caffe⁶ [26] et utilisent toutes des cartes graphiques puissantes pour l'entraînement, la plupart étant des NVIDIA GTX Titan GPU (À noter que la NVIDIA Titan TX1 est une carte graphique pour les systèmes embarqués).

Sur la base de données PASCAL VOC 2012, DeconvNet est le réseau obtenant les segmentations les plus précises, mais il s'agit également du réseau le plus lourd. Utiliser DecoupledNet peut être intéressant pour une petite base de cas même si les résultats ne sont pas meilleurs que DeconvNet. SegNet et ENet obtiennent les moins bons résultats, mais leur objectif est de réaliser des segmentations en temps réel (À noter que les résultats de SegNet et de ENet ne peuvent pas être comparés, les classes détectées des deux réseaux étant différentes). U-Net quant à lui, est un réseau très performant pour la segmentation de certaines cellules (Phc-U373 et DIC-HeLa sont des bases comprenant des images de diverses sortes de cellules) et son réseau plus petit est également moins lourd.

Conclusion

La plupart des réseaux que nous venons de présenter sont potentiellement intéressants dans le cadre des travaux du projet SAIAD. Suite à notre état de l'art, le Deep Learning a de bonnes chances de pouvoir reconnaître un organe déplacé et déformé. DeconvNet permettrait des segmentations précises, mais son entraînement serait trop conséquent. DecoupledNet pourrait être intéressant, car nous n'avons pas de grande base d'apprentissage, mais la précision des segmentations ne sera peut-être pas satisfaisante. La segmentation doit être la plus précise et fiable possible pour aider au mieux le chirurgien, limiter les erreurs d'interprétation et de diagnostics et préparer au mieux l'intervention.

Il sera nécessaire que notre entraînement ainsi que la réalisation de nos segmentations soient rapides, mais pas autant que peut l'être SegNet et ENet, la qualité des segmentations étant le point le plus important. Enfin, U-Net semble être le réseau le plus intéressant pour notre projet afin d'obtenir des résultats très précis.

Notons qu'ajouter des BN (*Batch Normalization* après chaque convolution et déconvolution et ajouter un CRF (*Conditional Random Field*) à la fin du réseau U-Net permettrait d'obtenir des résultats encore plus précis, même si le réseau s'en trouve alors légèrement alourdi.

6. Caffe est une librairie python pour le développement de réseau de neurones.

Chapitre 4

Implémentations et résultats

Après les états de l’art que nous avons présentés, cette dernière partie est consacrée à nos implémentations : méthode de segmentation FCN, normalisation utilisée pour notre base d’images, entraînement par validation croisée.

Les résultats seront, dans la suite de ce chapitre, présentés et discutés.

4.1 Méthode *Fully Convolutional Networks*

Nous avons choisi d’utiliser la méthode FCN, implémentée en python avec la librairie *caffe*, pour nos premiers tests sur les images de reins tumoraux. Le code source de cette méthode est proposé par ses auteurs sur github (service web d’hébergement et de gestion de développement logiciel).

Nous avons modifié le code afin de prendre en compte notre base d’images et nos classes détectées (rein, tumeur, artère, veine, médulla, cf figure 1) pour les segmentations.

4.1.1 Normalisation des images

Pour le bon fonctionnement de notre système, les images de la base doivent être *normalisées* (taille, couleur, ...). Dans le projet SAIAD, les images scanners et les segmentations correspondantes de sept patients différents sont disponibles. Les images scanners et leurs segmentations doivent être de la même taille lors de l’entraînement afin de pouvoir réaliser le lien entre les éléments et les segmentations attendues. De plus, les images scanners sont transformées en mode couleur RGB et les segmentations en noir et blanc avec des valeurs comprises entre 0 et N-1 pour chaque pixel où N est le nombre de classes.

Comme le montre la figure 4.1, nous avons choisi de labelliser la tumeur à 1, le rein à 2, la médulla à 3, la veine à 4 et l’artère à 5. Nous indiquons également le fond noir de l’image avec

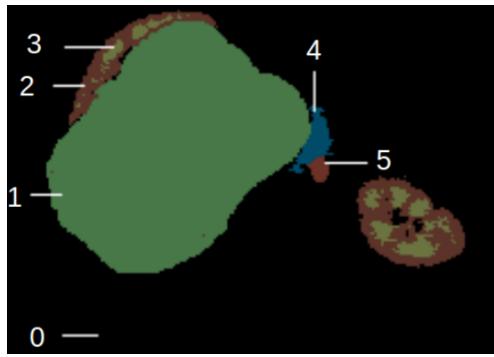


FIGURE 4.1 – Labellisation d'une segmentation

la valeur 0.

4.1.2 Entrainement

Une fois notre base d'images préparée, nous pouvons passer à la phase d'entraînement. Pour obtenir un réseau plus facile et rapide à entraîner, les paramètres des couches de la première à la cinquième convolution sont récupérés de l'entraînement avec la base PASCAL VOC 2012. Ceci facilite alors la phase d'apprentissage, car ces couches sont déjà entraînées pour détecter des objets. Les couches de convolutions remplaçant les couches FC et la déconvolution sont entraînées depuis zéro pour détecter nos classes en particulier.

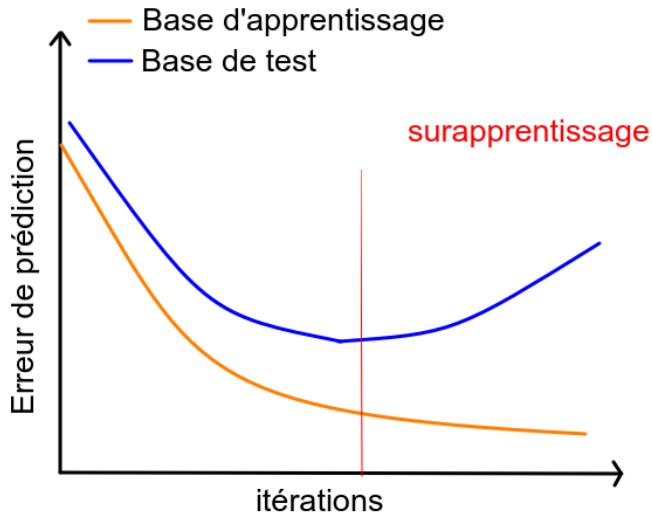


FIGURE 4.2 – Le surapprentissage durant l'entraînement

Nous utilisons une base d'entraînement pour l'apprentissage et une base de test pour tester notre réseau durant l'entraînement. Cela permet de vérifier qu'il n'existe pas de surapprentissage lors de l'entraînement et d'obtenir le nombre d'itérations maximales à atteindre pour un résultat optimal. En effet, si nous effectuons un surapprentissage, les segmentations de nos images de tests seront incorrectes alors que les segmentations sur la base d'apprentissage seront de bonne qualité. Le réseau n'est alors pas généralisable. Le moment le plus optimal pour arrêter l'entraînement se situe au moment où l'erreur calculée pour les images de tests est la plus faible avant d'augmenter pour passer en surapprentissage (cf figure 4.2)

Notre base d'apprentissage est composée de 12 images d'entraînements et 6 de tests d'un même patient. Les tests sont effectués sur l'ensemble des images actuellement mises à disposition dans le cadre de SAIAD.

Entraînement par validation croisée

Pour tester l'homogénéité de nos bases d'images, j'ai réalisé un entraînement par validation croisée, et plus particulière une *k-fold cross validation*. Le principe de cette validation croisée est de diviser notre base d'images en plusieurs sous-ensembles d'entraînement et de test afin de réaliser un entraînement sur chacune des configurations possibles avec ces sous-ensembles (cf figure 4.3). Ce principe d'entraînement permet alors de valider un modèle avec un jeu de données assez faible, les différentes permutations permettant de compenser le peu de données. Si les résultats obtenus sont similaires dans toutes les configurations, alors le jeu de données est homogène.

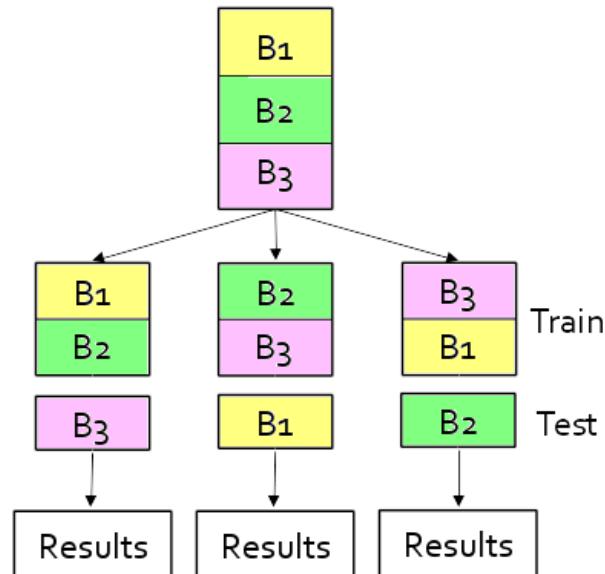


FIGURE 4.3 – Principe de la *k-fold cross validation*

Comme nous possédons une base contenant 18 images d'un même patient, il a été réalisé 3 sous-ensembles de 6 images ainsi qu'un entraînement sur ces 3 sous-ensembles. Les résultats obtenus illustrés par la Table 4.1 sont similaires d'un sous-ensemble à un autre ce qui permet de conclure que le jeu de données est homogène et que nous pouvons continuer nos entraînements sur ces bases d'images. Les résultats sont également relativement prometteurs pour la segmentation de la tumeur.

	SE1		SE2		SE3	
	Dice	IU	Dice	IU	Dice	IU
Moyenne	51%	42%	45%	37%	47%	38%
Médiane	53%	42%	44%	35%	49%	39%
Moyenne Tumeur	89%	83%	84%	76%	86%	81%
Médiane Tumeur	95%	90%	87%	79%	94%	89%

TABLE 4.1 – Résultats de la validation croisée sur les 3 sous-ensembles

4.1.3 Résultats avec FCN-32s/FCN-16s/FCN-8s

Nous avons dans un premier temps testé les trois types de réseaux de FCN : FCN-32s, FCN-16s et FCN-8s, pour nous assurer de leur performance, en utilisant seulement une image pour l'entraînement. Comme prévu, sur un même nombre d'itérations, FCN-8s est bien plus performant que FCN-16s qui l'est également plus que FCN-32s (cf figure 4.4). FCN-8s est le seul capable de segmenter les médullas et l'artère, comme le montre la figure 4.4, car il est capable de réaliser des segmentations plus fines. Il obtient un Dice moyen de 82% et un IU moyen de 71% (avec 99% de Dice et 98% d'IU pour la segmentation de la tumeur).

Nous nous basons donc naturellement sur le réseau FCN-8s pour la suite des entraînements.

4.1.4 Entraînement sur base non pré-traitée

Nous avons par la suite entraîné notre réseau avec la base de données comportant 12 images originelles de scanners. L'entraînement a duré une heure pour un total de 4000 itérations (une itération étant le passage dans le réseau et sa remontée). À chaque itération, l'erreur (ou coût), qui correspond à la différence entre l'image attendue et l'image calculée, est calculée à l'aide de la formule de l'équation 4. La figure 4.5 nous montre la courbe des erreurs calculées en fonction

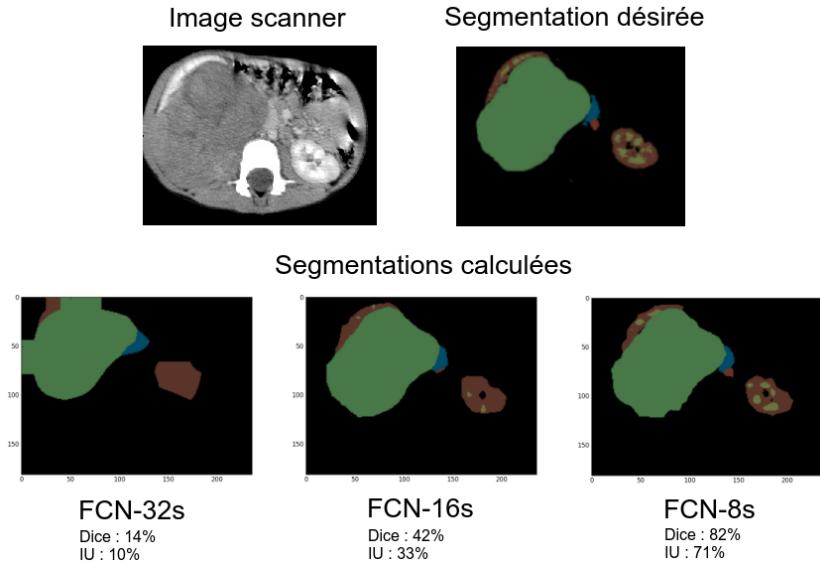


FIGURE 4.4 – Résultats des différents FCN

des itérations sur la base d’entraînement en violet et la courbe des erreurs sur la base de test en vert.

Bien que l’entraînement commence avec une erreur de 50000, cette dernière chute pour se stabiliser aux environs de 3000. La courbe de la base de test suit celle de l’entraînement : les erreurs se stabilisent également entre 2000 et 4000 itérations avec une légère baisse, mais nous n’avons pas pu observer de surapprentissage.

Le non-surapprentissage est dû tout d’abord à une base d’apprentissage trop petite et ensuite à une trop grande ressemblance des images de tests par rapport à ceux d’apprentissage, si bien que le réseau ne peut réussir à séparer les images d’apprentissage des images tests. Nous nous sommes donc arrêtés à 4000 itérations, étant donné que le réseau n’apprenait pour ainsi dire plus.

La courbe de l’erreur pour la base d’apprentissage n’est pas linéaire, car lors du calcul de l’erreur à l’aide de la technique de descente du gradient, nous obtenons des cas dans lesquels le calcul dépasse un minimum local ou global et s’éloigne alors d’une valeur optimale.

La table 4.2 représente les résultats en Dice (cf équation 9) et IU (cf équation 10) en pourcentage des segmentations sur les différentes images sur lesquelles le réseau s’est entraîné et la table 4.3 représente les résultats sur les images de tests. Il est à noter que les résultats donnant la valeur 1 ne sont pas forcément des segmentations parfaites, mais plutôt l’absence de segmentation d’un organe sur la segmentation attendue comme sur la segmentation désirée, ce qui donne un résultat 100% correct. De même les résultats à 0 ne signifient pas forcément

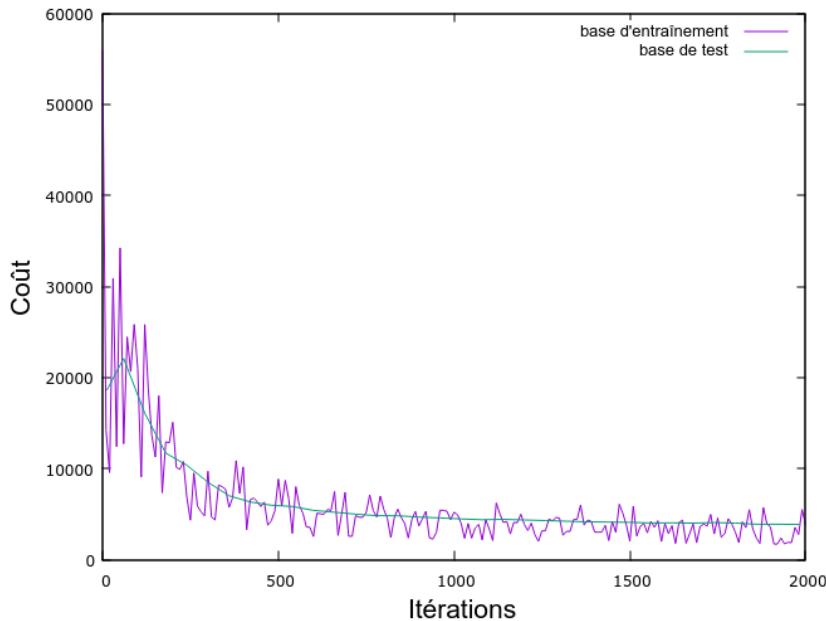


FIGURE 4.5 – Courbes de l’erreur en fonction des itérations pendant l’entraînement sur les images non pré-traitées

que l’organe en question n’a pas été segmenté, mais qu’il peut exister de nombreuses erreurs sur la segmentation calculée de sorte que les résultats ne peuvent être positifs.

Images	Tumeur		Rein		Médulla		Veine		Artère		Moyenne	
	Dice	IU										
1	0.88	0.76	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.58	0.56
2	0.95	0.91	0.80	0.67	1.00	1.00	0.00	0.00	0.43	0.27	0.64	0.57
3	0.93	0.87	0.35	0.22	0.21	0.11	0.78	0.64	0.55	0.38	0.56	0.44
4	0.95	0.90	0.38	0.23	0.00	0.00	0.03	0.02	0.00	0.00	0.27	0.23
5	0.96	0.92	0.27	0.16	0.00	0.00	0.73	0.57	0.62	0.45	0.52	0.42
6	0.96	0.93	0.53	0.36	0.24	0.14	0.03	0.02	0.52	0.35	0.46	0.36
7	0.97	0.94	0.37	0.23	0.37	0.23	0.00	0.00	0.42	0.26	0.43	0.33
8	0.95	0.91	0.28	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.21
9	0.93	0.86	0.35	0.21	0.00	0.00	0.00	0.00	0.18	0.10	0.29	0.23
10	0.93	0.87	0.26	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.20
11	0.60	0.43	0.31	0.18	0.00	0.00	0.00	0.00	0.52	0.35	0.29	0.19
12	0.00	0.00	0.63	0.58	0.00	0.00	1.00	1.00	0.58	0.41	0.44	0.37
Moyenne	0.83	0.77	0.46	0.35	0.24	0.21	0.21	0.19	0.32	0.21	0.41	0.34
Médiane	0.94	0.89	0.36	0.23	0.00	0.00	0.00	0.00	0.43	0.27	0.44	0.35

TABLE 4.2 – Résultats du Dice et de l’IU (en %) des segmentations calculées sur les images entraînées

Nous obtenons donc un Dice moyen de 41% et un IU moyen de 34% sur les images de la base d'entraînement. Les valeurs en IU des segmentations sont toujours égales ou inférieures aux valeurs en Dice, car le calcul de l'IU pénalise plus les différentes erreurs de segmentation que l'on peut commettre. Le réseau possède une grande facilité pour segmenter la tumeur (avec un Dice de 83% et un IU de 77%), mais peine avec les autres éléments. Ces éléments de plus petite taille peuvent fortement varier d'une coupe à une autre ce qui peut rendre la généralisation des ces éléments plus compliqués pour le réseau. La tumeur est quant à elle facilement segmentée, car sa forme est volumineuse, facilement différentiable par rapport aux autres éléments et similaire d'une coupe à une autre sur un même patient (rappelons que nous travaillons sur un jeu de données concernant un seul patient).

	Tumeur		Rein		Médulla		Veine		Artère		Moyenne	
Images	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU
1	0.88	0.78	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.38	0.36
2	0.94	0.89	0.35	0.21	0.00	0.00	0.37	0.23	0.00	0.00	0.33	0.27
3	0.95	0.91	0.26	0.15	0.00	0.00	0.15	0.08	0.60	0.43	0.39	0.31
4	0.96	0.92	0.29	0.17	0.00	0.00	0.07	0.04	0.32	0.19	0.33	0.26
5	0.88	0.79	0.15	0.09	0.00	0.00	0.00	0.00	0.10	0.05	0.23	0.18
6	0.78	0.64	0.27	0.16	0.00	0.00	0.00	0.00	0.44	0.28	0.30	0.22
Moyenne	0.90	0.82	0.22	0.13	0.17	0.17	0.10	0.06	0.24	0.16	0.33	0.27
Médiane	0.91	0.84	0.27	0.16	0.00	0.00	0.04	0.02	0.21	0.12	0.33	0.27

TABLE 4.3 – Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images non entraînées

Les résultats sur la base de tests (33% de Dice moyen et 27% d'IU moyen, cf figure 4.3) sont plus faibles que ceux précédemment obtenus sur la base d'apprentissage, car le réseau ne s'est pas entraîné sur ce jeu de données, à une exception près pour les résultats de segmentation de la tumeur (90% de Dice et 82% d'IU) qui sont supérieurs aux résultats de la tumeur pour la base d'entraînement.

Nous constatons donc avec les résultats des tableaux 4.2 et 4.3 que la méthode de segmentation FCN n'est pas la plus adaptée pour la segmentation des images de reins tumoraux du projet SAIAD. FCN possède des difficultés pour segmenter précisément des objets complexes de petite taille comme peut l'être la médulla sur nos segmentations. Néanmoins notre réseau possède des facilités pour segmenter la tumeur sur un même patient. Nous pourrions alors résoudre un problème de reproductibilité de la tumeur en segmentant manuellement seulement quelques coupes de celle-ci (extrémités et milieu) et réaliser ensuite un réseau apprenant à segmenter la tumeur sur toutes les coupes grâce à ces quelques segmentations manuelles. Avec un tel réseau, il serait alors possible de réaliser une segmentation complète d'une tumeur en effectuant seulement quelques segmentations manuelles.

Une autre cause de difficulté à segmenter est due au manque de netteté des images scanners et aux nuances de gris trop similaires d'un élément à un autre. Nous avons donc choisi de réaliser le même entraînement, mais cette fois-ci avec des images pré-traitées.

4.1.5 Entraînement sur base pré-traitée

Pour améliorer nos résultats de segmentations, nous avons réalisé un pré-traitement simple en modifiant le contraste et la luminosité des images de notre jeu de données (cf figure 4.6) pour que la différence entre les différents niveaux de gris des éléments soit plus importante.

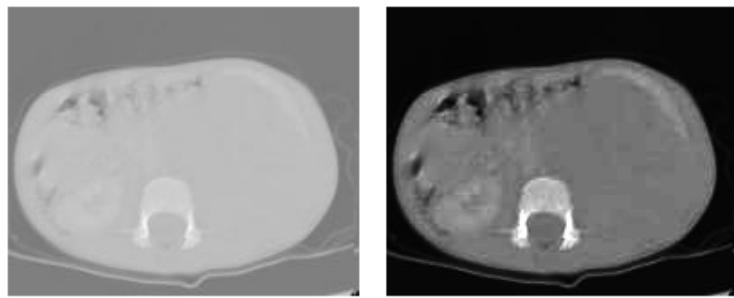


FIGURE 4.6 – Exemple du pré-traitement appliqu 

L'entraînement sur cette base d'images pré-traitées s'est av r  l g r m n t plus efficace que pour la base d'images non pré-trait es. Sur la figure 4.7 montrant l'entra nement vu pr c demment de la base d'images non pr e t es ainsi que l'entra nement de la base d'images pr e t es, les courbes d'entra nement et de test pour les images pr e t es sont en dessous des courbes pour la premi re base avec une erreur moyenne de 2000 contre les 3000 pr c demment obtenues.

Sur les tableaux 4.4 et 4.5 nous pouvons remarquer que les r sultats sont en hausse avec un Dice moyen de 58% et un IU moyen de 48%,   une exception pr s pour les r sultats de l'art re en baisse de 3   4%. Toutefois, ce sont les r sultats des segmentations sur les images d'apprentissage qui se sont le plus significativement am lior s (+17% de Dice et +14% d'IU) alors que les segmentations des images de tests ne se sont am lior es que de 6% de Dice et de 4% d'IU. Cette technique a tout de m me am lior  les segmentations et les figures 4.8, 4.9 et 4.10 montrent les segmentations obtenues pour la base d'images non pr e t es et pr e t es. La premi re colonne correspond aux images scanners d'origines pour l'entra nement sur les tableaux 4.8 et 4.9, ainsi que pour le test pour le tableau 4.10. La deuxi me colonne correspond aux m mes images auxquelles a  t  r alis  un pr e t ement simple avec modification du contraste et de la luminosit  et la troisi me colonne les segmentations attendues. Nous avons ensuite la colonne des segmentations calcul es pour les images non pr e t es (de la premi re

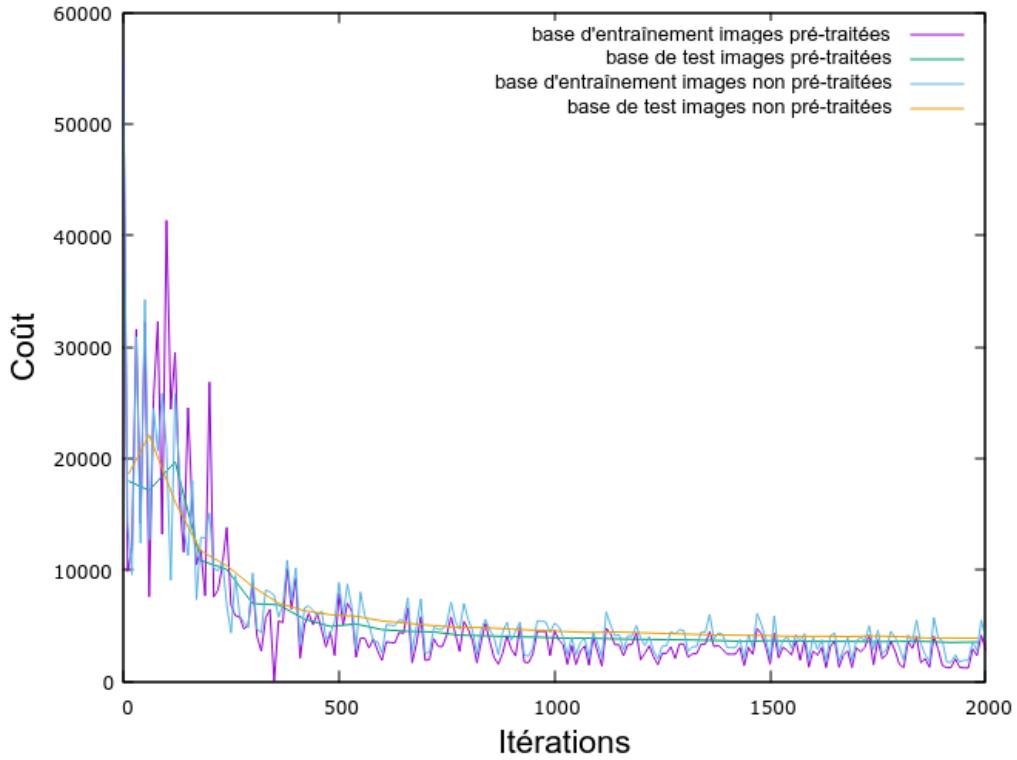


FIGURE 4.7 – Courbes de l'erreur (ou coût) en fonction des itérations pendant l'entraînement des images pré-traitées et non pré-traitées

Images	Tumeur		Rein		Médulla		Veine		Artère		Moyenne	
	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU
1	0.94	0.89	1.00	1.00	1.00	1.00	0.48	0.32	0.00	0.00	0.68	0.64
2	0.96	0.92	0.87	0.77	1.00	1.00	0.67	0.50	0.76	0.62	0.85	0.76
3	0.95	0.91	0.53	0.36	0.33	0.20	0.75	0.60	0.66	0.49	0.64	0.51
4	0.95	0.91	0.44	0.28	0.46	0.30	0.68	0.51	0.00	0.00	0.51	0.40
5	0.96	0.92	0.43	0.28	0.24	0.14	0.62	0.45	0.64	0.47	0.58	0.45
6	0.97	0.95	0.59	0.42	0.37	0.23	0.73	0.58	0.62	0.45	0.66	0.52
7	0.97	0.95	0.46	0.30	0.53	0.36	0.33	0.20	0.61	0.44	0.58	0.45
8	0.97	0.94	0.57	0.39	0.52	0.35	0.58	0.41	0.51	0.34	0.63	0.49
9	0.96	0.92	0.64	0.47	0.00	0.00	0.51	0.34	0.63	0.47	0.55	0.44
10	0.96	0.92	0.49	0.33	0.03	0.01	0.00	0.00	0.43	0.28	0.38	0.31
11	0.87	0.77	0.64	0.47	0.00	0.00	0.00	0.00	0.67	0.51	0.44	0.35
12	0.00	0.00	0.79	0.65	0.00	0.00	1.00	1.00	0.73	0.57	0.50	0.44
Moyenne	0.87	0.83	0.62	0.48	0.37	0.30	0.53	0.41	0.52	0.39	0.58	0.48
Médiane	0.96	0.92	0.58	0.41	0.35	0.22	0.60	0.43	0.63	0.46	0.58	0.45

TABLE 4.4 – Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images pré-traitées et entraînées

	Tumeur		Rein		Médulla		Veine		Artère		Moyenne	
Images	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU	Dice	IU
1	0.85	0.73	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.37	0.35
2	0.95	0.90	0.50	0.33	0.43	0.28	0.55	0.38	0.00	0.00	0.49	0.38
3	0.97	0.93	0.41	0.26	0.14	0.08	0.02	0.01	0.10	0.05	0.33	0.27
4	0.95	0.91	0.32	0.19	0.53	0.36	0.21	0.12	0.22	0.12	0.45	0.34
5	0.90	0.82	0.32	0.19	0.00	0.00	0.00	0.00	0.38	0.24	0.32	0.25
6	0.87	0.78	0.46	0.30	0.00	0.00	0.00	0.00	0.52	0.35	0.37	0.29
Moyenne	0.92	0.85	0.34	0.21	0.35	0.29	0.13	0.09	0.20	0.13	0.39	0.31
Médiane	0.93	0.86	0.37	0.23	0.29	0.18	0.01	0.01	0.16	0.09	0.37	0.32

TABLE 4.5 – Résultats du Dice et de l'IU (en %) des segmentations calculées sur les images pré-traitées et non entraînées

colonne) et la colonne des segmentations pour les images pré-traitées (de la deuxième colonne).

L'un des exemples les plus significatifs confirmant les valeurs des tableaux est celui de la ligne 2 sur la figure 4.9. Sur la segmentation calculée d'après l'image non pré-traitée, nous n'obtenons aucune segmentation des médullas, d'artère ni de veine et le réseau peine à effectuer une segmentation des deux reins. Sur la segmentation d'après l'image pré-traitée en revanche, nous obtenons des segmentations des médullas, d'artère et de veine et nous avons une amélioration pour les segmentations des reins. Nous voyons également que les segmentations de la tumeur se rapprochent des segmentations désirées. Ces résultats confirment donc les valeurs de Dice et d'IU précédemment calculées.

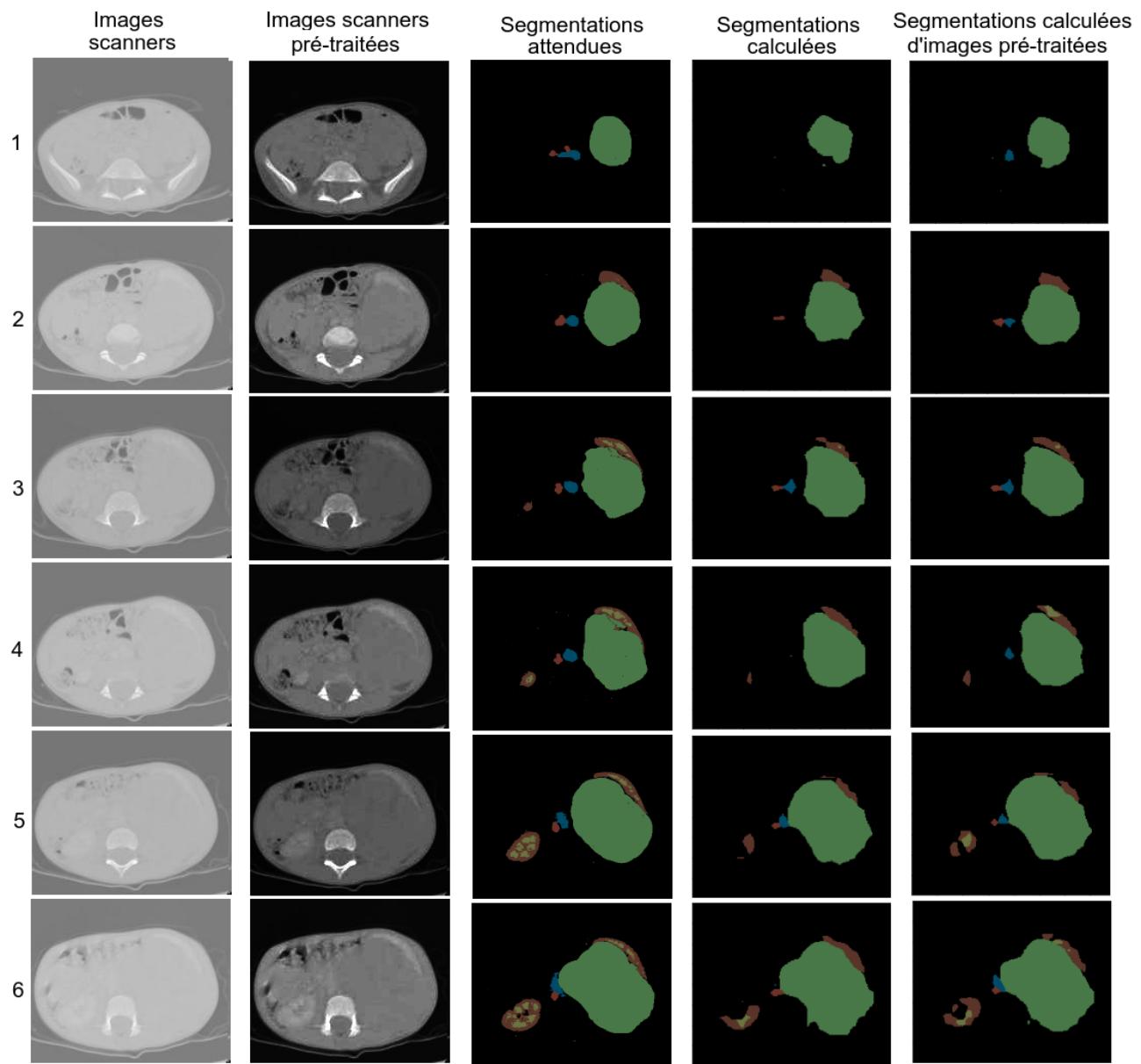


FIGURE 4.8 – Résultats des segmentations sur les images d'entraînement

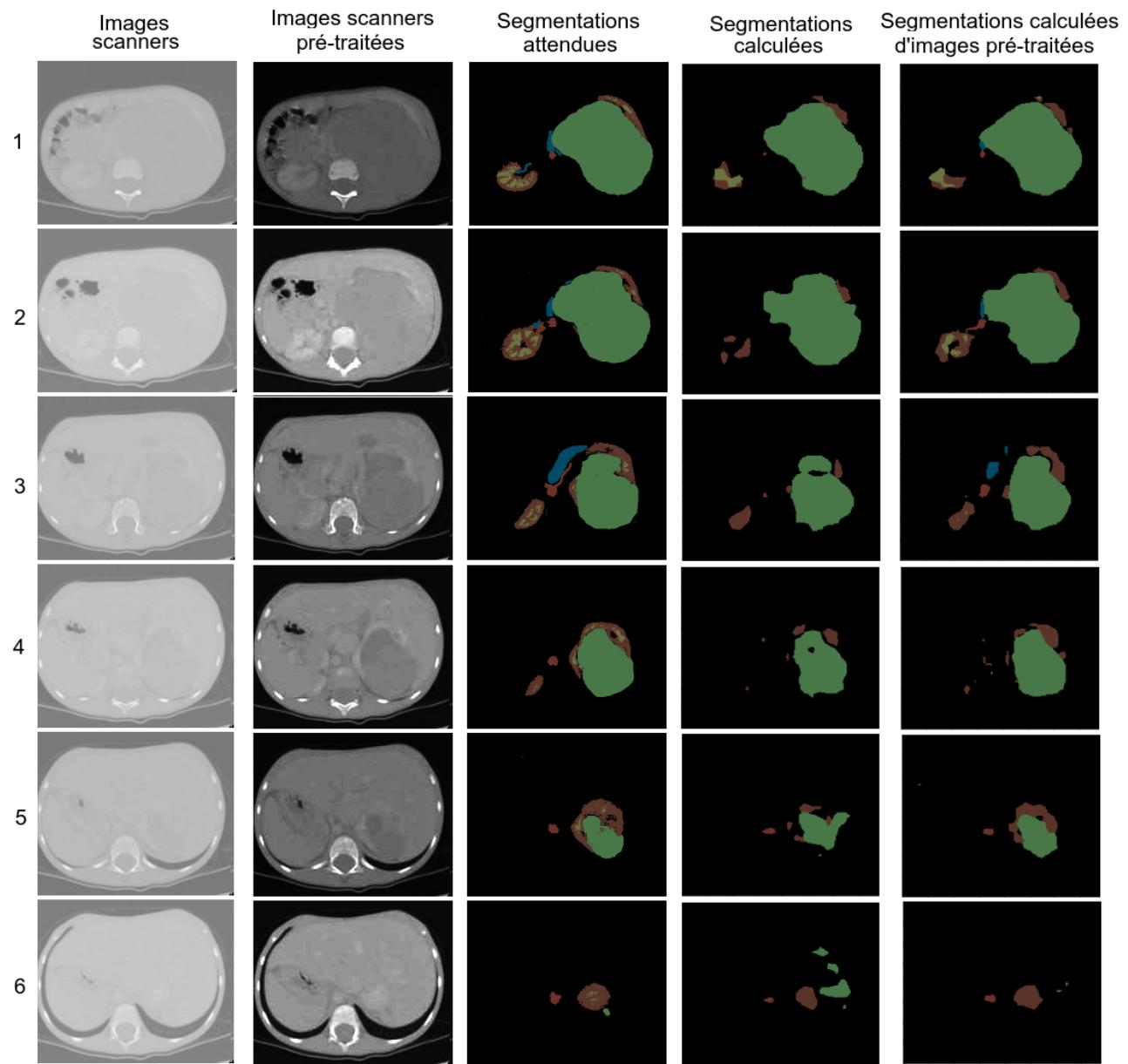


FIGURE 4.9 – Résultats des segmentations sur les images d'entraînement (suite)

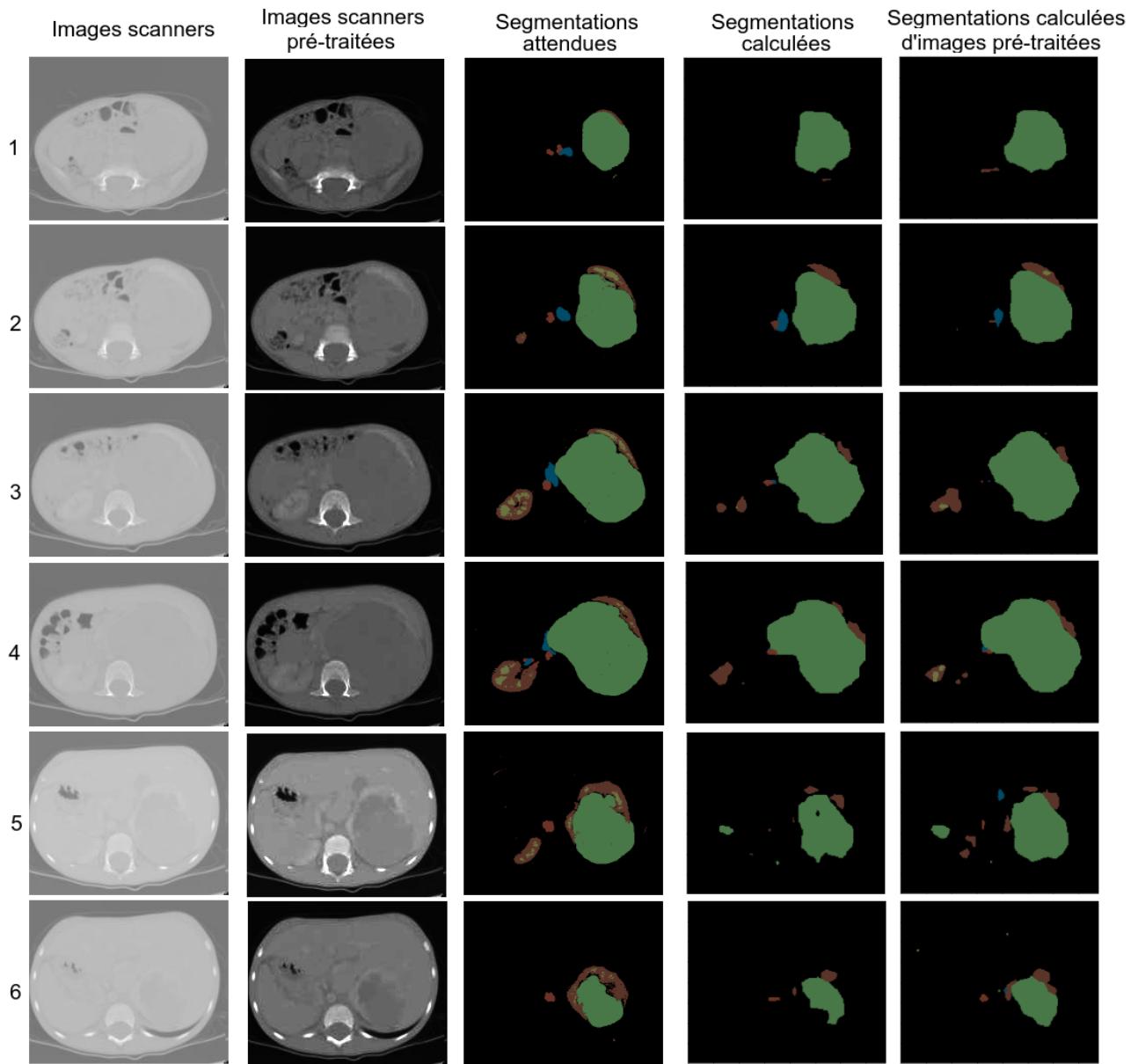


FIGURE 4.10 – Résultats des segmentations sur les images de test

4.1.6 Comparaison des résultats

Croissance de régions

La segmentation par Deep Learning et plus particulièrement par la méthode FCN permet des segmentations correctes et plus particulièrement pour la segmentation de la tumeur. Pour tester son efficacité par rapport à d'autres techniques de segmentation, nous avons choisi de réaliser une segmentation par Croissance de régions. Le principe de la segmentation par Croissance de région et d'ajouter manuellement des pixels germe sur l'images qui vont progressivement grossir sur les pixels voisins de valeurs proches.

Le résultat obtenu en comparaison du résultat par Deep Learning est illustré sur la figure 4.11. Nous utilisons pour les techniques des images scanners pré-traitées, mais le pré-traitement n'est pas le même. Pour le Deep Learning nous avons le pré-traitement de modification de contraste et de luminosité, alors que pour la Croissance de régions nous avons un pré-traitement plus conséquent, avec de la modification du contraste et de la luminosité, une égalisation d'histogramme, un filtre médian permettant de flouter et lisser l'image ainsi qu'un *unsharp mask* qui est une puissante technique de modification de contraste. La technique de Croissance de région n'est pas capable de réaliser une segmentation sur l'image pré-traitée du Deep Learning, car les nuances de gris sont trop proches, c'est pourquoi le pré-traitement appliqué est plus conséquent.

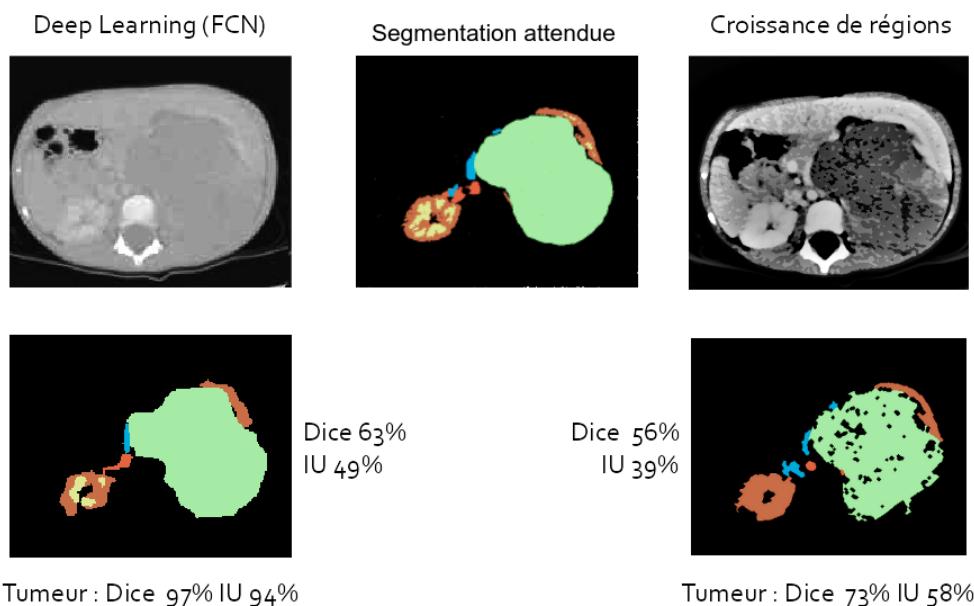


FIGURE 4.11 – Comparaison d'un résultat par Deep Learning avec un résultat par Croissance de région

La segmentation par FCN (Deep Learning) est plus précise que la segmentation par Croissance

de régions avec 63% de Dice et 49% d'IU contre 56% et 39%, alors que les nuances de gris des éléments sur l'image scanners sur Deep Learning sont plus proches.

Watershed

De même, le test de segmentation par Watershed. Le principe de cette méthode est de considérer les nuances de gris de l'image scanner comme un relief topographique d'où l'on simule une inondation. Le pré-traitement appliqué est alors le même que celui par Croissance de régions, car cette méthode fonctionne également avec les nuances de gris.

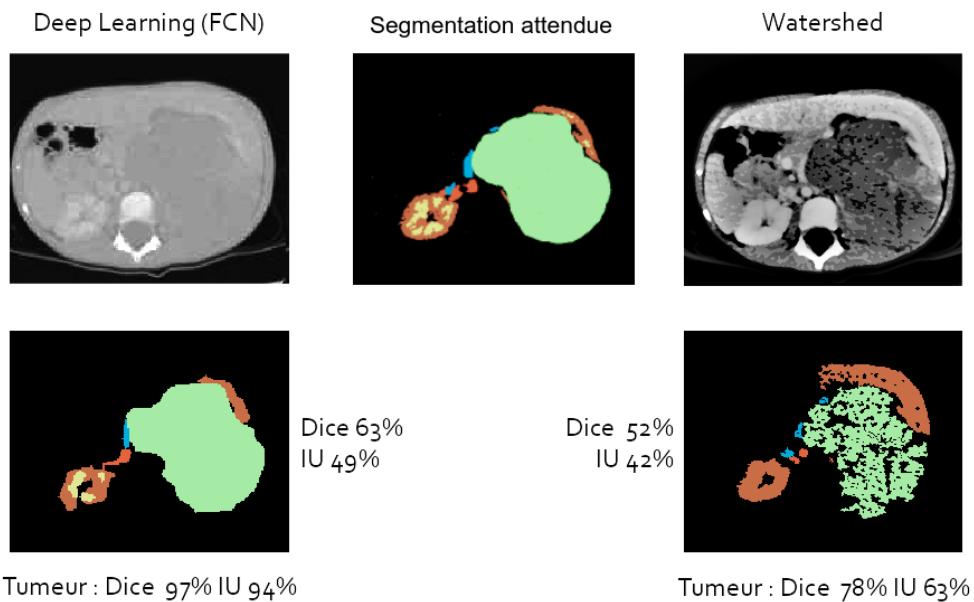


FIGURE 4.12 – Comparaison d'un résultat par Deep Learning avec un résultat par Watershed

Les résultats de la figure 4.12 permettent également de conclure que la segmentation par FCN est de meilleure qualité que la segmentation par Watershed.

Conclusion

Même si les résultats sont concluants pour ce qui est de la segmentation de la tumeur, ils le sont moins pour la segmentation de tous les autres éléments (veine, artère, médulla,...). Le réseau a de grandes difficultés pour segmenter les petits éléments et FCN n'est pas le réseau le plus adapté pour ce type de segmentation, mais sa facilité à segmenter la tumeur permet d'entrevoir une solution à un problème de reproductibilité de celle-ci sur toutes les coupes d'un patient.

L'utilisation d'une base d'images plus importante comportant plusieurs patients apporterait des résultats plus significatifs et un réseau permettant des segmentations plus généralisables. Mais, au cours de cette période de lancement du projet SAIAD, nous avons manqué d'images pour enrichir la bibliothèque nécessaire pour des calculs plus significatifs ainsi que d'images scanners et de segmentations.

Pour finir, nous avons montré que l'ajout d'un pré-traitement avant l'entraînement peut être une technique pour améliorer les résultats des segmentations.

Conclusion et perspectives

Conclusion

Ce travail a permis d'étudier l'apport de l'utilisation du Deep Learning dans la segmentation d'images médicales, le but final étant de lier plusieurs techniques de segmentation automatiques pour développer un outil performant.

Nous avons montré dans les différentes parties de ce rapport que la méthode de segmentation FCN (*Fully Convolutional Networks*) n'est pas assez efficace sur des images complexes pour obtenir une segmentation précise, bien que les résultats des segmentations pour les tumeurs soient assez concluants. L'ajout d'un pré-traitement à la base de données aurait un impact sur l'entraînement du réseau et permettrait de gagner en précision sur les segmentations calculées.

Nous avons rencontré quelques obstacles matériels comme le manque d'images scanners et de segmentation pour réaliser une grande base hétérogène et nous aurons besoin pour la suite du projet du mésocentre pour obtenir une grande puissance de calcul capable d'entraîner notre réseau avec une base d'images assez conséquente.

Perspectives

Les résultats de segmentation de la tumeur avec la méthode FCN permettent d'entrevoir une solution pour la reproductibilité de la tumeur sur toutes les coupes d'un même patient en utilisant quelques coupes segmentées manuellement, ce qui permettrait un gain de temps pour la segmentation.

Notre état de l'art nous a également permis de montrer qu'il serait intéressant de réaliser le réseau U-Net en ajoutant des BN (*Batch Normalization*) entre chaque convolution et un CRF (*Conditional Random Field*) à la fin du réseau pour obtenir des segmentations les plus précises, même pour les médullas.

Il pourrait également être intéressant de réaliser un réseau et un entraînement en parallèle pour

chaque élément, ce qui pourrait augmenter la performance des résultats.

Ce stage m'aura permis de découvrir l'intelligence artificielle par Deep Learning et son application dans le traitement d'image ainsi que dans le domaine médical. J'ai pu apprendre le fonctionnement d'un projet européen comme SAIAD et m'intégrer à l'institut FEMTO-ST. L'opportunité m'est donnée de continuer sur le projet SAIAD en tant qu'ingénieur et de pouvoir réaliser ma thèse dans ce domaine, plus particulièrement dans l'agrégation et l'arbitrage des différentes techniques de segmentation automatiques pour déterminer la configuration la plus optimale.

Mon travail se poursuit actuellement par la comparaison de mes résultats avec ceux d'autres techniques de segmentation (RàPC et modèles de Markov) également étudiées par M.Florent MARIE et M.Thibault DELAVELLE, dans le but de réaliser un outil utilisant ces différentes techniques afin d'obtenir des résultats à la fois rapides, et suffisamment précis et fiables pour les médecins.

Bibliographie

- [1] Kunihiko Fukushima and Sei Miyake. Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.
- [2] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1) :1929–1958, 2014.
- [3] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4) :640–651, 2016.
- [4] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *IEEE International Conference on Computer Vision, ICCV 2015*, pages 1520–1528, Washington, DC, USA, 2015. IEEE Computer Society.
- [5] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1495–1503. Curran Associates, Inc., 2015.
- [6] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, and Senior Member. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *Computer Vision and Pattern Recognition CoRR*, pages 1–14, 2015.
- [7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.
- [8] F Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6) :386–408, 1958.
- [9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3) :1, 1988.
- [10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.
- [11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, page 275, 2011.

- [12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.
- [13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [15] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge : A retrospective. *International Journal of Computer Vision*, 111(1) :98–136, 2015.
- [16] Austin Ray. Lung tumor segmentation via fully convolutional neural networks. Technical report, Standford University, 2016.
- [17] Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*, 2015.
- [18] Vladlen Koltun and Philipp Krähenbühl. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS, CoRR*, pages 1–9, 2012.
- [19] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks : Visualising image classification models and saliency maps. *CoRR*, 2013.
- [20] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video : A high-definition ground truth database. *Pattern Recognition Letters*, 30(2) :88–97, 2009.
- [21] A Alexander Kalinovsky and Vassili Kovalev. Lung image segmentation using deep learning methods and convolutional neural networks. In *Pattern Recognition and Information Processing - PRIP*. Minsk : Publishing Center of BSU, 2016.
- [22] Patrick Ferdinand Christ, Mohamed Ezzeldin A Elshaer, Florian Ettlinger, Sunil Tatarvarthy, Marc Bickel, Patrick Bilic, Markus Rempfler, Marco Armbruster, Felix Hofmann, Melvin D'Anastasi, et al. Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 415–423. Springer, 2016.
- [23] Liang-chieh Chen, George Papandreou, Senior Member, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *Computer Vision and Pattern Recognition, CoRR*, pages 1–14, 2016.
- [24] Liang-chieh Chen, Los Angeles, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L Yuille, and Los Angeles. Semantic image segmentation with deep convolutional

- nets and fully connected crfs. In *International Conference on Learning Representations*, pages 1–14, San Diego, United States, May 2015.
- [25] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet : A deep neural network architecture for real-time semantic segmentation. *Computer Vision and Pattern Recognition, CoRR*, page 10, 2016.
- [26] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

Annexes

Annexe 1 : Le neurone biologique

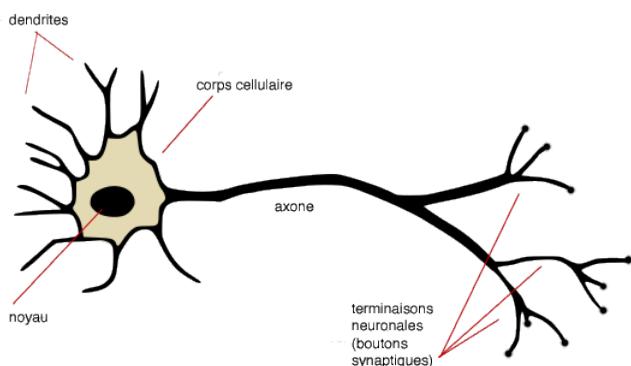


FIGURE 4.13 – Neurone biologique

Le neurone biologique (cf figure 4.13) est une cellule nerveuse du système nerveux. Il reçoit, traite et envoie des informations sous forme de signaux électriques à ses neurones voisins. Il reçoit des informations grâce aux dendrites qui sont les récepteurs du neurone. Il traite alors l'information dans son corps cellulaire où il effectue la somme des informations qui lui parviennent. Il renvoie alors le résultat dans l'axone, le signal de sortie. L'information arrive alors vers les synapses qui permettent de modifier les connexions entre les neurones. Les synapses possèdent une sorte de mémoire et en fonction de leur activation répétée ou non, les connexions synaptiques peuvent se modifier. Certaines synapses sont alors excitatrices et activent la connexion vers certains neurones tandis que d'autres peuvent être inhibitrices et stopper les signaux électriques.

Nous pouvons alors constater que les neurones artificiels se sont fortement inspirés des biologiques. Les liens entre neurones peuvent être comparés avec les dendrites et les axones. Nous pratiquons dans les deux cas la somme des informations et les poids des neurones artificiels ainsi que la fonction d'activation peuvent être comparés aux synapses.

Résumé

Ce rapport présente le travail que j'ai effectué dans le cadre du stage de fin d'études de Master 2 Informatique option Systèmes Distribués et Réseaux spécialité Recherche, de l'UFR Sciences et Techniques de l'Université de Bourgogne Franche-Comte. Le stage a eu lieu au sein de l'équipe DEODIS au laboratoire FEMTO-ST et fait partie du projet européen SAIAD (Segmentation Automatique de reins tumoraux chez l'enfant par Intelligence Artificielle Distribuée).

Nous introduisons une méthode particulière d'intelligence artificielle : Le Deep Learning, adapté dans le traitement et la segmentation d'images médicales. Après un état de l'art sur les différentes techniques existantes par Deep Learning, l'implémentation de programmes de segmentation d'images de reins tumoraux du projet SAIAD et une discussion des résultats des segmentations obtenues sont présentées.

Mots-clés

Deep Learning, Segmentation d'image, réseaux de neurones, tumeur du rein, projet européen SAIAD.

Abstract

This report presents my research work carried out during the internship of Master 2 course in Computer Science on Distributed Systems and Networks at UFR Sciences and Technologies in University of Bourgogne Franche-Comte. The internship took place within the DEODIS team in the FEMTO-ST laboratory and is part of the SAIAD project, which allows automated segmentations by distributed artificial intelligence of tumor kidneys in children.

This report introduces a particular method of artificial intelligence : Deep Learning, adapted in the treatment and segmentation of medical images. After a state of the art on the different techniques of segmentations existing by Deep Learning, the implementation of segmentation program on the tumor kidney images of the SAIAD project and discusses the results of the segmentations obtained are presented.

Keywords

Deep Learning, Image Segmentation, neural networks, tumor kidney, european project SAIAD.