UCLouvain

École polytechnique de Louvain

Deep learning pour la segmentation des cavités cardiaques en vue de la stratification d'embolie pulmonaire

Auteur: **Aline Marlair**
Promoteurs: **Christophe De Vleeschouwer, Benoît Macq**
Lecteurs: **Alexandre Englebert, Estelle Loyen, Ana Maria Barragan Montero**
Année académique 2022–2023
Master [120] : ingénieur civil biomedical

Abstract

In Belgium, over 6,000 people suffer from pulmonary embolism every year, and almost 400 of them die. In France, over 47,000 cases are recorded every year. Pulmonary embolism is one of the main causes of cardiovascular mortality, and a major health problem worldwide. The short-term effects can be dangerous if the patient is not properly treated when an embolism is detected. Pulmonary embolisms also have an impact on the heart. In order to identify the risks correctly, different biomarkers such as the volume or diameter of the heart chambers could be used. To calculate these values, a segmentation of the different chambers of the heart is necessary. This requires considerable time on the part of physicians. In the first part of this work, we aim to use artificial intelligence, and more specifically deep learning, to automatically segment the heart's chambers. The model we have chosen to use is a three-dimensional U-Net. It is trained using a database of public data from the *Multi Modality Whole Heart Segmentation* challenge, taken by CT scan. These data are twenty in number, with a resolution of $512 \times 512 \times C$, where the value of C varies according to the patient. During training, different resolutions are used and different parameters are tested. Results are quantified using the dice metric. The best dice obtained is 0.72 on average over the different parts of the heart with a resolution of $64 \times 64 \times 64$. A dice of 0.7 is achieved with a resolution of $128 \times 128 \times 128$ and the use of a pre-trained model. In the second part of this work, we received 48 data from *cliniques universitaires Saint Luc* to evaluate the performance of our model. These data also had a resolution of $512 \times 512 \times C$ but had to be modified to be similar to the training data. Finally, the results obtained seem promising, but our model still needs some improvement.

Résumé

En Belgique, plus de 6000 personnes par an sont victimes d'une embolie pulmonaire, presque 400 d'entre elles en meurent. En France, c'est plus 47000 cas recensés chaque année. Les embolies pulmonaires représentent une des principales causes de mortalité cardiovasculaire et constituent un problème de santé majeur dans le monde. Les effets à court terme peuvent s'avérer dangereux si le patient n'est pas soigné correctement lorsqu'une embolie est détectée. Les embolies pulmonaires ont également un impact sur le coeur. Afin d'identifier les risques correctement, différents biomarqueurs comme le volume ou le diamètre des cavités cardiaques pourraient être utilisés. Pour calculer ces valeurs, une segmentation des différentes cavités du coeur est nécessaire. Cela demande un temps considérable et précieux aux médecins. Dans la première partie de ce travail, nous tentons d'utiliser l'intelligence artificielle et plus particulièrement l'apprentissage profond afin de segmenter automatiquement les cavités du coeur. Le modèle que nous avons choisi d'utiliser est un U-Net en trois dimensions. Il est entraîné grâce à une base de données publiques issue du challenge *Multi Modality Whole Heart Segmentation*, celles-ci ont été prises par tomodensitométrie. Ces données sont au nombre de vingt et leur résolution est de $512 \times 512 \times$ C où la valeur de C est variable en fonction des patients. Lors de l'entraînement, différentes résolutions sont utilisées et différents paramètres sont également testés. Les résultats sont quantifiés grâce à la métrique de dice. Le meilleur dice obtenu est de 0,72 en moyenne sur les différentes parties du coeur avec la résolution de $64 \times 64 \times 64$. Un dice de 0,7 est atteint avec une résolution de $128 \times 128 \times 128$ et l'utilisation d'un modèle pré-entraîné. Dans la deuxième partie de ce travail, nous avons reçu 48 données des *cliniques universitaires Saint Luc* pour évaluer les performances de notre modèle. Ces données ont également une résolution de $512 \times 512 \times$ C mais ont du être modifiées pour être semblables aux données d'entraînement. Finalement, les résultats obtenus semblent prometteurs mais notre modèle nécessite encore certaines améliorations.

Remerciements

L'aboutissement de ce mémoire et de mes études n'auraient pas été possible sans le soutien, l'aide et les conseils de nombreuses personnes.

Avant tout, je voudrais remercier mes promoteurs, le professeur Benoit Macq et le Professeur Christophe De Vleeschouwer, de m'avoir permis de travailler sur un sujet aussi intéressant. Ils ont suivi et conseillé mon travail tout au long de l'année.

Ensuite, je suis profondément reconnaissantes envers Estelle Loÿen, Alexandre Englebert et Maxence Wynen pour tout le temps qu'ils ont consacré à répondre à mes questions, à me conseiller dans ma manière de travailler et à relire et corriger mon mémoire. Sans leur aide précieuse, ce travail n'aurait pas pu se concrétiser.

Je remercie Julie Nizet, Benoit Ghaye, Emmanuel Coche ainsi que Perrine Triqueneaux pour les données reçues aux *cliniques universitaires Saint Luc*. Je remercie également Ana Maria Barragan Montero pour sa lecture de mon mémoire.

Finalement, je tiens à remercier ma famille qui m'a soutenue durant ces années d'études. Plus particulièrement ma maman qui a passé de nombreuses heures à corriger ce travail, mon papa qui a également relu ce mémoire. Je remercie aussi mes amies pour l'ambiance de travail instaurée durant toute cette année ainsi que leur soutien. Pour finir, je remercie Gauthier pour ses nombreux conseils, toutes ses relectures et ses encouragements quotidiens.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 L'embolie pulmonaire

Les embolies pulmonaires sont l'une des principales causes de mortalité cardiovasculaire et constituent un problème de santé majeur dans le monde. En Belgique, elles touchent chaque année 6000 personnes et entre 300 et 400 d'entre elles en meurent [1]. En France, c'est plus de 47000 cas recensés sur l'année 2020, dont 52 % de femmes. Il est également à noter que cette pathologie touche principalement des personnes plus âgées avec une moyenne d'âge qui s'élève à 69 ans. Les patients ayant plus de 75 ans représentent 42 % des personnes touchées [2].
Les embolies peuvent survenir rapidement et de manière inattendue. Il s'agit d'une obstruction de l'une des artères des poumons, la plupart du temps due à un caillot de sang qui se forme dans une veine de la jambe. Celui-ci est causé par une thrombose veineuse profonde (TVP) où le caillot est une masse de sang qui se solidifie. Plusieurs facteurs influencent l'apparition de TVP, comme par exemple le fait d'avoir peu d'activités physiques [3]. Ce paramètre explique pourquoi elles sont plus présentes chez les personnes âgées, qui ont souvent plus de mal à se déplacer qu'une personne plus jeune. Des lésions endothéliales, par exemple dues à la chirurgie peuvent également être la cause d'une thrombose.

Une fois qu'un caillot se forme dans la jambe, il peut facilement migrer vers d'autres parties du corps, comme les poumons. Lorsqu'il atteint la circulation pulmonaire, aussi appelée petite circulation, c'est la respiration qui est impactée [4].

Lorsqu'il n'y a pas d'embolie, le fonctionnement du coeur et des poumons est le suivant, les poumons reçoivent du sang provenant du côté droit du cœur. Une fois le sang oxygéné, il repart vers le côté gauche du cœur qui renvoie le sang dans le reste du corps. Cependant, en cas d'embolie, un caillot est présent dans les poumons. Cela implique que toute leur capacité n'est pas utilisée et le sang est moins bien oxygéné. Une mauvaise oxygénation du sang peut avoir des conséquences plus graves. Tout ce mécanisme est illustré à la figure 1.1.

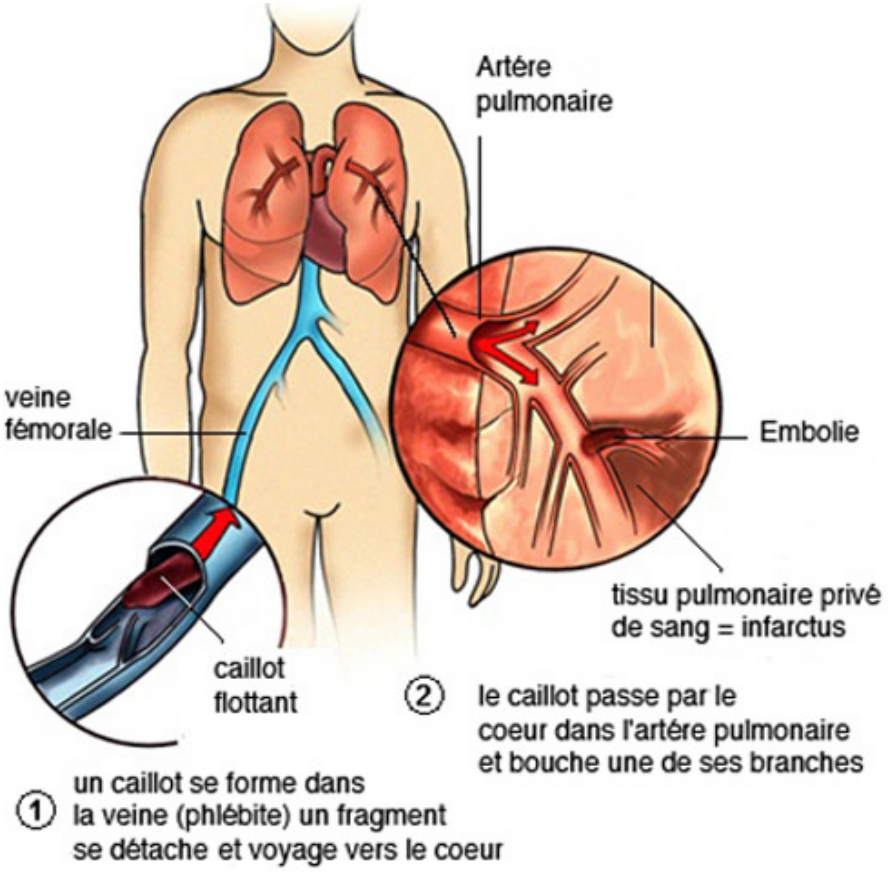


Figure 1.1 – Mécanisme de l'embolie pulmonaire. L'image dans le cercle en bas à gauche montre la thrombose veineuse profonde qui peut remonter jusqu'aux poumons et par conséquent former une embolie. L'image dans le cercle à droite illustre le caillot dans les poumons. Ce schéma provient de [5].

Bien qu'une embolie pulmonaire puisse causer la mort, elle peut souvent être traitée si le diagnostic et les soins sont effectués suffisamment tôt. Les embolies pulmonaires peuvent également être le point de départ de problèmes médicaux futurs ou aggraver des problèmes existants. Ces conséquences peuvent être un arrêt cardiaque, un épanchement pleural, une arythmie ou encore une hypertension pulmonaire.

Une étude menée aux États-Unis a révélé un taux de mortalité à 30 jours, toutes causes confondues, de 5,4 % chez les patients ayant reçu un diagnostic d'embolie pulmonaire [6]. Des recherches ont montré que certains paramètres mesurés lors d'une embolie peuvent être des indicateurs de risque à court terme comme la dilatation des ventricules.

L'embolie pulmonaire aiguë se traduit par l'obstruction de minimum deux artères lobaires, qui sont des divisions de l'artère pulmonaire. Cette obstruction induit une résistance au passage du sang et donc une augmentation de la pression qui provoque la dilatation du ventricule droit. Cette dilatation induit quant à elle la réduction dans les même proportions du ventricule gauche [7].

Il est donc intéressant de pouvoir évaluer ces paramètres lors de l'embolie, par exemple, à l'aide des images du coeur qui doivent être prises pour faire le diagnostic de l'embolie. Pour résumer, une embolie pulmonaire peut avoir de graves conséquences dont la mort, notamment si sa gravité est mal évaluée.

1.2 Objectif du mémoire

L'objectif de ce mémoire est de mettre au point un outil fiable pour aider les médecins à faire une stratification de la gravité de l'embolie. La motivation principale étant que les patients reçoivent le traitement et le suivi les plus adaptés à leur cas.

Pour atteindre cet objectif, différentes parties sont abordées. La première consiste à produire un modèle capable de segmenter les différentes cavités du coeur grâce aux techniques d'intelligence artificielle. La littérature montre différentes techniques pour arriver à ce résultat. Pour la plupart d'entre-elles, c'est l'apprentissage profond, en anglais *Deep Learning*, qui est utilisé et plus particulièrement les réseaux de neurones convolutifs. Ces différentes méthodes seront présentées dans la suite de ce travail. Pour celui-ci, nous avons implémenté un U-Net en trois dimensions. Afin de réaliser la segmentation, des images médicales sont nécessaires. Celles utilisées dans ce mémoire ont été prises par tomodensitométrie, en anglais *Computed Tomography (CT)*.

La seconde partie, plus courte, traitera de l'évaluation du modèle sur les données reçues aux *Cliniques universitaires Saint Luc*. En effet, la base de données utilisées pour entraîner le modèle diffère de celle sur laquelle nous voulons effectuer la segmentation car les machines et paramètres d'acquisition ne sont pas les mêmes dans les deux cas. En effet, nos données d'entraînement sont plus centrées sur le coeur. Nous pouvons également souligner des différences de statistiques dans l'intensité des pixels des images. Les résultats obtenus pour cette partie ne peuvent pas être quantifiés, ils sont simplement évalués sur base visuelle.

1.3 Structure du mémoire

Ce travail de fin d'études est organisé comme suit. Le chapitre 1 constitue l'introduction faite ci-dessus. Le chapitre 2 présente les bases théoriques relatives à la prise d'images par tomodensitométrie. Ensuite, c'est la partie théorique concernant l'intelligence artificielle qui est présentée dans le chapitre 3. Le chapitre 4 relate l'état de l'art concernant la segmentation d'images médicales et plus précisément de la segmentation des cavités cardiaques en reprenant différents articles pertinents. Le chapitre 5 définit le matériel utilisé pour créer le modèle. Il détaille également la méthode suivie. Pour finir, il présente les résultats obtenus ainsi que leur analyse. Le chapitre 6 explicite la manière dont nous avons évalué notre modèle sur les données à segmenter. Il met également en avant les différences entre les deux bases de données utilisées. La fin de ce chapitre montre les derniers résultats obtenus en utilisant la meilleure version de notre modèle. Finalement, le chapitre 7 envisage certaines pistes d'amélioration et le chapitre 8 permet de conclure ce travail.

Chapitre 2

Imagerie médicale par rayons X

Afin de réaliser ce travail, des images médicales du coeur sont nécessaires. De manière générale, les images médicales servent à observer ce qui se trouve à l'intérieur du corps comme les organes ou les os. Cela permet d'établir des diagnostics pour toutes sortes de pathologies, y compris les embolies. Les images utilisées dans ce mémoire sont prises par tomodensitométrie. Ce type d'imagerie médicale se base sur des techniques de radiographie. Afin de mieux comprendre comment les images nécessaires sont récoltées, les sections suivantes expliquent la base de la radiographie et comment cette méthode est exploitée pour la tomodensitométrie.

2.1 Radiographie

La radiographie conventionnelle consiste à envoyer des rayons X vers un patient. L'énergie des rayons X est atténuée en passant au travers des tissus du corps. L'intensité de l'atténuation dépend de la profondeur et de la densité des tissus traversés. Au plus le tissu est dense et profond, au plus l'intensité diminue, et inversement, comme le montre la figure 2.1.

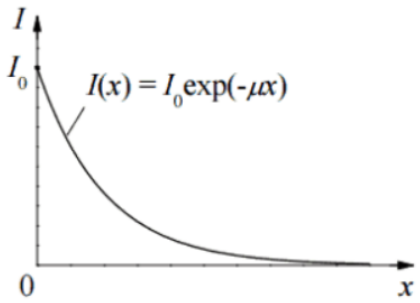


Figure 2.1 – Évolution de l'intensité des rayons X en fonction de la profondeur et de la densité du tissu. L'atténuation se fait de manière exponentielle en fonction de la profondeur parcourue par le rayon X où x est la profondeur du tissu et I_0 est l'intensité initiale du rayon X [8].

L'évolution se fait de manière exponentielle suivant la loi de Lambert-Beer représentée par l'équation 2.1 où I_0 est l'énergie initiale du rayon X, I(x) est l'intensité d'énergie restante à une distance x de la source de rayons, μ est un coefficient d'atténuation linéaire [9].

$$I(x) = I_0 \ \ e^{-\mu x} \tag{2.1}$$

Pour connaître l'intensité finale, un film est placé à l'opposé du rayon X. L'image à gauche de la figure 2.2 montre le rayon X au-dessus qui traverse le patient et arrive sur le film détecteur en dessous. L'image à droite de la figure 2.2 montre que l'image finale obtenue est en deux dimensions bien que le corps observé soit en trois dimensions. En réalité, ces trois dimensions sont comprimées en deux dimensions selon l'axe des rayons X. Cela a pour conséquence que certains tissus, pourtant différents, se retrouvent superposés sur l'image produite. Un exemple facilement observable est la superposition des côtes, qui sont d'un gris assez clair sur l'image, et des poumons, qui sont presque noirs. Ceci représente un inconvénient car il y a une diminution de la visibilité d'un organe d'intérêt, les poumons.

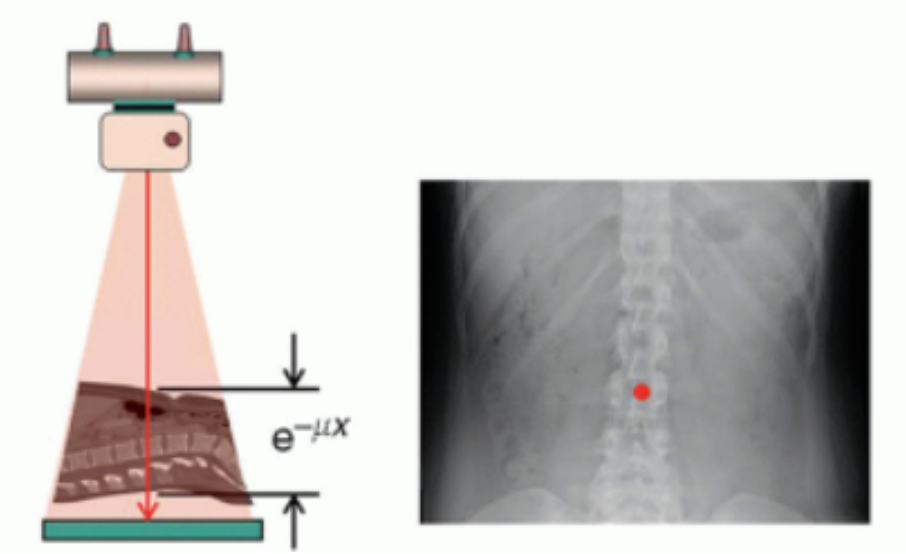


FIGURE 2.2 – Principe de la radiographie. L'image de gauche illustre les rayons X qui arrivent sur le patient ainsi que le film récepteur. L'image de droite montre le résultat obtenu [10].

Malheureusement, les images en deux dimensions peuvent empêcher le diagnostic de certaines pathologies, elles présentent également une contrainte lorsque certains paramètres particuliers, comme le volume des organes, doivent être étudiés. Dans ce cas, des images en trois dimensions sont nécessaires [11].

2.2 Principe de la tomodensitométrie

Comme pour la radiographie, la tomodensitométrie est réalisée grâce à une source de rayons X ainsi qu'un film de réception. Une table rotative est également utilisée pour collecter une multitude d'images en deux dimensions. Ces images sont acquises avec un angle différent afin d'observer le corps sous tous les points de vue. Ensuite, une unité de traitement se charge de la reconstruction pour obtenir des images en trois dimensions sur base de celles en deux dimensions. Cette technique est illustrée à la figure 2.3.

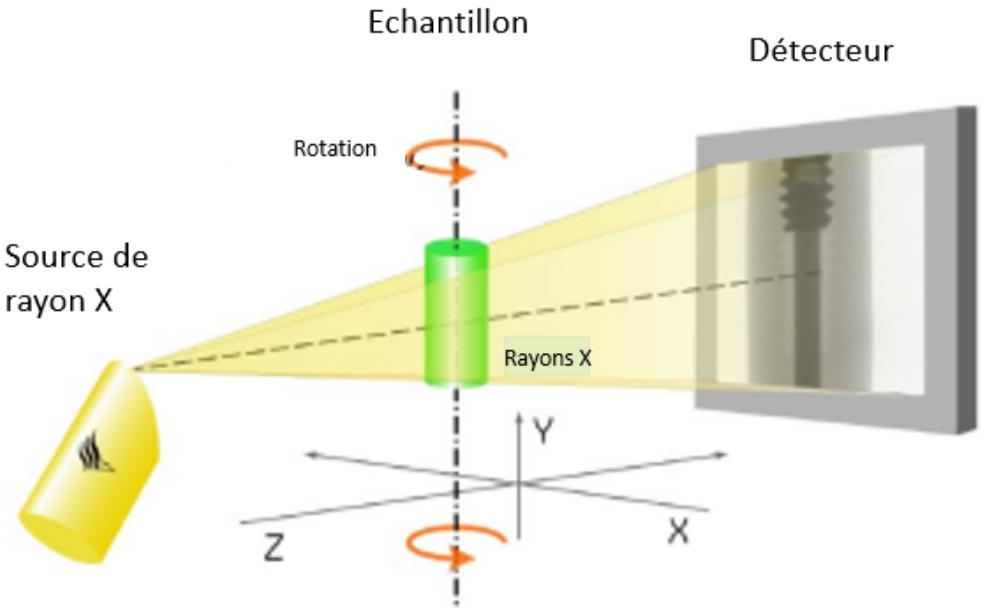


FIGURE 2.3 – Principe de la tomodensitométrie. L'image illustre la source de rayons X en jaune, celle-ci traverse le patient, représenté en vert. Ensuite, le détecteur capte les rayons X de l'autre coté du patient et recompose une image en fonction de l'atténuation de l'énergie du rayon. Le principe de rotation du système est représenté par les deux flèches orange. L'image est extraite et traduite depuis [12].

Avant d'arriver aux techniques actuelles de tomodensitométrie, plusieurs générations ont été testées [13]. La partie théorique concernant cette évolution a été reprise du cours d'imagerie médicale donnée à l'UCLouvain par Greet Kerckhofs [14].

2.2.1 Première génération

La première génération de scanner ne possède qu'une seule source de rayons X sous forme d'un faisceau fin. Celui-ci traverse le corps du patient et n'est détecté que sur un seul film détecteur. Dans cette configuration, ces deux éléments bougent simultanément pour balayer toute la largeur du corps. Ensuite, des rotations sont effectuées afin de parcourir tous les angles de vue. La simplicité de ce processus peut être mise en avant, par contre il présente l'inconvénient d'être assez lent. Un schéma illustre ce système à la figure 2.4.

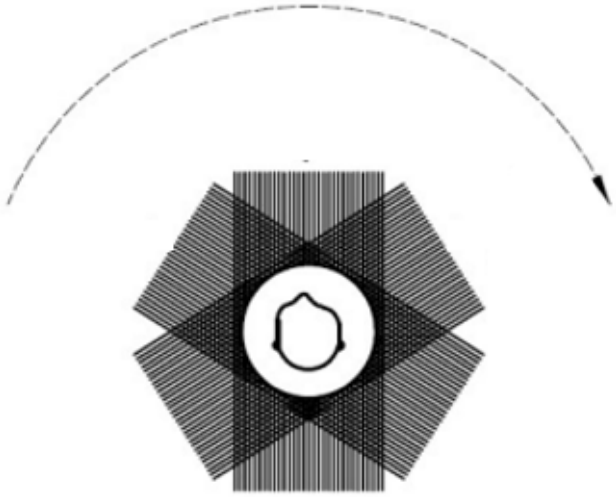


Figure 2.4 – Système de première génération. Le fin faisceau est représenté par une ligne noire. Le balayage des rayons X est montré grâce à la translation des lignes noires. Le fait d'avoir différents groupes de faisceaux avec un angle différent montre la rotation effectuée. Image issue de [14].

2.2.2 Seconde génération

Les systèmes de seconde génération utilisent le même principe de balayage et de rotation que celui de la génération précédente. Cependant, le faisceau n'est plus étroit mais en éventail. Dans ce cas, plusieurs détecteurs sont nécessaires. Cela permet de réduire le nombre de balayages puisqu'une partie plus large peut être ciblée en une seule fois, ce qui représente un gain de temps par rapport à la première génération. Cette technique est montrée à la figure 2.5.

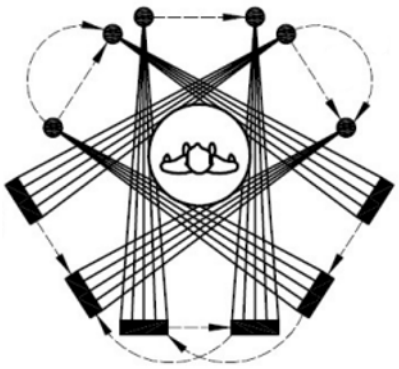


Figure 2.5 – Système de seconde génération. Celui-ci est composé d'un faisceau en éventail, on peut voir que différents rayons, lignes noires, partent d'un même point, une même source. Le balayage est représenté par la translation de la source et ces faisceaux. La rotation est illustrée par les différents angles de départ de la source. Plusieurs détecteurs sont nécessaires, un pour chaque rayon de l'éventail. Image issue de [14].

2.2.3 Troisième génération

Pour le scanner de troisième génération, toute la largeur du corps est couverte et détectée à chaque rotation. Les détecteurs continuent de tourner en même temps que le faisceau. Une fois de plus, un gain de temps peut être constaté en comparaison à la génération précédente. Le système est visible à la figure 2.6.

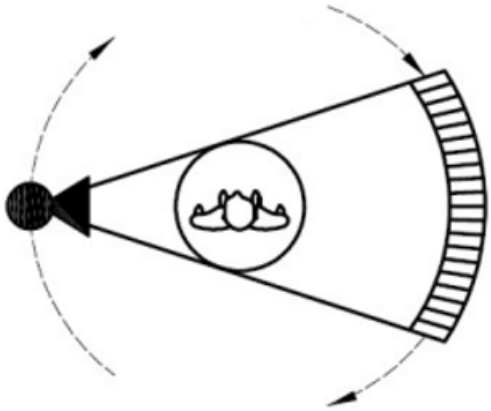


FIGURE 2.6 – Système de troisième génération. La source en éventail permet de couvrir toute la largeur du corps, plusieurs détecteurs sont utilisés, ce sont les rectangles disposés en arc de cercle sur la partie droite de l'image. Image issue de [14].

2.2.4 Quatrième génération

Ce système est également composé d'une source de rayons X qui tourne autour du patient. Ce qui diffère de la technique précédente est que les détecteurs ne font plus partie de la rotation. Il y en a tout autour du patient, leur nombre correspondant au nombre de vues prises pour réaliser une image complète. Cela est illustré à la figure 2.7.

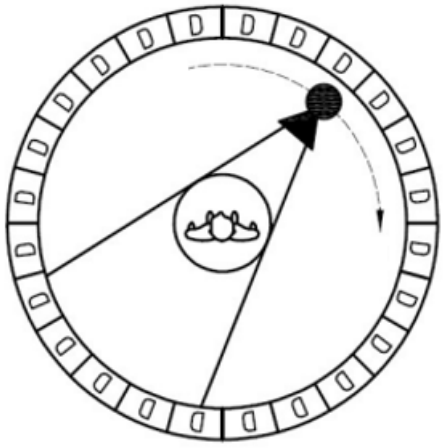


FIGURE 2.7 – Système de quatrième génération. La source de rayons X est représentée par le point noir, sa rotation est visible grâce à la flèche. Les détecteurs sont fixes et partout autour du patient, ils sont montrés grâce au rectangle avec la lettre D à l'intérieur. Image issue de [14].

2.2.5 Cinquième génération

La différence majeure entre ce système et ceux présentés ci-dessus est l'absence d'un mécanisme de rotation. En effet, plusieurs sources de rayons X sont présentes et disposées de manière circulaire autour du patient. Elles sont activées et désactivées les unes après les autres. Cela permet de retirer tous les inconvénients engendrés par la rotation. Cette technique de cinquième génération est illustrée à la figure 2.8.

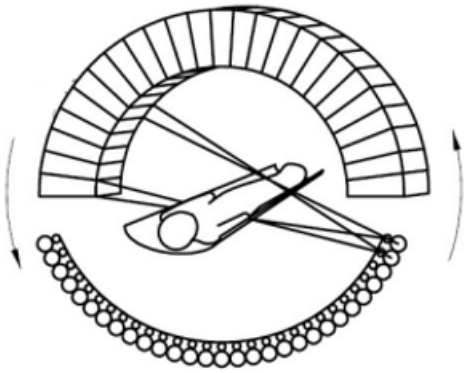


Figure 2.8 – Système de cinquième génération. Tout le mécanisme de ce système reste fixe. Les sources et les détecteurs sont disposés de manière circulaire autour du patient, ils sont représentés respectivement par les ronds et les rectangles. Image issue de [14].

2.2.6 Sixième génération

Cette technique comprend un tube à rayons X qui fait le tour du patient, donc en forme de cercle et le patient subit une translation, la table sur laquelle il se trouve avançant vers le cercle formé de rayons X. La conséquence de ceci est que la trajectoire décrite par les rayons X par rapport au patient est hélicoïdale. Cela est visible sur la figure 2.9.

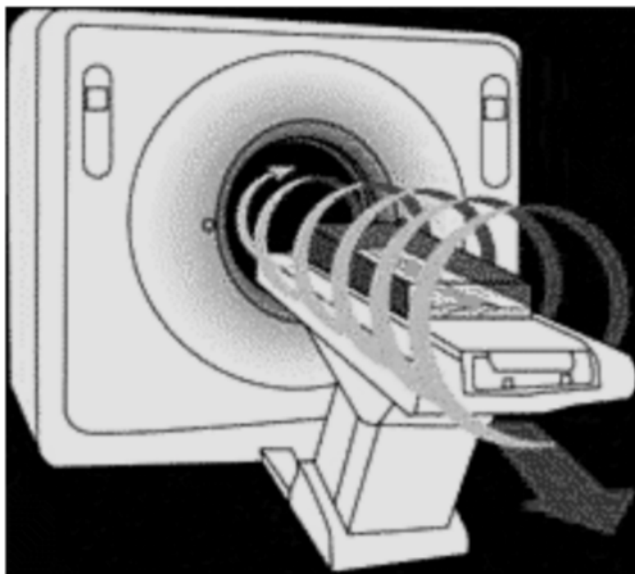


Figure 2.9 – Système de sixième génération. L'image représente la partie extérieur de ce système. La flèche décrit la trajectoire hélicoïdale des rayons X par rapport au patient. Image issue de [14].

2.3 Modèle de tomodensitométrie utilisé pour récolter les données de ce mémoire

Le modèle utilisé pour récolter une partie des images exploitées dans ce mémoire est un Phillips IQon qui se trouve aux *cliniques universitaires Saint Luc*. La structure de ce modèle est semblable à celle de sixième génération. En effet, le patient subit une translation grâce à la table sur laquelle il est placé, comme on peut le voir sur la figure 2.10. Les rayons X se trouvent autour de la table afin de prendre des images du patient sous tous les angles. Ce type d'appareil permet une précision de l'image de ± 1 mm dans le plan du portique et de ± 5 mm dans les directions X et Y [15].

FIGURE 2.10 – Image du Phillips IQon issue du site de Phillips [15]. Le patient subit une translation vers le cercle décrit par les rayons X.

Les images obtenues grâce à la tomodensitométrie sont en trois dimensions, plus précisément $512 \times 512 \times C$ où C peut avoir une valeur différente en fonction du patient. Pour visualiser une image en trois dimensions, le plus simple est de l'observer en deux dimensions selon trois coupes différentes. Un rappel des différentes coupes est disponible à la figure 2.11.

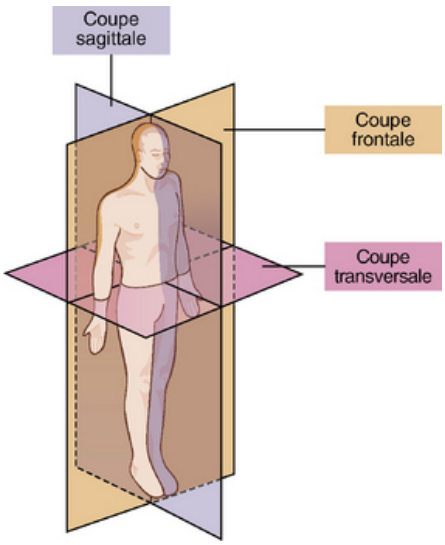
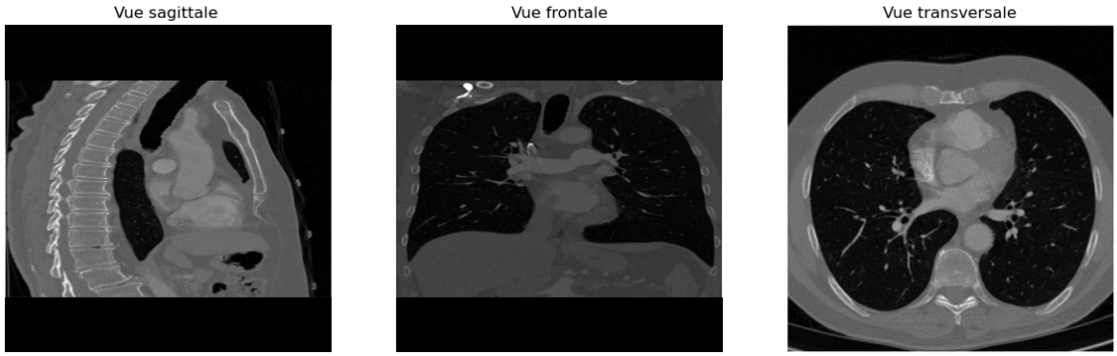
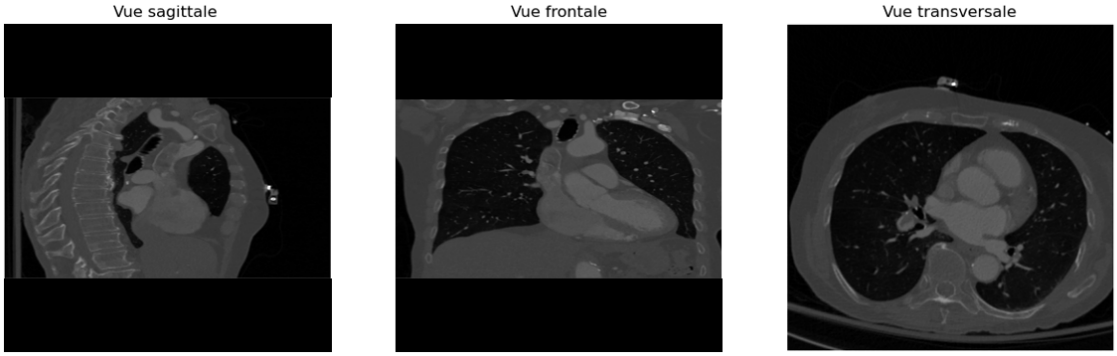


FIGURE 2.11 – Cette figure montre les différentes coupes possible du corps : la vue sagittale en bleu, frontale en orange et transversale en rose. L'image est issue de [16].

Deux exemples de ce type d'image se trouvent à la figure 2.12, nous avons recadré les images par soucis d'homogénéité lors de l'affichage des différents patients. Elles ont dès lors une dimension de $512 \times 512 \times 256$. Dans les deux exemples, les images en vue sagittale et frontale ont une dimension de 512×256, alors que la vue transversale est de dimension 512×512. Ces images permettent d'observer la diversité des données. Le niveau de gris n'est pas toujours le même, l'orientation peut être différente, cela est visible si l'on compare les vues transversales des deux patients.



(a) Patient 1



(b) Patient 2

FIGURE 2.12 – Exemple d'images de deux patients. Les vues sagittale et frontale ont une dimension de 512 × 256 tandis que la vue transversale a une dimension de 512 × 512. Images tirées de la base de données reçues des cliniques universitaires Saint Luc.

2.4 Résumé

Ce chapitre permet de comprendre comment les données exploitées dans ce travail ont été acquises, c'est-à-dire par tomodensitométrie.

La radiographie se base sur l'utilisation de rayons X qui passent au travers des patients. L'atténuation de l'intensité des rayons se fait en fonction de la densité ainsi que la profondeur des tissus. C'est grâce à cela que nous pouvons différencier les différentes parties du corps sur les images.

La tomodensitométrie utilise le principe de radiographie mais pour obtenir des images reconstruites en trois dimensions. Plusieurs générations d'appareil de tomodensitométrie ont été créées avec différents types de rotation avant d'arriver à la génération actuellement utilisée dans les hôpitaux.

Chapitre 3

Apprentissage profond

Les techniques d'imagerie médicale comme la tomodensitométrie, vue au chapitre 2 sont utilisées pour la détection, le diagnostic ainsi que l'évaluation de la réponse au traitement de maladies. Ces interprétations sont faites par des experts humains, typiquement des radiologues. Cependant, des erreurs peuvent être commises dues à la fatigue ou à la grande variété de pathologies. Il faut également soulever le fait que si deux médecins donnent une segmentation d'un coeur sur base d'une même image, les deux résultats sont différents. Si on utilise l'intelligence artificielle, celle-ci permet une généralisation des résultats et donne toujours le même résultat si le même modèle doit évaluer plusieurs fois la même image. L'assistance aux radiologues a évolué au cours du temps, ils ont d'abord bénéficié d'une assistance par ordinateur afin d'avoir une meilleure visualisation des images grâce à un pré-traitement de celles-ci. Ensuite, ce fut au tour de l'apprentissage profond , *Deep Learning* en anglais, de venir en aide à ces experts en permettant une automatisation des tâches [17]. L'apprentissage profond étant une sous-catégorie de l'apprentissage automatique.

Dans ce chapitre, nous faisons un rappel sur l'apprentissage automatique et les différentes méthodes d'apprentissage possibles. Ensuite, notre intérêt se portera sur les techniques d'apprentissage profond dédiées aux images médicales ainsi que la manière dont les réseaux de neurones sont entraînés.

3.1 Apprentissage automatique

Appelé *Machine Learning* en anglais, l'apprentissage automatique est le coeur de ce travail. Au cours de ces dernières années, celui-ci a connu une grande évolution. Son application s'étend sur divers domaines, comme la reconnaissance visuelle d'objets, la reconnaissance vocale ou encore la détection d'objets. Tout comme l'apprentissage automatique, l'apprentissage profond a subi un essor dans le domaine de l'imagerie médicale [18]. Peu importe le domaine d'utilisation, le principe est d'entraîner un modèle à réaliser une tâche précise. Pour entraîner un modèle et permettre sa généralisation à des données qui lui sont inconnues, trois jeux de données sont nécessaires. Le premier, appelé **ensemble d'entraînement**, permet d'entraîner le modèle à réaliser la tâche. Le deuxième, l'**ensemble de validation**, sert à valider le modèle, c'est-à-dire vérifier si la tâche demandée est effectuée correctement et d'adapter les différents hyper-paramètres en fonction des résultats obtenus. Et pour finir l'**ensemble de test** est uniquement utilisé pour évaluer le modèle final obtenu [19].

3.1.1 Méthodes d'apprentissage

Un modèle peut être entraîné de différentes manières, nous pouvons citer l'apprentissage supervisé, l'apprentissage non-supervisé, l'apprentissage semi-supervisé ou encore l'apprentissage auto-supervisé.

L'**apprentissage supervisé**, en anglais *Supervised Learning*, se base sur l'utilisation de données annotées, c'est-à-dire que toute donnée d'entrée est associée à la sortie désirée du modèle. Si de telles données sont disponibles, le modèle va pouvoir apprendre avec un principe de superviseur qui montre au modèle le résultat attendu. Ce type d'apprentissage est généralement utilisé pour des problèmes de segmentation ou de classification. Dans ce cas, chaque donnée contient les différentes étiquettes associées aux différentes classes disponibles dans cette donnée. Dans un premier temps, le modèle subit une phase d'entraînement pendant laquelle il apprend à associer une étiquette à une classe. Dans un second temps, il sera utilisé sur des données non annotées que l'on veut classer ou segmenter [20]. C'est grâce à cette méthode que l'apprentissage profond a réellement évolué en permettant d'entraîner des modèles performants. Le désavantage d'utiliser des données annotées est qu'elles ont un coût correspondant au temps de faire les annotations sur chaque donnée. Cela veut dire que ce type de données est parfois présent mais en quantité insuffisante. C'est ce type d'apprentissage qui est utilisé dans ce mémoire.

L'**apprentissage non-supervisé**, en anglais *Unsupervised Learning*, correspond à l'opposé de l'apprentissage supervisé. En effet, le modèle doit apprendre sans avoir de sortie spécifiquement définie. L'objectif de ce genre de modèle est plutôt de faire ressortir la structure statistique de la base de données proposée en entrée [21].

Pour comparer ces deux techniques avec un exemple, prenons des images de foie et des images de coeur. Pour l'apprentissage supervisé, en entrée le modèle va recevoir différentes images pour lesquelles il sait si c'est un foie ou un coeur. Après l'entraînement, il sera capable de dire si une image donnée est un foie ou un coeur. Dans le cas de l'apprentissage non-supervisé, le modèle va uniquement recevoir les images de foie et de coeur, sur base de celles-ci, il va créer différents groupes, en anglais *clusters*, par rapport aux caractéristiques communes entre les images.

L'**apprentissage semi-supervisé**, en anglais *Semi-Supervised Learning*, se situe entre l'apprentissage supervisé et le non-supervisé. Il s'agit d'une technique assez récente qui permet de réduire le nombre de données annotées nécessaires. Pour ce type d'apprentissage, nous disposons de peu de données annotées et de beaucoup de données non-annotées. Le modèle est entraîné une première fois avec les données annotées. Celui-ci va être utilisé pour créer des annotations sur les données non-annotées. Ensuite, le modèle va être entraîné une seconde fois mais avec l'ensemble des données, ce qui devrait améliorer ses performances [22] [23].

L'**apprentissage auto-supervisé**, en anglais *Self-supervised Learning*, utilise également des données non-annotées pour pallier au manque de données annotées. Il s'agit en réalité d'une catégorie d'apprentissage non-supervisé. Sur base des données non-annotées, des pseudo-annotations sont créées et le modèle apprend à prédire ses pseudo-annotations. Ce type d'apprentissage crée des modèles généraux qui peuvent ensuite être peaufinés grâce à une faible quantité de données annotées [24].

3.2 Réseau de neurones pour la segmentation d'images médicales

Un des objectifs de ce mémoire est de pouvoir effectuer une segmentation sur base d'images médicales. Une première recherche a été réalisée sur les techniques de segmentation générales afin d'obtenir une vue d'ensemble sur ce qui a déjà été testé.

La vision par ordinateur a été la première aide mise en place pour assister les médecins dans leur prise de décisions (diagnostic, traitement). En effet, cette technique utilise du pré-traitement d'images comme une augmentation de contraste, un filtre pour le bruit mais également de la reconstruction en trois dimensions, tout cela offrant une meilleure visualisation des détails sur l'image. Sur base de celle-ci, les médecins peuvent reconnaître plus facilement les organes et ainsi les segmenter manuellement [25]. L'apprentissage profond va permettre d'aller encore plus loin en réalisant cette segmentation de manière automatique.

3.2.1 Réseau de neurones artificiels

La première architecture développée est le réseau de neurones artificiels, en anglais *Artificial Neural Network*. Celui-ci est inspiré des modèles biologiques, ce qui lui permet de dépasser les performances de ce qui avait été fait avant. Ce type d'architecture se compose de neurones, c'est-à-dire des noeuds de calculs liés les uns aux autres comme le montre la figure 3.1.

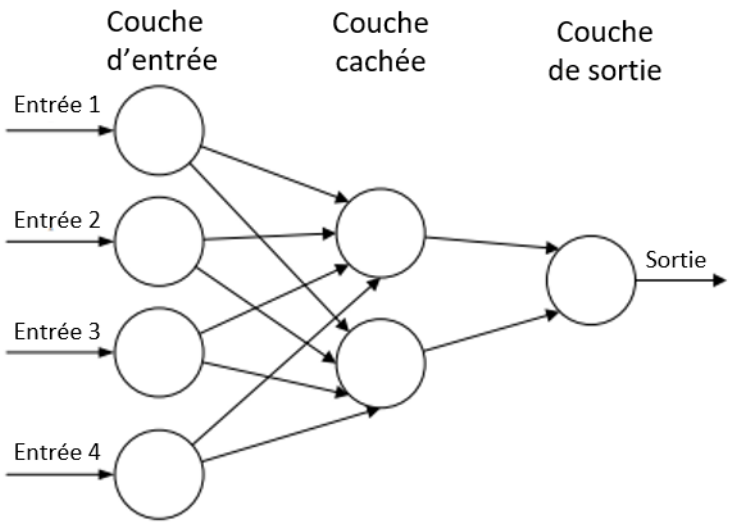


Figure 3.1 – Structure du réseau de neurones artificiels avec 3 couches. La couche d'entrée est la couche à laquelle les données d'entrée, ici entrée 1 à 4, sont fournies. La couche cachée est la couche d'apprentissage, il peut y avoir un nombre variable de ce type de couche. La couche de sortie permet de récupérer le résultat. Les couches sont composées de neurones, tous les neurones de chaque couche étant reliés aux neurones de la couche précédente et de la couche suivante. L'image est issue de [26] et a été traduite en français.

Le réseau est composé de différentes couches, la première étant la couche d'entrée à laquelle les données sont fournies, généralement sous forme de vecteur multidimensionnel. Ensuite, cette couche d'entrée va distribuer l'information aux couches cachées de la structure. C'est dans ces couches que l'apprentissage a lieu, chaque couche se base sur le résultat de la couche précédente afin d'apporter les modifications nécessaires à l'amélioration du résultat en sortie. L'apprentissage profond est caractérisé par cette suite de couches [26].

Une des architectures les plus reconnues pour traiter des images est le réseau de neurones convolutif, en anglais *Convolutionnal Neural Network (CNN)*. Il possède une architecture simple mais précise qui permet de réaliser des tâches complexes comme la reconnaissance de formes.

3.2.2 Réseau de neurones convolutif

Dans les années 90, Yann Le Cun s'intéresse à la reconnaissance d'images et crée un réseau neuronal convolutif [27]. Puisque ce type d'architecture est adapté au traitement des images, il est étudié plus en détails dans cette section [28].

Architecture

En ce qui concerne l'architecture globale, celle-ci présente différentes couches, les plus importantes étant les couches convolutives. On y trouve aussi des couches de regroupement et des couches entièrement connectées. Ceci est illustré à la figure 3.2.

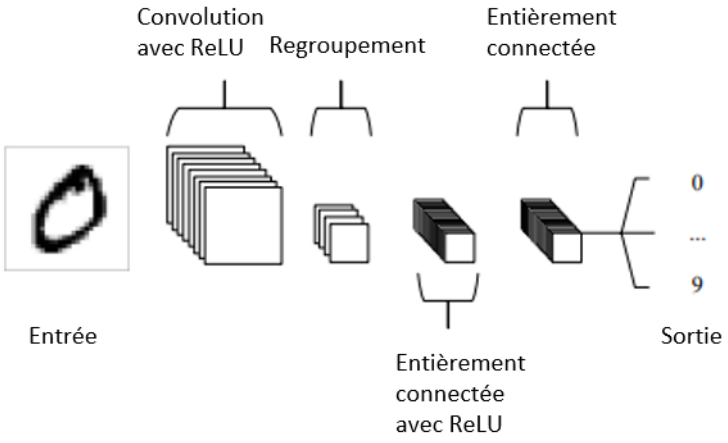


Figure 3.2 – Architecture d'un réseau de neurones convolutif avec 5 couches. Les trois types de couches de ce type d'architecture sont représentés : les couches convolutives étant les plus importantes. Il y a également les couches de regroupement et les couches entièrement connectées. L'image est issue de [26] et a été traduite en français.

La couche d'entrée va contenir les données d'entrée, à savoir dans notre cas, les valeurs des pixels de l'image. Le principe de cette architecture est d'extraire des caractéristiques spécifiques à l'aide des couches de convolution, ce qui permet de reconnaître ce que l'on veut identifier sur une image. Ainsi, à partir d'une image, ce type de réseau calculera plusieurs convolutions afin d'en retirer un maximum de spécificités.

Couches convolutives

La couche convolutive est la couche la plus importante d'une architecture basée sur la convolution. Elle implique l'utilisation d'un noyau, en anglais *kernel*. Celui-ci est en réalité une matrice de dimension relativement petite par rapport à celle de l'entrée. Le rôle de cette couche est d'effectuer une convolution entre l'entrée de la couche et les différents noyaux pour produire des cartes d'activation. Chaque entrée subit différentes convolutions avec tous les noyaux. Le rôle de chaque noyau est de mettre en avant une caractéristique spécifique de l'image. Ces caractéristiques doivent aider à la segmentation pour différencier les objets. Chaque carte d'activation représente une caractéristique que le modèle doit apprendre à identifier. Un exemple illustre cela à la figure 3.3. On peut y voir qu'une partie de l'entrée, de la taille du noyau, est considérée pour effectuer une convolution avec le noyau. La valeur de sortie correspondante est calculée grâce à l'équation 3.1, représentant une case de la carte d'activation [29].

$$Sortie(i,j) = \sum_{i}^{n}\sum_{j}^{n} Entrée(i,j) * Noyau(i,j) \tag{3.1}$$

Dans notre cas, n = 2 avec i et j commençant à 0. La valeur de 16 est obtenue comme suit : $9 \cdot 0 + 4 \cdot 2 + 1 \cdot 1 + 1 \cdot 4 + 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 + 0 \cdot 2 + 1 \cdot 1 = 16$.

A la fin de la convolution, toutes les cases de la sortie doivent être remplies. Pour atteindre ce résultat, le noyau doit effectuer des translations dans l'entrée afin que celle-ci soit complètement couverte.

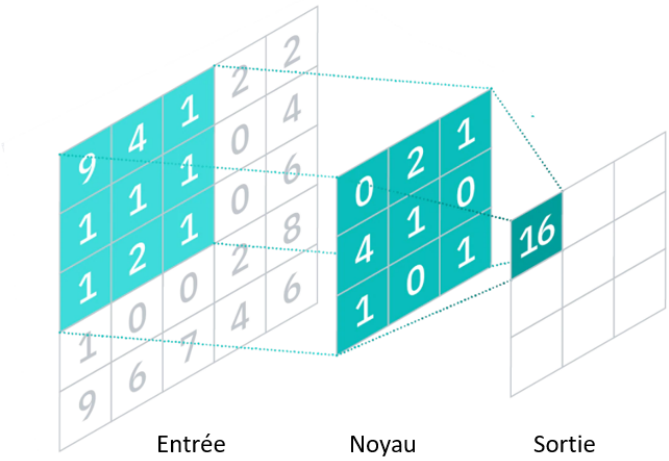


FIGURE 3.3 – Exemple de convolution. Une convolution est réalisée entre une partie de l'entrée et le noyau afin de former une valeur dans la carte d'activation qui est la sortie. L'image est issue de [30] et a été traduite en français.

Le résultat de la convolution va dépendre de trois paramètres : la profondeur, l'enjambement et le remplissage avec des zéros , respectivement *Depth, Stride et Zero-padding* en anglais.

La **profondeur** de la sortie peut varier en fonction du nombre de neurones dans chaque couche, pour une même entrée. Pour une profondeur plus faible, moins de neurones sont nécessaires mais cela implique également une réduction de la performance du modèle dans la reconnaissance de formes par exemple.

L'**enjambement** permet de déterminer le degré de chevauchement. Il quantifie le pas de translation du noyau horizontalement. Pour un enjambement de 1, il y a un grand chevauchement car le pas de translation est plus petit que la dimension du noyau. Plus l'enjambement est faible, plus la dimension de la carte d'activation est grande et a un impact plus important. Au contraire, plus il est important, plus le résultat de la convolution est de faible dimension. La figure 3.4 présente un enjambement de 1.

Enfin, le **remplissage avec des zéros** consiste en la mise en place d'un contour de zéros, comme cela est visible à la figure 3.4. Cette technique permet de contrer l'effet de réduction de dimensions dû à la taille du noyau. Si la taille du noyau est supérieure à 1, il existe un effet de bord qui induit une réduction de dimensions. Le remplissage avec des zéros permet de contrer cet effet et sa largeur dépend de la réduction de dimension.

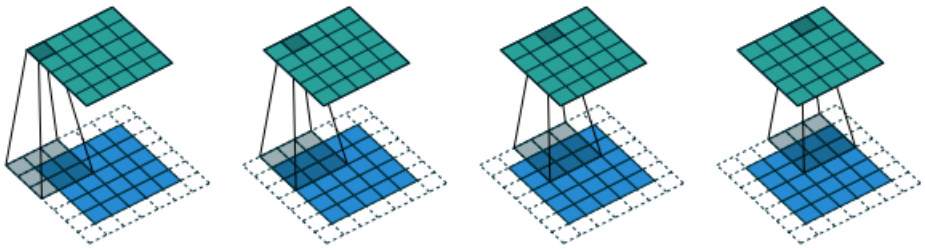


FIGURE 3.4 – Principe de remplissage avec des zéros. Une bordure de zéros est mise en place autour de l'image afin de limiter la réduction de dimension induite par l'effet de bord dû à la taille du noyau [26].

Finalement, les couches linéaires, comme par exemple les couches convolutives, sont accompagnées d'une couche non linéaire comme une couche de rectification linéaire unitaire (ReLU). Celle-ci consiste à appliquer une fonction d'activation telle qu'une sigmoïde à la sortie de la couche précédente. En effet, il n'existe pas nécessairement de relations linéaires entre les données de l'image et sa classification ou segmentation. C'est pourquoi une fonction d'activation non-linéaire est nécessaire car, sans elle, le réseau serait réduit à des classifications linéaires.

Couches de regroupement

Appelées *Pooling Layers* en anglais, les couches de regroupement permettent de diminuer le nombre de paramètres nécessaires afin de décrire les couches les plus profondes du réseau. En conséquence, la dimension des cartes d'activation est réduite. Deux techniques différentes sont distinguables.

Une **couche de regroupement basée sur le maximum** consiste à diviser la carte d'activation en un certain nombre de groupes, souvent de taille 2×2. Pour chacun de ces groupes, seule la valeur maximale est conservée. Un exemple est illustré à la figure 3.5.

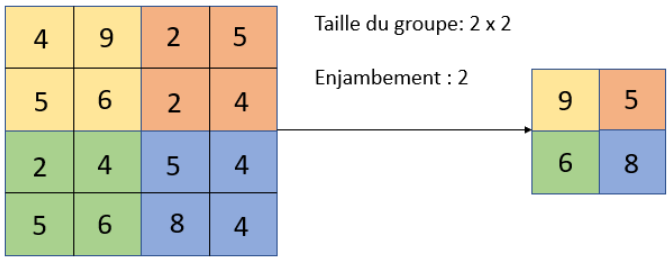


FIGURE 3.5 – Exemple de couches de regroupement basée sur le maximum. Pour chaque groupe 2 × 2, la valeur maximale est gardée. Image inspirée de [31].

La **couche de regroupement basée sur la moyenne** se base sur le même principe. Cependant, dans ce cas, c'est la moyenne de toutes les valeurs qui est prise. Un exemple est disponible à la figure 3.6.

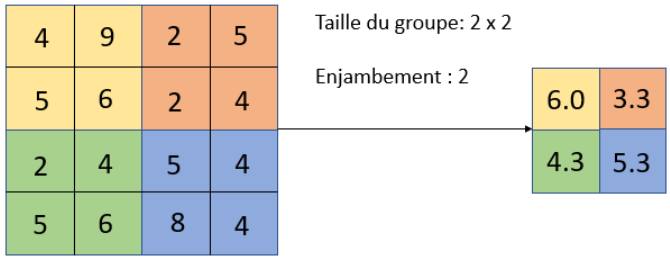


FIGURE 3.6 – Exemple de couches de regroupement basée sur la moyenne. Pour chaque groupe 2 × 2, la valeur moyenne est gardée. Image inspirée de [31].

Couches entièrement connectées

Les neurones de ce type de couches sont directement connectés aux neurones de la couche précédente et de la couche suivante. En effet, la première couche reçoit des données en entrée et celle-ci produit différents résultats qui correspondent aux entrées des neurones de la couche suivante et ainsi de suite.

Design du réseau

Le design du réseau dépend principalement de la tâche à effectuer. En effet, un réseau, qui doit classer ou segmenter un seul objet, pourra être plus petit en terme de couches, aura moins de caractéristiques à identifier ainsi que moins de paramètres à calculer qu'un réseau qui doit classer ou segmenter plusieurs objets différents. La difficulté est de trouver le bon nombre de couches sachant qu'il n'existe pas de relation linéaire entre le nombre de classes et le nombre de couches nécessaires pour réaliser correctement la tâche.

Le design du réseau comprend également le choix des hyper-paramètres comme la vitesse d'apprentissage, le nombre d'epochs ou encore la taille des groupes. Ceux-ci sont définis à la section 3.3.7.

Bien que l'utilisation d'un réseau de neurones convolutif semble prometteuse et qu'elle ait déjà prouvé son efficacité, elle présente plusieurs désavantages comme le fait d'avoir des couches entièrement connectées mais également le fait d'avoir une perte de détails lors des couches de regroupement. Pour pallier à ce problème, une architecture encodeur-décodeur a été mise en place, le U-Net [32].

3.2.3 U-Net

L'intérêt de l'architecture U-Net par rapport à un auto-encodeur classique est la présence de sauts de connexion, *Skip connections* en anglais. Ceux-ci permettent une meilleure préservation des détails [33] . Le principe est de contracter l'image pour en extraire le contexte et ensuite de faire une expansion. L'expansion doit se faire de manière symétrique à la contraction. Cette technique permet une plus grande précision qu'un réseau de neurones convolutif classique. Un autre avantage de ce type de modèle est qu'il demande peu d'images pour avoir un entraînement efficace. Un exemple de ce type d'architecture est disponible à la figure 3.7, son nom U-Net venant de la forme en U de l'architecture. Sur cette figure, différents éléments sont visibles : les cases bleues correspondent aux cartes de caractéristiques à plusieurs canaux, les boites blanches correspondent à une copie des cartes de caractéristiques qui permettent de faire le saut de connexion et enfin, les flèches montrent les différentes opérations réalisées [34].

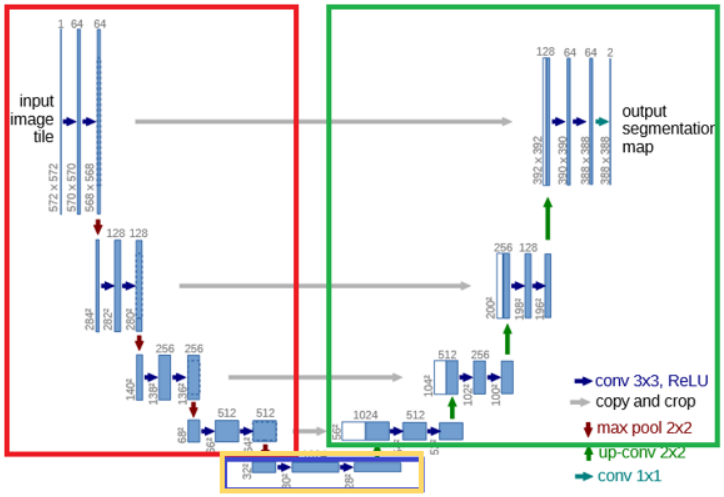


FIGURE 3.7 – Exemple d'architecture U-Net. La partie encodeur est en rouge, la partie décodeur est en vert et le pont entre les deux est en jaune. Les flèches grises entre la partie encodeur et décodeur représentent les saut de connexion. L'image est extraite et modifiée à partir de [34].

Encodeur

La partie encodeur de cette architecture est représentée par le cadre rouge de la figure 3.7. L'encodeur est similaire au réseau de neurones convolutif présenté à la section 3.2.2. En effet, des couches du même type s'y retrouvent, à l'instar des couches de convolution ou des couches de regroupement. La contraction consiste en l'application répétée de deux couches de convolution et d'une couche de regroupement. Chacune des couches convolutives est suivie d'une rectification linéaire unitaire (ReLU), suivant le même principe que dans les réseaux de neurones convolutifs. Dans l'architecture illustrée sur la figure 3.7, ce sont des images en deux dimensions de taille 572 × 572 qui sont données en entrée du réseau. Après avoir subi les deux premières convolutions avec un noyau 3 × 3, représentées par deux flèches bleues à la figure 3.7, on obtient 64 cartes d'activations de tailles 568 × 568. Ensuite, la couche de regroupement basée sur le maximum, représentée par une flèche rouge, est appliquée avec un groupe de taille 2 × 2 pour arriver à une dimension de 284 × 284. Ces deux étapes vont être appliquées de manière alternée, en tout quatre fois, pour obtenir 512 cartes de dimension de 32 × 32.

Pont

Cette partie se trouve dans le cadre jaune sur la figure 3.7. Le pont correspond à la partie qui fait le lien entre l'encodeur et le décodeur. Il s'agit de deux couches convolutives, chacune suivie d'une couche ReLU. Elle permet d'ajuster la taille désirée pour commencer la partie décodeur. Après cette étape, 1024 cartes de taille 28×28 sont obtenues.

Décodeur

Le décodeur permet la reconstruction de l'image, cette partie se trouve dans le cadre vert sur la figure 3.7. A chaque étape, un sur-échantillonnage de chaque carte d'activation est effectué, c'est-à-dire une augmentation de la dimension afin que celle-ci atteigne la dimension initiale à la fin du décodeur. Chacune de ces étapes est suivie de deux couches de convolution 2×2, ce qui permet de diviser par deux le nombre de cartes d'activation. Une couche de rectification linéaire unitaire est également présente après chaque couche de convolution. Par exemple, au début du décodeur, il y a 1024 cartes d'activation de dimensions de 28×28. Après le premier sur-échantillonage, il reste 1024 cartes ayant une résolution de 56×56. Après les deux couches de convolutions, le nombre de cartes est passé à 512 avec une résolution de 52×52.

Saut de connexion

Le saut de connexion est représenté par les flèches grises sur la figure 3.7, il permet de faire une copie de l'image à chaque étape de l'encodeur, d'en garder une partie et de la faire parvenir à l'étape correspondante du décodeur. Cette partie du U-Net est cruciale. En effet, à chaque couche de regroupement, il y a une perte de détails de plus hautes résolutions. La partie intacte de l'image permet au décodeur de recréer les détails perdus lors des convolutions.

U-Net en trois dimensions

Dans ce travail, nous souhaitons segmenter des images en 3 dimensions. Il faut donc adapter l'architecture de la figure 3.7 pour qu'elle puisse prendre en entrée des images dans cette dimension.

L'architecture globale reste identique, en forme de U avec les mêmes types de couches aux mêmes endroits. Cependant, les différentes couches, qui précédemment étaient en deux dimensions sont, dans ce cas, en trois dimensions. Cela veut également dire qu'au lieu de travailler avec des pixels, ce sont des voxels qui sont pris en compte [35]. Cela est illustré à la figure 3.8.

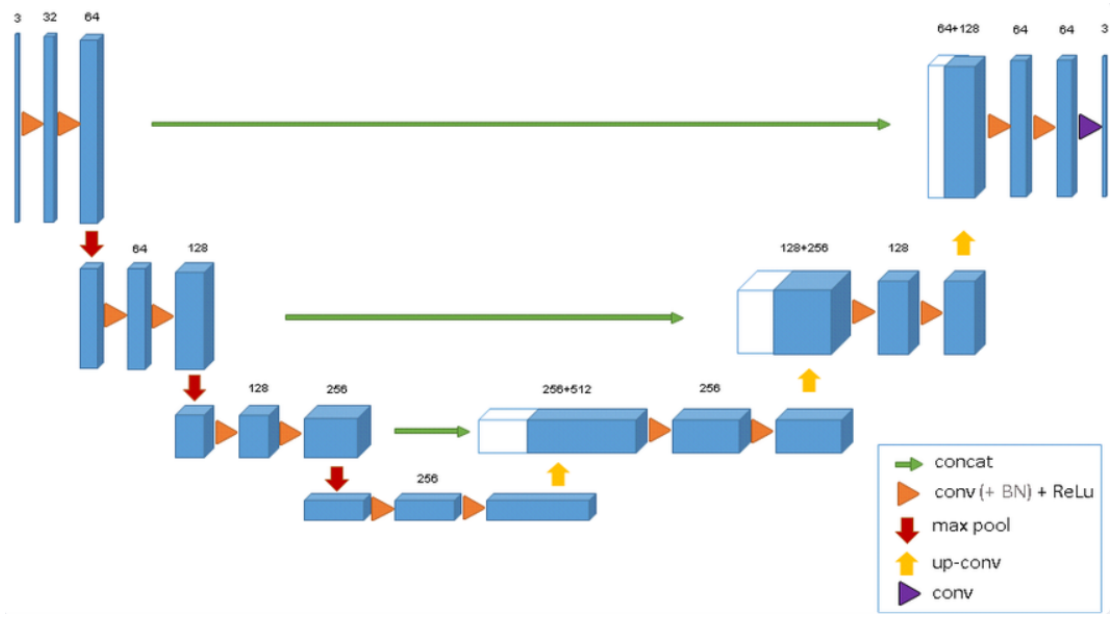


FIGURE 3.8 – Exemple d'architecture U-Net en trois dimensions. Les triangles orange correspondent à des couches de convolutions, les flèches rouges sont les couches de regroupement. Les flèches vertes représentent les sauts de connexion. Image extraite de [35].

3.3 Entraînement d'un réseau de neurones

Afin d'entraîner un réseau de neurones, plusieurs choses sont à prendre en compte, comme la fonction de coût qui permet de quantifier la différence entre la prédiction du modèle et la sortie attendue. Il y a également la vitesse d'apprentissage, qui détermine la vitesse à laquelle la valeur des poids va être ajustée lors de la descente de gradient. Avant de s'intéresser à ces paramètres, il faut comprendre comment fonctionne un neurone.

3.3.1 Fonctionnement d'un neurone

Le fonctionnement d'un neurone se base sur le principe du perceptron. Celui-ci représente le fonctionnement d'un neurone biologique, il peut uniquement effectuer des classifications linéaires. Comme le montre la figure 3.9, le neurone prend différentes entrées, chacune étant prise en compte selon son importance, qui se traduit par les poids w_i. Il faut également mentionner la présence d'un biais caractérisé par l'entrée 1 et le poids w_0.

La première étape de l'apprentissage consiste à effectuer la somme pondérée des entrées, appelée entrée nette de la fonction :

$$Entrée\ nette\ de\ la\ fonction = \sum W_i X_i = w_0{*}1 + w_1{*}X_1 + w_2{*}X_2 + ... + w_m{*}X_m \quad (3.2)$$

Ensuite, une fonction d'activation ψ est appliquée. Celle-ci donne la sortie finale du neurone sous une forme binaire ou continue, selon la fonction utilisée.

$$Sortie = \psi * (\sum W_i X_i) \quad (3.3)$$

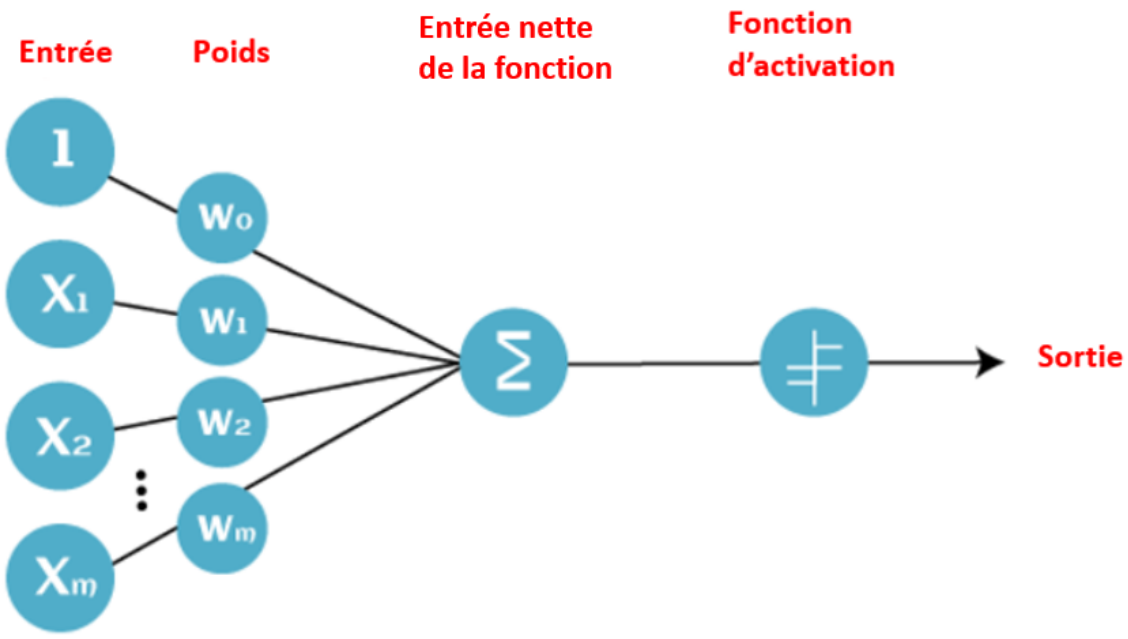


FIGURE 3.9 – Principe de perceptron. Les différentes entrées sont prises en compte selon un certain poids, chaque entrée a son propre poids. L'entrée 1 associé au poids w_0 représente le biais. L'image est issue de [36] et traduite en français.

La fonction d'activation permet de déterminer si le neurone va s'activer ou non. Il en existe plusieurs types, les deux plus connues sont la fonction sigmoïde et la rectification linéaire unitaire, elles sont représentées à la figure 3.10 [37].

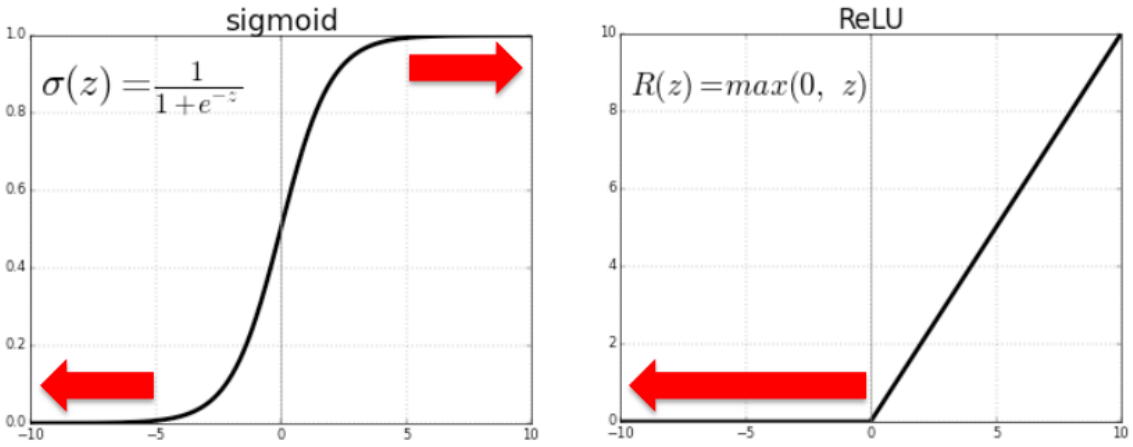


FIGURE 3.10 – Exemple de fonctions d'activation. A gauche, le graphe de la fonction sigmoïde, à droite, celui de la rectification linéaire unitaire. L'image est issue de [37].

3.3.2 Métriques

Pour évaluer les performances d'un modèle, différentes métriques peuvent être utilisées. Cette section en présente trois : le dice, l'erreur quadratique moyenne et la somme de la différence quadratique.

Dice

Le dice permet de quantifier la similarité entre deux ensembles de pixels ou de voxels. Dans ce mémoire, il est utilisé pour comparer la performance du modèle et celle d'un être humain, souvent expert dans le domaine comme des radiologues dans le cas de segmentation d'images médicales. La différence se base sur des calculs d'intersection entre les surfaces ou les volumes [38]. Voici comment il se calcule :

$$Dice = \frac{2|\mathcal{O} \cap \mathcal{O}_m|}{2|\mathcal{O} \cap \mathcal{O}_m| + |\mathcal{B} \cap \mathcal{O}_m| + |\mathcal{O} \cap \mathcal{B}_m|} \tag{3.4}$$

Où $\mathcal{O}$ est la segmentation connue, $\mathcal{B}$ est l'arrière plan, $\mathcal{O}_m$ est la segmentation générée par le modèle m et $\mathcal{B}_m$ est l'arrière plan défini par le modèle m.

Erreur quadratique moyenne

Appelée *Root Mean Square Error (RMSE)* en anglais, l'erreur quadratique moyenne est une métrique régulièrement utilisée pour évaluer la performance d'un modèle. Elle est particulièrement adaptée lorsque la distribution de l'erreur est supposée gaussienne [39]. Cette erreur est calculée sur base de l'équation 3.5.

$$erreur = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{3.5}$$

Où e_i est l'erreur de l'échantillon. Pour les images, e_i est la différence de valeur des pixels correspondants sur les deux images que nous souhaitons comparer. La valeur de n est le nombre d'échantillons disponibles. Pour les images, c'est le nombre de pixels.

Somme de la différence quadratique

Appelée *Sum of the Squared Differences (SSD)* en anglais, la somme de la différence donne une quantification de la distance entre deux données [40]. L'équation de cette métrique est semblable à celle de l'erreur quadratique moyenne mais elle ne présente pas de racine carrée. La somme de la différence quadratique se calcule suivant l'équation 3.6.

$$erreur = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \tag{3.6}$$

Où e_i est l'erreur de l'échantillon. Pour les images, e_i est la différence de valeur des pixels correspondants sur les deux images que nous souhaitons comparer. La valeur de n est le nombre d'échantillons disponibles. Pour les images, c'est le nombre de pixels.

3.3.3 Fonction de coût

Appelée *Loss Function* en anglais, la fonction de coût quantifie la différence entre la prédiction faite par le modèle et le résultat désiré. Au plus sa valeur est grande, au plus le résultat obtenu est éloigné de ce qui est souhaité. Au contraire, au plus le coût est faible, au plus la prédiction est proche du résultat voulu.

Lors de l'entraînement, l'objectif est dès lors de minimiser la valeur du coût. La fonction provient généralement d'une fonction de métrique comme le dice présenté à la section précédente.

Elle peut également être une combinaison de plusieurs métriques différentes. Bien entendu, la fonction de coût va dépendre de ce qui doit être comparé. Pour ce mémoire, plusieurs fonctions sont comparées : la fonction issue de la métrique dice, ainsi que deux améliorations de cette fonction, la fonction de log cosh dice et celle du focal dice [41].

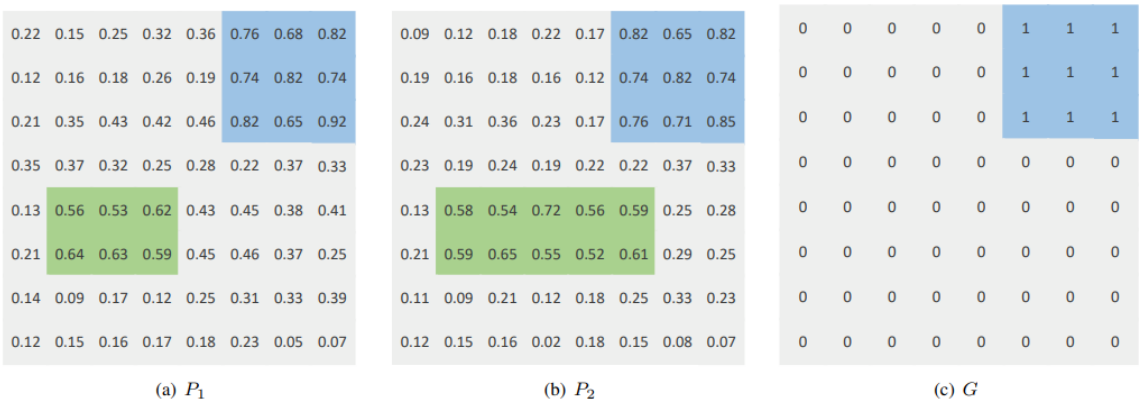
Fonction de coût dice

Cette fonction de coût se base sur la métrique dice. Elle permet d'équilibrer l'importance entre l'avant plan d'une image et son arrière plan. Elle se calcule de la façon suivante :

$$Fonction\ \ de\ \ coût = 1 - dice = 1 - \frac{2|\mathcal{O} \cap \mathcal{O}_m|}{2|\mathcal{O} \cap \mathcal{O}_m| + |\mathcal{B} \cap \mathcal{O}_m| + |\mathcal{O} \cap \mathcal{B}_m|} \tag{3.7}$$

Où $\mathcal{O}$ est la segmentation connue, $\mathcal{B}$ est l'arrière plan, $\mathcal{O}_m$ est la segmentation prédite par le modèle m et $\mathcal{B}_m$ est l'arrière plan défini par le modèle m.

Cependant, il persiste certains déséquilibres. En effet, certains arrières plans sont plus complexes à classer que d'autres et cette fonction de coût n'en tient pas compte. Dès lors, si deux classifications sont comparées comme à la figure 3.11 où P1 et P2 sont deux cartes de probabilités prédites de la même image et G est le masque réel correspondant.



(a) P_1 (b) P_2 (c) G

FIGURE 3.11 – Représentation du déséquilibre avec une fonction de coût issue de la métrique dice. Le coût de la carte P1 est plus élevée que celui de la carte P2 alors qu'elle présente de meilleurs résultats car l'arrière plan n'a pas assez d'importance dans le calcul. L'image est issue de [41].

L'exemple traite de la classification d'une image, si la valeur est de 1, une lésion est présente, si elle vaut 0, il n'y a pas de lésion. On observe que la P1 a un coût plus élevé alors qu'elle montre un meilleur résultat que la P2. L'aire bleue indique la classification correcte de la valeur 1, il y a une lésion. Les zones vertes montrent des classifications en tant que 1 alors que des 0 étaient attendus, une lésion est prédite alors qu'il n'y en a pas, ce sont les faux positifs.

Combinaison de la fonction de coût dice et focal

Cette combinaison introduit une stratégie d'échantillonage équilibré, ce que la fonction de coût précédente ne peut pas faire. La première étape consiste à diviser les cartes de probabilités prédites en trois parties, celles-ci sont décrites dans le tableau 3.1.

Tableau 3.1 – Description de la division des cartes de probabilité pour la fonction de coût focal dice

A	Vrai positifs Faux négatifs	Une lésion est prédite et une lésion existe Un tissu sain est prédit mais une lésion existe
B	Faux positifs Une partie des vrais négatifs	Une lésion est prédite mais le tissu est sain Un tissu sain est prédit et le tissu est sain
C	Le reste des vrais négatifs	Un tissu sain est prédit et le tissu est sain

Le schéma à la figure 3.12 représente cette division des cartes de probabilités prédites.

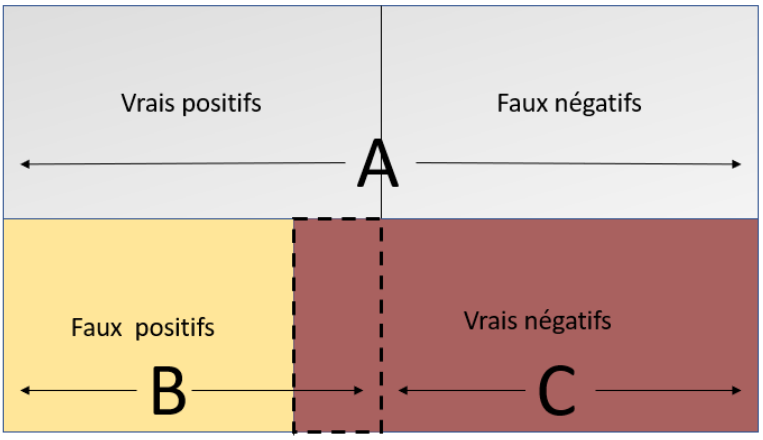


Figure 3.12 – Division des cartes de probabilités prédites en trois parties afin de donner plus d'importance à l'arrière plan, représenté par les vrais négatifs.

Cette technique permet de donner davantage d'importance aux classifications plus difficiles. Cela revient à résoudre le problème de la simple fonction de coût dice qui ne donne pas assez d'importance à l'arrière plan représenté par les vrais négatifs. La formule pour la fonction de coût dice focal est la suivante :

$$Coût \quad dice \quad focal \quad = 1 - \frac{2\sum_{i=1}^{N} p_i g_i}{\sum_{i=1}^{|A|+|B|} p_i^2 + \sum_{i=1}^{N} g_i^2} \tag{3.8}$$

Où p_i appartient à P qui est la probabilité prédite du i-ème voxel et g_i appartient à G qui est le masque à prédire du i-ème voxel.

Fonction de coût basée sur le log cosh dice

Ce type de fonction est également inspiré de la fonction dice. Elle permet de traiter des données asymétriques. C'est-à-dire des données pour lesquelles la taille des objets est très déséquilibrée [42][43]. Elle se calcule de la manière suivante :

$$Fonction\ de\ coût = Log(cosh(1-dice)) = Log(cosh(1-\frac{2|\mathcal{O} \cap \mathcal{O}_m|}{2|\mathcal{O} \cap \mathcal{O}_m| + |\mathcal{B} \cap \mathcal{O}_m| + |\mathcal{O} \cap \mathcal{B}_m|})) \tag{3.9}$$

Où $\mathcal{O}$ est la segmentation connue, $\mathcal{B}$ est l'arrière plan, $\mathcal{O}_m$ est la segmentation générée par le modèle m et $\mathcal{B}_m$ est l'arrière plan défini par le modèle m.

3.3.4 Rétro-propagation de l'erreur

Pour rappel, l'objectif est d'optimiser la valeur des poids des neurones du réseau afin de segmenter correctement les images. Ils vont donc être ajustés à chaque itération en fonction de l'erreur, celle-ci doit être minimisée. Elle est définie à l'équation 3.10, où t_i correspond à la cible et y_i correspond à la prédiction faite par le modèle.

$$E = (t_i - y_i)^2 \tag{3.10}$$

L'optimisation des poids et donc l'entraînement du réseau se fait par rétro-propagation de l'erreur. Tout le raisonnement suivant provient du cours LELEC2870 Machine Learning donné à l'UCLouvain [37].

Si nous reprenons une schématisation de neurones à la figure 3.13. Trois couches sont représentées, les couches l-1, l et l+1. Les indices i, j et k correspondent aux différents neurones. Les poids sont notés W, les entrées de neurones a et leurs sorties Z.

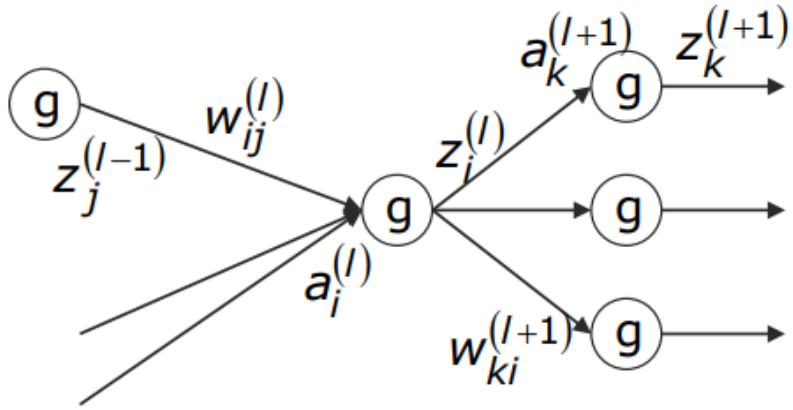


Figure 3.13 – Schéma avec notations de neurones. Trois couches sont représentées, les couches l-1, l et l+1. Les indices i, j et k correspondent aux différents neurones. Les poids sont notés W, les entrées de neurones a et leurs sorties Z. Image issue du cours LELEC2870 Machine Learning donné à l'UCLouvain [37].

Sur base de la figure 3.13, l'entrée $a_i^{(l)}$ peut s'écrire comme la combinaison linéaire des sorties des neurones de la couche précédente, l-1.

$$a_i^{(l)} = \sum_j W_{ij}^{(l)} Z_j^{(l-1)} \tag{3.11}$$

Puisque l'on veut minimiser l'erreur en adaptant les poids, il faut calculer la dérivée de l'erreur par rapport aux poids.

$$\frac{\partial E}{\partial W_{ij}^{(l)}} = \frac{\partial E}{\partial a_i^{(l)}} \frac{\partial a_i^{(l)}}{\partial W_{ij}^{(l)}} = \delta_i^{(l)} Z_j^{(l-1)} \tag{3.12}$$

Où $\delta_i^{(l)} = \frac{\partial E}{\partial a_i^{(l)}}$. Ce terme peut être connu pour la dernière couche.

$$\delta_i^{(l)} = \frac{\partial E}{\partial a_i^{(l)}} = \frac{\partial E}{y_i^{(l)}} \frac{y_i^{(l)}}{\partial a_i^{(l)}} = \frac{\partial E}{y_i^{(l)}} g'(a_i^{(l)}) \tag{3.13}$$

Dans l'équation 3.13, le premier terme peut être calculé car l'équation de l'erreur est connue et donnée à l'équation 3.10.

$$\frac{\partial E}{y_i^{(l)}} = -2(t_i - y_i) \tag{3.14}$$

Le deuxième terme de l'équation 3.13 est déjà connu. Cela veut dire que le gradient à la dernière couche est également connu. Afin de le déterminer pour les autres couches, il faut réaliser une rétro-propagation du gradient. On obtient l'équation 3.15 pour les autres couches.

$$\delta_i^{(l)} = \frac{\partial E}{\partial a_i^{(l)}} = \sum_k \frac{\partial E}{\partial a_k^{(l+1)}} \frac{\partial a_k^{(l+1)}}{\partial a_i^{(l)}} \tag{3.15}$$

Si l'on résume ce raisonnement, les étapes suivantes sont à réaliser pour la rétro-propagation de l'erreur :

- Appliquer une entrée au réseau et la propager dans celui-ci afin d'évaluer toutes les activations $Z_i^{(l)}$ et les sorties des neurones $a_i^{(l)}$.
- Évaluer le terme d'erreur $\delta_i^{(o)}$ de la dernière couche
- Rétro-propager $\delta_i^{(l)}$ afin de trouver le terme $\delta_i^{(l-1)}$.
- Calculer toutes les dérivées $\frac{\partial E}{\partial W_{ij}^{(l)}} = \delta_i^{(l)} Z_j^{(l-1)}$
- Ajuster les poids en fonction des dérivées en faisant une descente de gradient.

3.3.5 Descente de gradient

Cette section explique la dernière partie de la rétro-propagation de l'erreur, la descente de gradient. Cette technique est très intéressante, mais présente certains inconvénients, comme demander beaucoup de temps. Cette méthode présente également le risque de rester bloqué dans un minimum local. La valeur des poids est modifiée à chaque itération selon l'équation 3.16.

$$W(t+1) = W(t) - \alpha \frac{\delta E}{\delta W}|_{W(t)} \tag{3.16}$$

Où α est la vitesse d'apprentissage, elle permet d'adapter la vitesse à laquelle la descente de gradient est effectuée. Sa valeur peut être fixe tout au long de l'apprentissage ou être modifiée au cours du temps. Si elle est mal choisie, l'erreur ne convergera pas vers le minimum global, mais au contraire, si elle est bien choisie, le modèle convergera. Lors d'une descente de gradient classique, toutes les données sont utilisées à chaque étape. Cela veut dire qu'au plus la quantité de données est grande, au plus cette technique prendra du temps.

Descente de gradient stochastique

Afin de pouvoir réaliser une descente plus rapidement, même lorsqu'une grande quantité de données est traitée, la méthode de la descente de gradient stochastique a été mise en place. C'est toujours l'équation 3.16 qui guide la modification des poids. Cependant, dans ce cas, ce ne sont pas toutes les données qui sont prises en compte en même temps mais une seule donnée à la fois et plusieurs itérations sont réalisées. Cela veut dire qu'à chaque itération, la valeur des poids est modifiée mais uniquement sur base d'une seule donnée. Ainsi, la convergence se fera plus rapidement même avec beaucoup de données. Le choix de la donnée se fait de manière aléatoire, c'est pourquoi il est important de mélanger les données.

Une variante de cette méthode, avec des mini-groupes, permet également de ne pas utiliser toutes les données en même temps mais uniquement un petit groupe de données.

Méthode d'Adam

La méthode d'Adam reprend le même principe que la descente de gradient stochastique. Dans ce cas, la différence réside dans le taux d'apprentissage α. Auparavant, il était le même pour tous les poids, maintenant un taux d'apprentissage différent est appliqué pour chaque poids [44].

3.3.6 Sur-apprentissage

En anglais *Overfitting*, le sur-apprentissage est une conséquence à laquelle il faut prêter attention lorsque l'on entraîne un réseau de neurones. Une manière d'identifier le sur-apprentissage est d'observer une dégradation des performances sur les données de validation alors que l'apprentissage continue sur les données d'entraînement. La dégradation des performances sur les données de validation peut s'observer sur le graphe du dice en fonction du nombre d'epochs effectuées. Le nombre d'epochs correspond au nombre d'itérations des poids par donnée. Par exemple, la figure 3.14 montre un début de dégradation des performances lorsque la courbe fait un plateau.

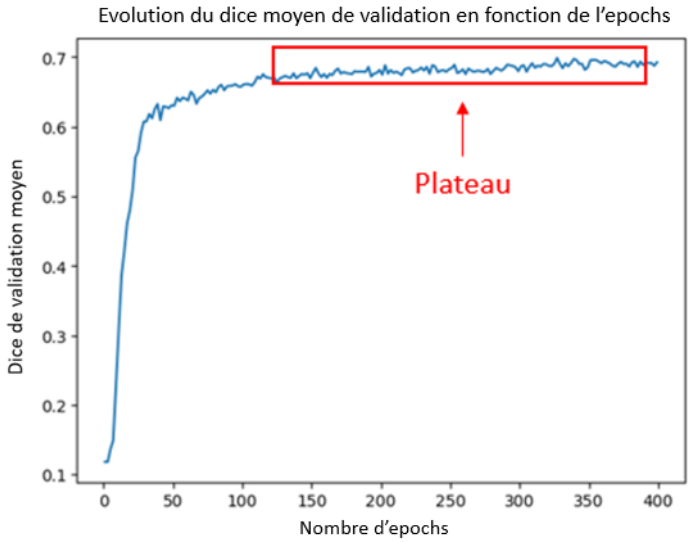


FIGURE 3.14 – Exemple de sur-apprentissage. La dégradation des performances sur les données de validation commence lorsque le plateau apparaît.

C'est un phénomène qui survient lorsque le modèle fait de très bonnes prédictions sur les données d'entraînement mais de mauvaises sur les données de test. Cela peut avoir pour conséquence une mauvaise généralisation sur des données inconnues au modèle. Pour résumer, le modèle a trop appris sur les mêmes données.

3.3.7 Paramètres modifiables

Afin d'obtenir les meilleurs résultats possibles, plusieurs paramètres appelés hyper-paramètres, peuvent être modifiés. On y retrouve la taille des groupes, le nombre de couches, la vitesse d'apprentissage, le dropout, l'échauffement, l'augmentation de la base de données.

La **taille des groupes**, *Batchsize* en anglais, correspond au nombre de données qui sont traitées en même temps lors de la descente de gradient.

Le **dropout** permet d'améliorer la convergence du modèle et d'essayer d'empêcher le sur-apprentissage. Il consiste à retirer certains neurones pendant une partie de l'apprentissage et de les remettre par la suite. Lorsqu'ils ne sont pas pris en compte, les autres neurones apprennent davantage. Une fois que tous les neurones sont à nouveau actifs, le résultat est supposé être meilleur.

L'**échauffement** consiste à atteindre progressivement la vitesse d'apprentissage. Une fois la vitesse désirée atteinte, celle-ci va diminuer. Cette diminution peut se faire de différentes manière, par exemple linéairement ou avec un cosinus.

Lorsque l'on dispose de peu de données, il faut **augmenter la quantité de données**. L'opération consiste à amplifier la base de données en effectuant des rotations, des translations, des changements d'intensité ou encore du bruitage sur les données de départ. Grâce à cette technique, le modèle devient normalement plus robuste car toutes les images ne sont pas toujours présentées de la même manière.

3.4 Résumé

Ce chapitre rassemble les différents concepts théoriques en ce qui concerne les modèles utilisés pour segmenter des images médicales mais également les techniques pour entrainer ces modèles.

Nous avons vu les méthodes d'apprentissage qui existent en fonction du type de données disponibles. Nous avons également explicité les structures du réseau de neurones artificiels, du réseau de neurones convolutif et enfin du U-Net.

Ensuite, nous nous sommes concentrés sur l'entraînement d'un réseau de neurones en commençant par la compréhension du fonctionnement d'un neurone. Nous avons vu différentes métriques possibles pour évaluer les résultats. Différentes fonctions de coût ont été présentées. Nous avons montré comment la valeur des poids entre chaque neurone est calculée et modifiée durant l'entraînement. Pour finir, nous avons identifié les différents paramètres que nous pouvons modifier pour optimiser les modèles.

Chapitre 4

Exemples de segmentation des cavités cardiaques

Un des objectifs de ce mémoire est de réaliser la segmentation des différentes cavités cardiaques en se basant sur des images en trois dimensions prises par tomodensitométrie. Le chapitre 3 nous a permis de découvrir différents modèles d'apprentissage profond : le réseau de neurones artificiels, le réseau de neurones convolutif et le U-Net. Des recherches ont été effectuées afin de vérifier qu'il est envisageable de segmenter le coeur grâce à l'apprentissage profond. La littérature nous montre que le monde médical s'intéresse déjà depuis longtemps à la segmentation des différentes cavités du coeur. En effet, cela peut aider à la compréhension de son fonctionnement. C'est pourquoi, certains challenges, comme le challenge de la segmentation multimodale du cœur entier, *Multi-Modality Whole Heart Segmentation challenge (MM-WHS)* en anglais, ont été mis en place afin de pousser les chercheurs à améliorer encore et encore les techniques utilisées [35][45][46]. Celles-ci seront présentées dans le chapitre 5. Les articles relatant ce genre de travaux sont nombreux et présentent diverses méthodes similaires.

Dans la suite de ce chapitre, les prochaines sections exposent six papiers jugés proches de ce mémoire. Ces articles traitent différentes méthodes. Certaines utilisent les coupes des données en trois dimensions pour donner des images en deux dimensions au modèle. La segmentation est effectuée en deux dimensions et les images en trois dimensions sont reconstruites par la suite. Tandis que d'autres articles utilisent directement les données en trois dimensions. Les articles sont comparés grâce à la métrique dice.

4.1 Segmentation automatique du ventricule gauche grâce à des réseaux de neurones convolutifs

Zreik et al. en 2016 se sont concentrés sur la délimitation du ventricule gauche [47]. Ce papier présente une méthode en deux étapes. La première consiste à créer une boite en trois dimensions autour du ventricule. Pour y parvenir, une combinaison de trois réseaux de neurones convolutifs est mise en place, comme sur la figure 4.1. Chacun de ces réseaux traite une coupe possible de l'image, c'est-à-dire les coupes sagittale, frontale et transversale. Pour chaque vue, c'est un groupe de taille 48×48 qui est donné en entrée. Afin d'entraîner ces réseaux, 50 images en trois dimensions prises par tomodensitométrie ont été utilisées et 30 epochs ont été réalisées. A l'issue de cette première étape, un autre réseau du même type combine les sorties des trois réseaux pour déterminer la probabilité de chaque voxel comme faisant partie du ventricule gauche.

Ce réseau est entraîné avec cinq données sur 70 epochs.

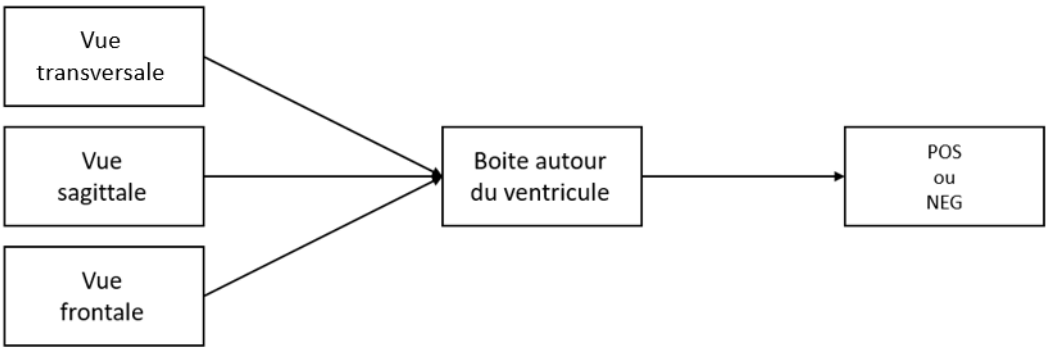


FIGURE 4.1 – Schéma de la méthode en deux étapes de la segmentation du ventricule gauche. Une première étape combine 3 réseaux de neurones pour former une boite autour du ventricule. Un dernier réseau détermine la probabilité de chaque voxel comme faisant partie ou non du ventricule.

La figure 4.2 montre le résultat en image. Sur la gauche, il y a l'image de base que l'on veut délimiter. Celle du milieu représente la délimitation réalisée par un radiologue, c'est le résultat attendu. Enfin, l'image de droite représente le résultat obtenu.

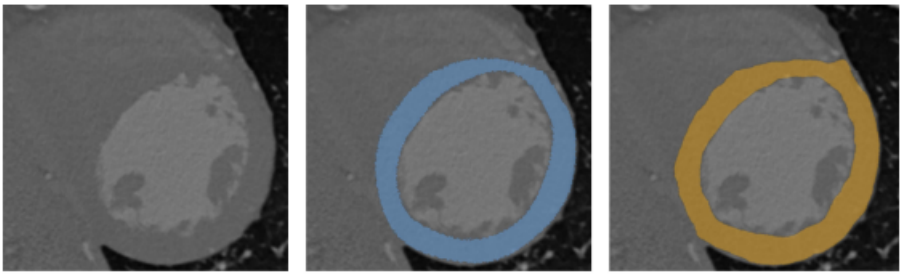


FIGURE 4.2 – Résultat en image de la méthode en deux étapes de la segmentation du ventricule gauche. L'image de gauche correspond à l'image sur laquelle la délimitation doit être effectuée. L'image du milieu représente la délimitation attendue. L'image de droite est le résultat obtenu grâce au modèle. Image issue de l'article [47].

Finalement, le modèle est testé sur cinq données. Un dice moyen de 0.85 est atteint. Le résultat final est également caractérisé par une distance moyenne absolue de 1,1 mm. Ces résultats ont été obtenus sans augmentation de la base de données [47].

4.2 Segmentation automatique du cœur entier à l'aide de l'apprentissage profond et du contexte de forme

Chunliang Wang et Örjan Smedby en 2018 ont écrit ce papier à la suite du challenge MM-WHS, cité précédemment [48]. Pour celui-ci, l'objectif est de segmenter toutes les cavités du coeur. La base de données est constituée de 20 données récoltées par tomodensitométrie, aucune augmentation n'a été appliquée sur ces données. Dans ce cas, la méthode suit trois étapes.

Premièrement, une pré-segmentation est réalisée à partir de trois U-Net en deux dimensions. Ensuite, une estimation de la forme du coeur est faite sur base de modèle statistique volumétrique. Finalement, trois U-Net sont encore utilisés pour finaliser la segmentation. Le schéma de cette méthode est illustré à la figure 4.3.

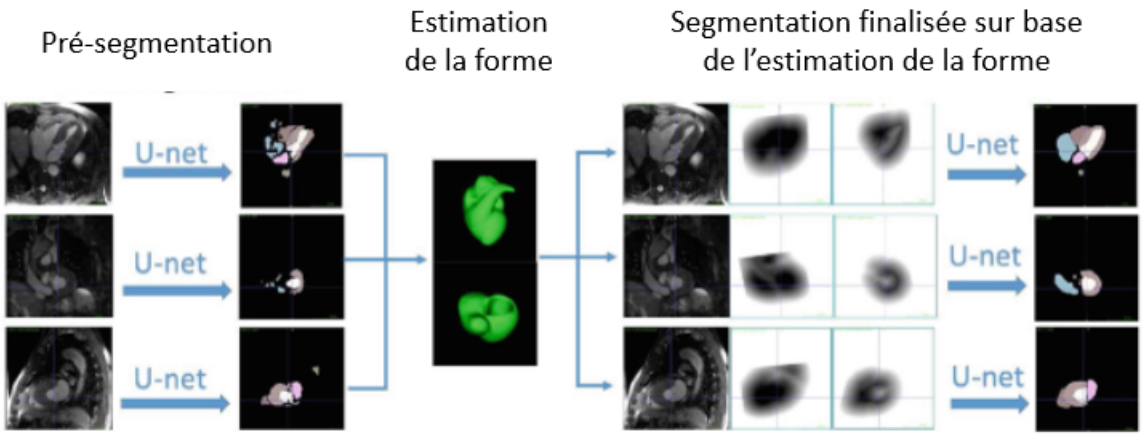


FIGURE 4.3 – Schéma de la méthode en trois étapes de la segmentation des cavités du coeur. Il y a d'abord une pré-segmentation, ensuite une estimation du volume du coeur. Pour finir, une finalisation de la segmentation est réalisée. Schéma issu de l'article [48] et traduite en français.

Le résultat de la segmentation est montré à la figure 4.4.

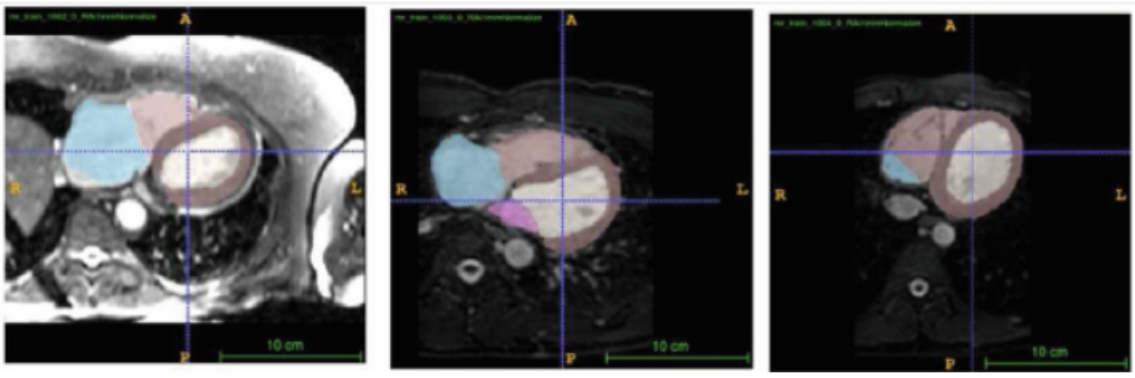


FIGURE 4.4 – Résultat en image de la méthode en trois étapes de la segmentation des cavités du coeur. Figure issue de [48].

Pour cette technique, des valeurs de dice sont disponibles pour les différentes parties du coeur. Ils sont référencés dans le tableau 4.1. Il s'agit de dices moyens obtenus sur 5 validations croisées, chacune en utilisant 16 données pour l'entraînement et 4 pour calculer le dice. Le dice global pour tout le coeur est de 0,79 [48].

Tableau 4.1 – Dices obtenus avec la méthode en trois étapes de la segmentation des cavités du coeur

Parties du coeur	Dices
Ventricule gauche	0,90
Ventricule droit	0,80
Oreillette gauche	0,9
Oreillette droite	0,85
Myocarde du ventricule gauche	0,87
Aorte	0.83
Artère pulmonaire	0.67

4.3 Réseaux convolutifs hybrides

Xin Yang et al. en 2018 propose une technique dite hybride car elle utilise une combinaison de différentes fonctions de coût. Ces fonctions sont utiles lors de l'entraînement pour quantifier la différence entre la sortie du modèle et la segmentation attendue. Dans cet article, ce n'est plus une combinaison de plusieurs modèles en deux dimensions qui est utilisée mais directement une architecture traitant des données en trois dimensions et plus particulièrement un réseau entièrement convolutif en trois dimensions [49]. Les données utilisées sont les mêmes que pour l'article précédent, c'est-à-dire 20 images récoltées par tomodensitométrie. Cette architecture est de type U-Net, elle comporte une partie encodeur et une partie décodeur. Cette dernière utilise deux fonctions de coût différentes. La première fonction permet de résoudre les problèmes de déséquilibre de volume entre les cavités, la deuxième équilibre l'entraînement pour plusieurs classes où chacune correspond à une partie du coeur. Le schéma de cette méthode se trouve à la figure 4.5, tandis que la figure 4.6 montre les résultats obtenus en trois dimensions.

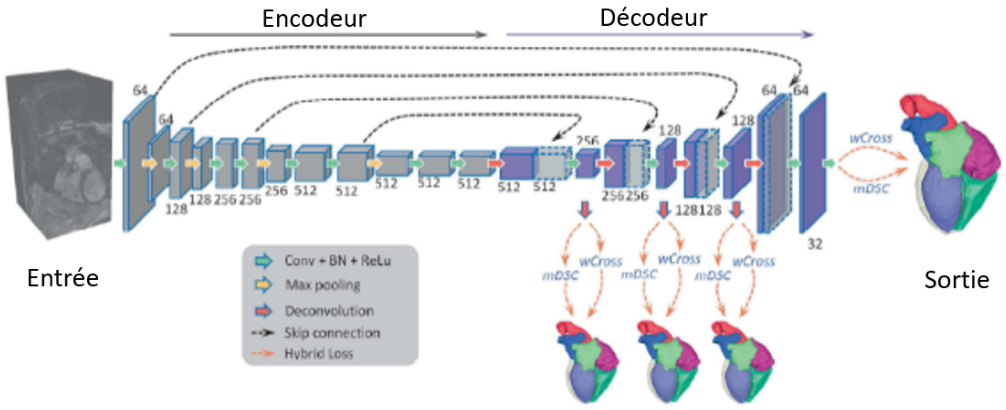


Figure 4.5 – Schéma de la méthode hybride pour la segmentation des cavités du coeur. La partie encodeur se trouve à gauche, tandis que la partie décodeur se trouve à droite. C'est lors de celle-ci que les différentes fonctions de coût sont utilisées. Schéma issu de [49] et traduit en français.

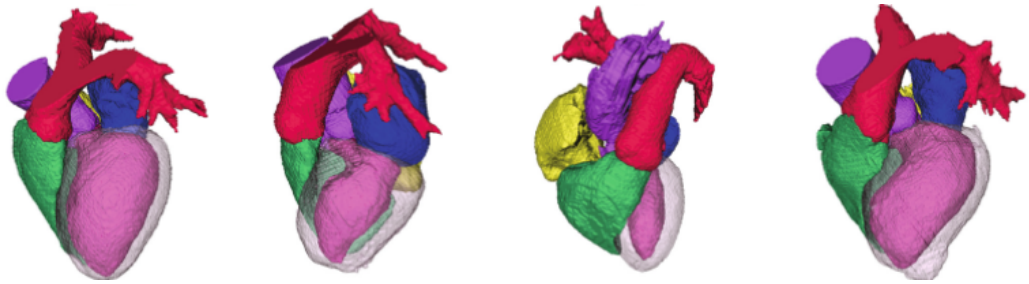


FIGURE 4.6 – Résultat visuel grâce à une reconstruction en trois dimensions de l image. Les résultats sont obtenus par la méthode hybride. Image issue de l'article [49].

Des mesures de dice sont également disponibles pour les sept différentes parties segmentées dans le tableau 4.2. Ils correspondent à la moyenne des dice obtenus sur 10 données. Le dice moyen pour tout le coeur est de 0,80.[49].

TABLEAU 4.2 – Dices obtenus avec la méthode hybride pour la segmentation des cavités du coeur

Parties du coeur	Dices
Ventricule gauche	0,83
Ventricule droit	0,62
Oreillette gauche	0,9
Oreillette droite	0,84
Myocarde du ventricule gauche	0,68
Aorte	0.91
Artère pulmonaire	0.80

4.4 Segmentation automatique du coeur grâce à un U-Net

Akifumi Yoshida et al. en 2021 propose deux manières de segmenter le coeur avec un modèle U-Net en deux dimensions [50]. Le nombre de patients étudiés s'élève à 20. Pour dix de ces patients, deux jeux de données sont disponibles. Dans chacun des jeux, le coeur est à une phase différente de battement. Les données d'entraînement sont composées d'une série d'images en deux dimensions de taille 512 × 512. Chaque donnée est accompagnée de l'annotation correspondante. Elles ont été acquises par tomodensitométrie. Les images ont été cadrées à une taille de 256 × 256 pour être centrées sur le coeur.

La première méthode développée dans cet article consiste à segmenter les quatre cavités principales du coeur en même temps, avec le même modèle. Dans ce cas, 30 epochs sont réalisées, les données sont utilisées par groupe de 8, la fonction d'optimisation est celle d'Adam et la vitesse d'apprentissage initiale est de 0,001. La métrique de dice a été utilisée pour évaluer le résultat final. Le dice final est en réalité une moyenne calculée en réalisant 20 entraînements, à chaque fois 19 données servent d'ensemble d'entraînement et une donnée est utilisée pour calculer le dice.

La seconde méthode développée dans cet article consiste à segmenter les quatre cavités indépendemment les unes des autres et ensuite reconstituer une image avec toutes les cavités. Pour cet entraînement, le nombre d'epochs s'élève à 5. La même méthode d'évaluation est utilisée.

Les différents résultats obtenus avec les deux méthodes se trouvent dans le tableau 4.3. Nous observons que les résultats sont meilleurs lorsque les cavités sont segmentées séparément. Le dice moyen pour tout le coeur est de 0,79 pour la première méthode et de 0,82 pour la seconde.

Tableau 4.3 – Résultats obtenus avec les deux techniques

Parties du coeur	Dices première méthode	Dices deuxième méthode
Ventricule gauche	0,87	0,85
Ventricule droit	0,75	0,84
Oreillette gauche	0,78	0,80
Oreillette droite	0,77	0,80

La figure 4.7 montre les résultats visuels obtenus avec les deux méthodes. Les images du haut représentent les annotations faites par un expert. L'image en bas à gauche est la segmentation des cavités avec la première méthode. L'image en bas à droite est la segmentation avec la deuxième méthode.

(a) Utilisation d'un modèle pour toutes les parties à segmenter

(b) Utilisation d'un modèle pour chaque partie à segmenter

Figure 4.7 – Comparaison des résultats pour les deux méthodes de segmentation. Les images du haut représentent les annotations faites par un expert. L'image en bas à gauche est la segmentation des cavités avec la première méthode. L'image en bas à droite est la segmentation avec la deuxième méthode. Images issues de l'article [50].

4.5 Segmentation du coeur avec un U-Net en trois dimensions avec deux types de données

Qianqian Tong et al. en 2018 présentent un U-Net en trois dimensions alors que les articles présentés précédemment utilisaient des U-net en deux dimensions [51]. Une autre particularité de ce travail est qu'il combine l'utilisation de données prises par tomodensitométrie mais également des données acquises par imagerie par résonance magnétique (IRM). Les deux types de données proviennent également du challenge MM-WHS et sont chacun composés de 20 données. Les données ne correspondent pas d'un type d'imagerie à l'autre, il s'agit de patients différents.

La première étape réalisée dans cet article est de supprimer l'impact des tissus environnants du coeur en faisant une délimitation grossière du coeur grâce à un U-Net en trois dimensions. Le résultat de cette étape est visible à la figure 4.8 où nous observons différentes couches d'un même patient, toutes les images présentent une vue transversale. Les données ont été cadrées sur base de cette délimitation. Les données cadrées ont été utilisées pour réaliser une augmentation de la quantité de données en appliquant des rotations, la quantité finale n'est pas donnée dans l'article. Finalement, les données ont été sous-échantillonnées à une dimension de $128 \times 128 \times 64$.

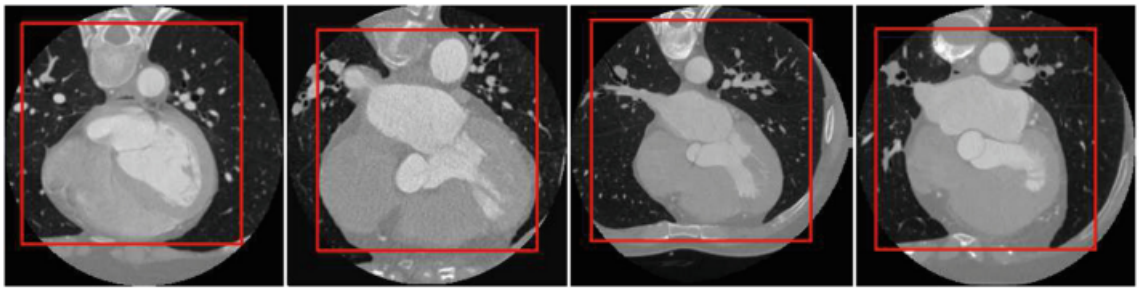


Figure 4.8 – Résultats de la délimitation grossière du coeur avec un U-Net en trois dimensions. Différentes couches du même patient sont présentées, toutes selon une vue transversale. Images issues de l'article [51].

Pour ce qui est de la deuxième étape, des caractéristiques multimodales sont extraites grâce à la fusion des deux types de données. Sur base de celles-ci, un U-Net en trois dimensions est entraîné. Le modèle résultant permet de réaliser une segmentation de chaque partie du coeur. Le résultat visuel pour différentes couches du même patient en vue transversale se trouve à la figure 4.9. Une reconstruction en trois dimensions y est également visible. Les résultats sont montrés sur les données obtenues par tomodensitométrie.

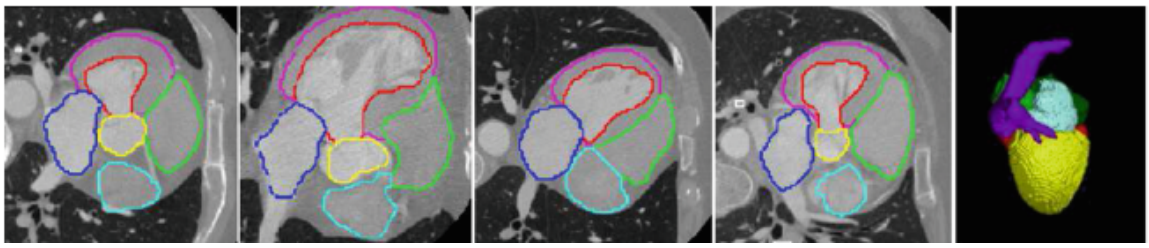


Figure 4.9 – Résultat de la segmentation finale sur différentes couches d'un même patient en vue transversale. La dernière image est une reconstruction en trois dimensions du résultat. Images issues de l'article [51].

Ces résultats sont quantifiés par un dice pour chaque partie segmentée. Aucune précision n'est donnée quant à la manière de calculer ce dice, si cela a été fait sur plusieurs données ou une seule. Les résultats montrent de meilleures performances du modèle pour segmenter les données acquises par tomodensitométrie que par résonance magnétique, ils obtiennent respectivement un dice moyen de 0,84 et 0,67.

Le tableau 4.4 reprend les différentes valeurs de dice atteintes pour toutes les parties du coeur et pour les deux types de données.

Tableau 4.4 – Dices obtenus avec les deux types de données

Parties du coeur	Tomodensitométrie	IRM
Ventricule gauche	0,89	0,70
Ventricule droit	0,81	0,68
Oreillette gauche	0,88	0,67
Oreillette droite	0,81	0,65
Myocarde du ventricule gauche	0,83	0,62
Aorte	0,86	0,59
Artère pulmonaire	0,69	0,47

4.6 Segmentation du coeur avec un U-Net en trois dimensions avec une augmentation artificielle de la quantité de données

Marija Habijan et al. en 2019 présentent également un U-Net en trois dimensions utilisant les données du challenge MM-WHS [52]. La base de données est divisée en 15 données d'entraînement et 5 données de validation.

Une augmentation de la quantité de données est effectuée en se basant sur une analyse en composantes principales, *Principal Components Analysis* en anglais. Les transformations peuvent être représentées par une suite d'opérations appliquées aux données, ces opérations peuvent être les suivantes : rotation, mise à l'échelle ou déformation légère. Dans cet article, deux architectures sont utilisées, une pour localiser le coeur, l'autre pour le segmenter. Le schéma de la méthode utilisée se trouve à la figure 4.10.

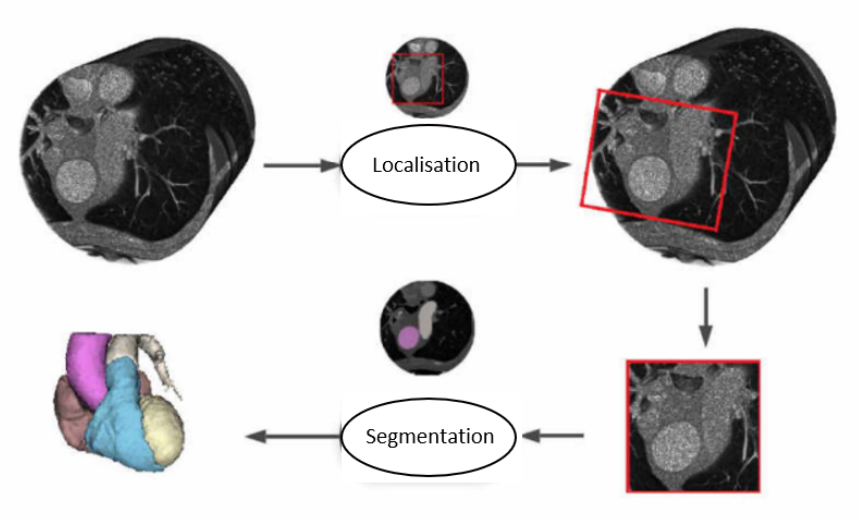


FIGURE 4.10 – Méthode pour la segmentation du coeur grâce à un U-Net en trois dimensions. La partie supérieure de l'image montre la localisation du coeur. Tandis que la partie basse montre la segmentation des différentes parties du coeur. Image issue et traduite depuis l'article [52].

Des tests ont été réalisés avec différentes vitesses d'apprentissage, celle donnant les meilleurs résultats est de 0,005. Un exemple de segmentation obtenue se trouve à la figure 4.11. Sur celle-ci, on observe en haut de gauche à droite, une image en vue transversale, la segmentation attendue et la segmentation obtenue. En bas, une reconstruction en trois dimensions de la segmentation attendue à gauche et une reconstruction de la segmentation prédite à droite.

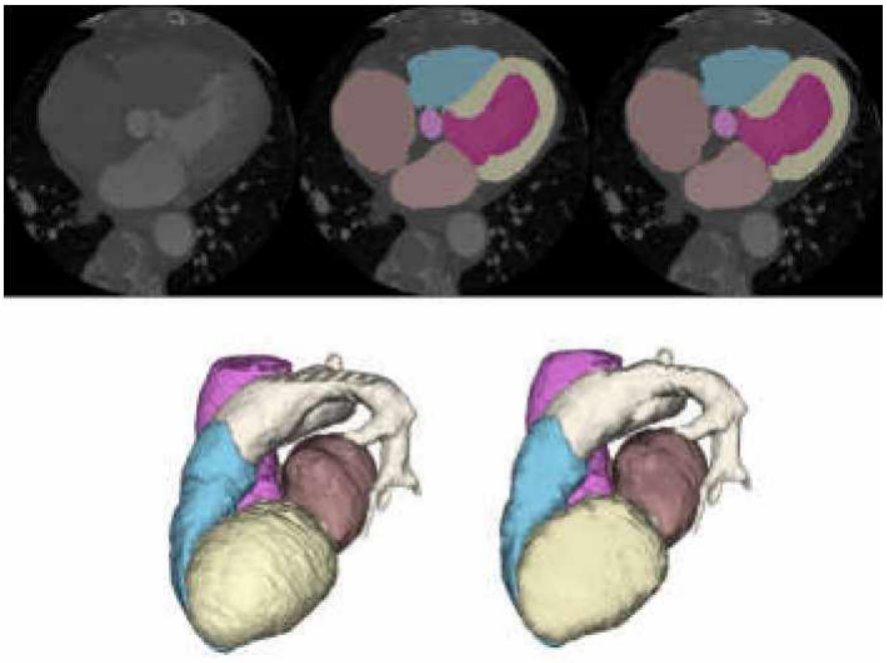


FIGURE 4.11 – Résultat de la segmentation avec un U-Net en trois dimensions et une augmentation de la quantité de données. En haut de gauche à droite, une image en vue transversale, la segmentation attendue et la segmentation obtenue. En bas, une reconstruction en trois dimensions de la segmentation attendue à gauche et une reconstruction de la segmentation prédite à droite. Images issues de l'article [52].

Comme nous l'avons dit, le dice a été calculé sur 5 données de validation. Le tableau 4.5 reprend la valeur des dices moyens avec et sans l'augmentation de la quantité de données. Le dice moyen pour le coeur est de 0,87 sans augmentation et de 0,89 avec augmentation. Ces résultats montrent une amélioration avec une augmentation artificielle de la quantité de données pour toutes les cavités du coeur exceptée l'oreillette droite, l'aorte et l'artère pulmonaire.

Tableau 4.5 – Dices obtenus avec et sans augmentation de la quantité de données

Parties du coeur	Avec augmentation	Sans augmentation
Ventricule gauche	0,92	0,91
Ventricule droit	0,88	0,87
Oreillette gauche	0,90	0,89
Oreillette droite	0,85	0,86
Myocarde du ventricule gauche	0,85	0,84
Aorte	0,92	0,91
Artère pulmonaire	0,85	0,84

4.7 Résumé

Ces articles définissent différentes techniques utilisant des U-Net ou non et des modèles en deux ou trois dimensions. Leurs résultats montrent qu'il est envisageable d'entraîner une architecture U-Net en trois dimensions avec des images prises par tomodensitométrie afin de segmenter les différentes cavités du coeur. En effet, le dice dépasse presque toujours 0,7. Selon [53], le dice moyen entre les segmentations faites par différents radiologues est d'environ 0,7. C'est pourquoi l'objectif est d'obtenir au minimum un dice de 0,7.

Nous pouvons facilement comparer les résultats obtenus pour les articles 2, 3, 5 et 6 car l'entraînement est réalisé sur la même base de données. Pour les articles 2 et 3, le dice moyen obtenu sur tout le coeur est de 0,80 et 0,79, ces résultats sont semblables. Ceux-ci ont utilisé soit des U-Net en deux dimensions, soit un autre type d'architecture en trois dimensions. Les articles 5 et 6 quant à eux obtiennent des dices moyens sur tout le coeur qui s'élèvent à 0,84 et 0,89. Ces résultats sont obtenus en utilisant des U-Net en trois dimensions mais toujours combinés à d'autres techniques comme une première localisation du coeur ou une augmentation de la quantité de données.

Ces articles obtiennent déjà de bons résultats. Dans ce travail, nous combinerons plusieurs de ces idées afin de vérifier si l'utilisation seule d'un U-Net en trois dimensions permet d'obtenir au minimum un dice de 0,7. Si ce n'est pas le cas, nous tenterons progressivement d'ajouter des techniques présentes dans cet état de l'art.

Chapitre 5

Segmentation des différentes cavités du coeur

Ce chapitre relate le travail effectué afin de créer le modèle permettant de segmenter les différentes parties du coeur. La section *Matériels* correspond aux données nécessaires à la réalisation du modèle. Ensuite, la section *Méthodes* présente les différentes expériences réalisées en détaillant les paramètres modifiés. Finalement, la dernière section permet d'exposer et de discuter les résultats obtenus.

5.1 Matériels

Les données exploitées pour cette partie proviennent d'une base de données publiques annotées. Elles viennent plus particulièrement du challenge *Multi Modality Whole Heart Segmentation* réalisé en 2017 [35][45][46]. Dans cette base de données, il y a sept classes correspondant aux sept régions différentes du coeur. Ces classes correspondent aux parties du coeur suivantes : le ventricule gauche (VG), le ventricule droit (VD), l'oreillette gauche (OG), l'oreillette droite (OD), le myocarde du ventricule gauche (MVG), l'aorte (A) et l'artère pulmonaire (AP). En plus de cela, une huitième classe correspond à l'arrière-plan, c'est-à-dire à l'ensemble des pixels ne faisant pas partie du coeur. Grâce à ces annotations, nous pouvons réaliser un apprentissage supervisé. La figure 5.1 est une visualisation de ces données. Différentes couleurs sont attribuées afin de pouvoir différencier chaque partie. Une échelle de couleur sur la droite de chaque image permet de lier chaque couleur au numéro attribué à la classe

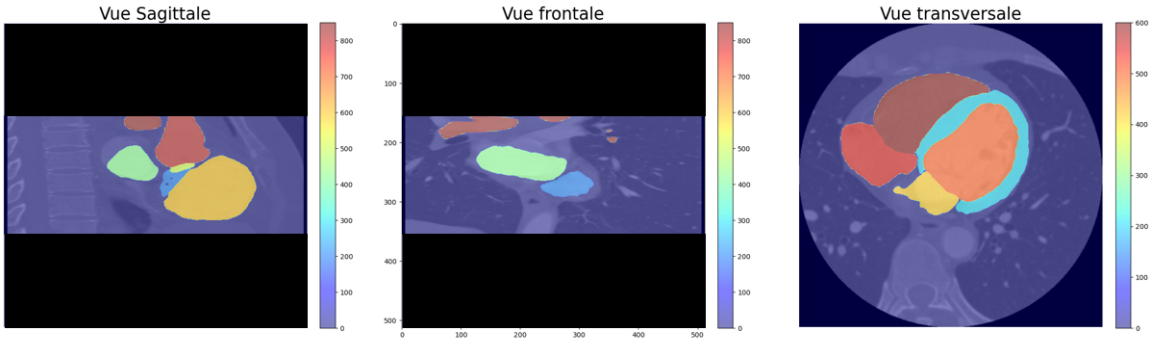


Figure 5.1 – Exemple de données annotées. Les trois vues sont représentées, pour chacune, la segmentation associée à l'image est visible. Images issues de la base de données du Multi-Modality whole heart segmentation challenge [35].

Ensuite, chaque numéro est associé à une partie du coeur, la liste est disponible dans le tableau 5.1. Pour la suite du travail, les différents numéros seront compris entre 0 et 7.

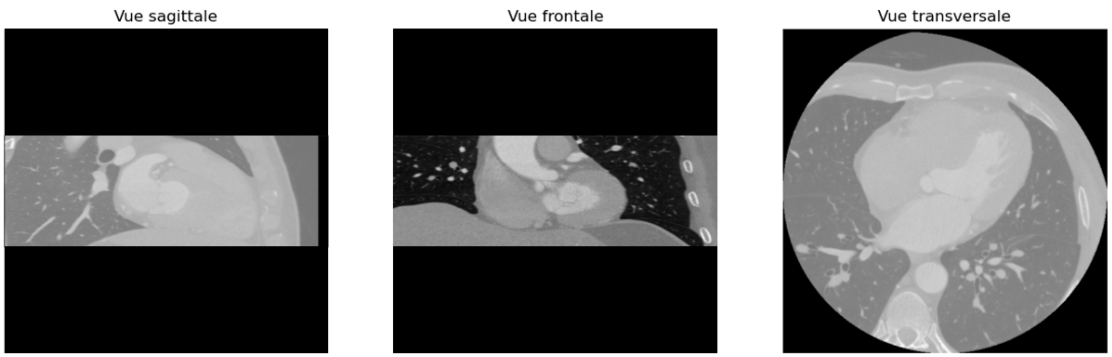
Tableau 5.1 – Lien entre le numéros des étiquettes et les parties du coeur

Numéros initiaux	Numéros attribués	Parties du coeur
0	0	Arrière plan
500	1	Ventricule gauche
600	2	Ventricule droit
420	3	Oreillette gauche
550	4	Oreillette droite
205	5	Myocarde du ventricule gauche
820	6	Aorte
850	7	Artère pulmonaire

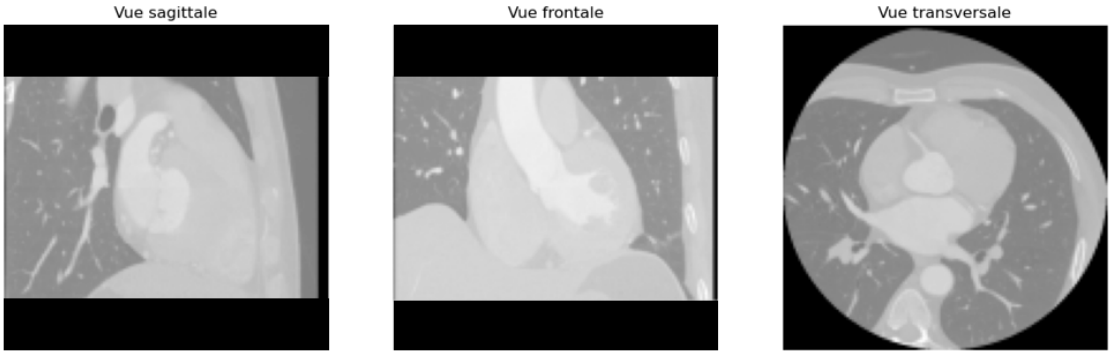
Les données ont une résolution de 512 × 512 × C, où la valeur de C varie entre 177 et 363 en fonction des patients. Les différentes coupes permettant de reconstruire une image en trois dimensions, ont été prises en vue transversale où la résolution dans le plan est de 0,78 mm× 0,78 mm et les coupes sont acquises tous les 1,60 mm.

La quantité de données s'élève à 20, elles sont acquises par tomodensitométrie, centrées sur le coeur. Elles ont été annotées manuellement par des radiologues et collectées au sein d'un environnement clinique. Les données présentent une qualité d'image variable, certaines images présentent plus de bruit ce qui peut rendre la segmentation plus compliquée.

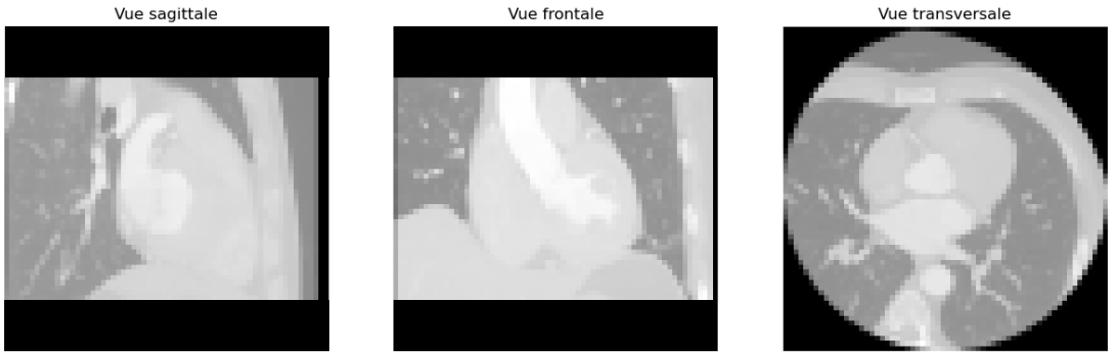
Pour entraîner le modèle, nous avons testé différentes résolutions des données puisque nos ressources ne sont pas suffisantes pour traiter les images en 512 × 512 × C. Nous avons testé des résolutions de 64 × 64 × 64 et de 128 × 128 × 128. La figure 5.2 montre les images dans les trois résolutions différentes. On observe une dégradation de la qualité des données, les détails sont moins visibles et les images deviennent de plus en plus floues lorsque la résolution diminue.



(a) Résolution 512 × 512 × 256



(b) Résolution 128 × 128 × 128



(c) Résolution 64 × 64 × 64

FIGURE 5.2 – Visualisation des données sous différentes résolutions. On observe une dégradation de la qualité des données, les détails sont moins visibles et les images deviennent de plus en plus floues lorsque la résolution diminue.

5.2 Méthodes

Pour rappel, le choix d'utiliser un U-Net en trois dimensions a été fait. Celui-ci a été implémenté grâce à la librairie *MONAI (Medical Open Network for Artificial Intelligence)* [54]. *MONAI* a été créé dans le but d'accélérer la recherche dans le domaine de l'imagerie médicale. Il met à disposition des architectures neuronales déjà "pré-construites", c'est-à-dire qu'un grand nombre de paramètres restent modifiables afin d'adapter au mieux le modèle à la tâche désirée.

Pour entrainer notre modèle, la méthode d'Adam a été utilisée. En effet, celle-ci est généralement utilisée dans l'entraînement d'architecture U-Net. Son utilisation est combinée à l'utilisation de mini-groupes. Pour la fonction d'activation, elle dépend directement de la tâche à accomplir. Nous avons vu au chapitre 3 que pour un modèle U-Net c'est la fonction ReLU qui est utilisée. Finalement, les résultats seront évalués selon la métrique dice.

Comme détaillé à la section 3.1, lorsqu'on entraîne un modèle, il faut séparer les données. Pour ce travail, 16 données ont été utilisées pour l'entraînement du modèle et 4 données ont servi à la validation. Les valeurs de dice données dans la suite du travail correspondent au dice moyen sur les 4 données de validation et sur les différentes parties du coeur sauf si le contraire est précisé.

La suite de cette section va présenter les différents tests qui sont réalisés afin de créer et déterminer les paramètres du modèle. Pour chacun de ces tests, la vitesse d'apprentissage est évaluée en même temps qu'un autre paramètre. Nous utilisons 5 vitesses différentes : 0,00005, 0,0001, 0,0005, 0,001, 0,005.

5.2.1 Tests avec une résolution de 64 × 64 × 64

Cette section reprend tous les tests qui ont été réalisés avec des données de résolution 64 × 64 × 64. Avec cette résolution, nous avons testé l'influence de la fonction de coût, de la taille du groupe de données, du nombre de couches, du dropout, du nombre de neurones par couche ou encore d'un échauffement du modèle.

Test A64 : Segmentation de deux classes

Le premier test que nous avons réalisé, nous a permis de vérifier le fonctionnement de l'implémentation de notre modèle. Pour ce faire, nous avons regroupé les différentes classes pour former une seule tâche de segmentation binaire ce qui permet également de simplifier le modèle. En effet, il est plus compliqué de segmenter différentes classes qu'une seule sur la même image. Au plus il y a de classes, au plus le modèle doit identifier de caractéristiques et il faudra plus de cartes de caractéristiques. En conséquence, le modèle aura certainement besoin de plus de couches ou de plus de neurones par couche [55]. Le tableau 5.2 reprend les différents paramètres utilisés pour effectuer ce test.

Tableau 5.2 – Paramètres utilisés pour le test A64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	5
Nombre d'epochs	10
Vitesse d'apprentissage	0.001
Nombre de neurones dans la plus petite couche	32
Dropout	0
Échauffement	Non

Test B64 : Première segmentation avec les 7 classes

Une fois l'implémentation vérifiée, un test a été réalisé pour tenter de segmenter les 7 classes différentes. Pour rappel, chaque classe est associée à un numéro dans le tableau 5.1. Nous avons utilisé un nombre plus élevé d'epochs, nous en effectuons 50 au lieu de 10. Ce nombre plus élevé est également une conséquence du nombre de classes plus élevé qui rend l'entraînement plus compliqué. Le tableau 5.3 résume les différents paramètres utilisés pour effectuer ce test.

Tableau 5.3 – Paramètres utilisés pour le test B64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	5
Nombre d'epochs	50
Vitesse d'apprentissage	0.001
Nombre de neurones dans la plus petite couche	32
Dropout	0
Échauffement	Non

Test C64 : Choix de la fonction de coût

Le premier paramètre testé pour optimiser notre modèle est la fonction de coût. Nous testons les 3 fonctions présentées dans le chapitre 3, appelées dice, focal dice et log cosh dice. Pour réaliser nos tests, nous avons choisi des paramètres par défaut, ceux-ci vont évoluer au cours des tests. Les paramètres utilisés par défaut se trouvent dans le tableau 5.4.

Tableau 5.4 – Paramètres utilisés pour le test C64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0
Échauffement	Non

Test D64 : Influence de la taille du groupe

Nous avons testé l'influence de la taille du groupe. Puisque la quantité de données que nous avons est relativement limitée, nous avons également dû limiter les valeurs de taille de groupe possibles. Si nous donnons nos 16 données d'entraînement en une fois, le modèle a moins d'occasion pour modifier les poids des neurones et les résultats seront moins bons. Nous avons dès lors testé des tailles allant de 1 à 4. Cependant, nous savons également qu'une taille trop faible ne va pas produire un modèle assez général [56]. Grâce à ce test, nous allons pouvoir déterminer si la taille de 1 produit réellement un modèle moins général que la taille de 2. Les paramètres par défaut utilisés pour ce test se trouvent dans le tableau 5.5.

Tableau 5.5 – Paramètres utilisés pour le test D64

Paramètres	Valeurs
Taille des groupes	Variable
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0
Échauffement	Non

Test E64 : Influence du nombre de couches

Le paramètre suivant est le nombre de couches du modèle. Ce paramètre est testé pour déterminer la profondeur adéquate du modèle. La littérature montre que la qualité de l'image ainsi que sa taille influence le nombre de couches d'un réseau. Au plus la qualité est faible, au plus le réseau a besoin de couches supplémentaires pour effectuer la même tâche mais au plus l'image est grande, au plus il faut de couches [57]. En effet, si la résolution est plus élevée, une couche supplémentaire est nécessaire afin d'obtenir la même taille à la fin de l'étape encodeur du modèle. Le nombre de poids nécessaires dans le réseau est plus grand pour une résolution plus élevée. Pour ce test, nous faisons varier le nombre de couches entre 2 et 5. Le tableau 5.6 reprend les différents paramètres de ce test.

Tableau 5.6 – Paramètres utilisés pour le test E64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	Variable
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0
Échauffement	Non

Test F64 : Influence du dropout

Nous savons qu'ajouter du dropout permet théoriquement d'améliorer les performances d'un modèle et de réduire l'effet de sur-apprentissage. Cela est testé pour différents niveaux de dropout. Il est caractérisé par un ratio, nous avons testé son effet pour des valeurs comprises entre 0 et 0,4, sachant qu'une valeur de 0,4 est normalement trop élevée [58]. Les paramètres utilisés pour ce test sont repris dans le tableau 5.7.

Tableau 5.7 – Paramètres utilisés pour le test F64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	Variable
Échauffement	Non

Test G64 : Influence du nombre de neurones par couche

Nous avons également décidé de faire varier le nombre de neurones par couche. En fonction des résultats obtenus en ajoutant du dropout, cela pourrait montrer une influence relativement importante. Pour ce test, nous utilisons un dropout de 0,3 afin de quantifier l'impact du nombre de neurones par couche. Nous avons fait varier le nombre de neurones de la plus petite couche de 8 à 64, lors des tests précédents ce nombre était de 64. Le nombre de neurones des autres couches est adapté à la plus petite couche.

Nous n'avons pas pu tester le modèle avec 128 neurones dans la plus petite couche car les ressources de calculs n'étaient pas suffisantes. Le tableau 5.8 reprend les différents paramètres utilisés.

Tableau 5.8 – Paramètres utilisés pour le test G64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	Variable
Dropout	0,3
Échauffement	Non

Test H64 : Influence d'un échauffement du modèle

Le dernier paramètre que nous testons est l'échauffement du modèle. Il permet de modifier la vitesse d'apprentissage au cours du temps. En effet, des études ont montré que l'utilisation d'une phase d'entraînement puis une diminution progressive de la vitesse d'apprentissage permet d'éviter la création d'instabilité dans les couches les plus profondes [59]. Nous avons testé des périodes d'échauffement allant de 0 à 50 epochs. Les paramètres d'entraînement se trouvent dans le tableau 5.9.

Tableau 5.9 – Paramètres utilisés pour le test H64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

5.2.2 Tests avec une résolution de 128 × 128 × 128

Cette section reprend tous les tests réalisés avec des données de résolution 128 × 128 × 128. Avec cette résolution, nous avons testé l'influence de la taille du groupe de données, du nombre de couches et du dropout. Pour ces tests, un dropout de 0,3 est déjà présent, le nombre de couches a également été augmenté à 4 par défaut car les images ont une taille plus élevée.

Test D128 : Influence de la taille du groupe

Afin de confirmer l'influence de la taille du groupe, nous avons à nouveau réalisé ce test avec les données de résolution supérieure. Nous avons de nouveau fait varier la taille du groupe de 1 à 4. Les paramètres par défaut sont repris dans le tableau 5.10.

Tableau 5.10 – Paramètres utilisés pour le test D128

Paramètres	Valeurs
Taille des groupes	Variable
Nombre de couches	4
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

Test E128 : Influence du nombre de couches

Comme expliqué au test E64, la qualité et la taille des images influencent le nombre de couches du réseau. Dans ce cas, nos images sont de meilleure qualité mais sont plus grandes. Il est donc nécessaire d'évaluer à nouveau le nombre de couches adéquates. Comme pour la résolution précédente, nous avons testé un nombre de couches allant de 2 à 5. Le tableau 5.11 résume les paramètres d'entraînement.

Tableau 5.11 – Paramètres utilisés pour le test E128

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	Variable
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

Test F128 : Influence du dropout

Tous les tests précédents ont été réalisés avec un dropout de 0,3. Nous avons testé des valeurs plus faibles en le faisant varier de 0 à 0,3. Les paramètres d'entraînement sont disponibles dans le tableau 5.12.

Tableau 5.12 – Paramètres utilisés pour le test F128

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	4
Nombre d'epochs	200
Vitesse d'apprentissage	Variable
Nombre de neurones dans la plus petite couche	64
Dropout	Variable
Échauffement	Non

5.2.3 Test I128 : Modèle pré-entraîné

L'utilisation d'un modèle pré-entraîné est également une technique qui a été testée dans ce travail. Dans notre cas, il s'agit en réalité d'apprentissage par transfert, en anglais

Transfert Learning. Cette méthode consiste à transférer les connaissances acquises en résolvant un problème donné à un autre problème, ici on résout le problème avec une résolution de $64 \times 64 \times 64$ et on utilise le savoir récolté pour résoudre le problème avec une résolution de $128 \times 128 \times 128$. La tâche à effectuer est la même mais les données diffèrent par leurs résolutions [60]. La figure 5.3 montre un schéma expliquant cette méthode dans la pratique.

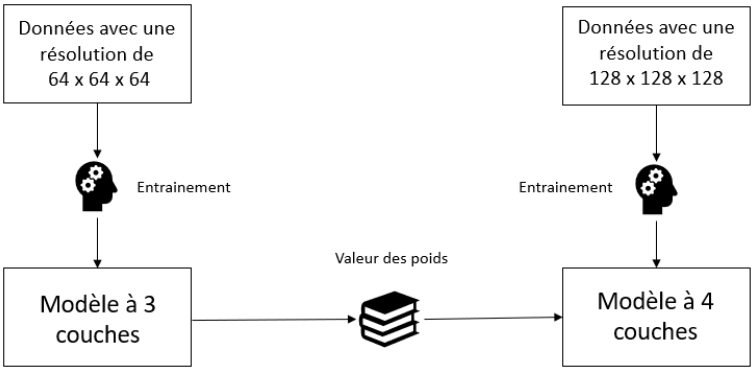


FIGURE 5.3 – Apprentissage par transfert. Le premier modèle à 3 couches est entraîné avec les données de résolution $64 \times 64 \times 64$. La valeur des poids calculée durant cet entraînement est transférée dans les poids correspondant du modèle à 4 couches. Les données de résolution $128 \times 128 \times 128$ sont utilisées pour entraîner le modèle à 4 couches.

Dans la pratique, nous avons fait un entraînement avec la résolution $64 \times 64 \times 64$ en utilisant la meilleure combinaison de paramètres pour entraîner ce modèle. Cette combinaison sera déterminée grâce aux résultats des tests réalisés en utilisant cette résolution. Ensuite, nous avons utilisé les poids calculés lors de l'entraînement de ce modèle pour initialiser les poids correspondants dans le second modèle, celui-ci comporte une couche supplémentaire. Finalement, nous avons entraîné ce second modèle grâce aux données avec une résolution de $128 \times 128 \times 128$. Lors de celui-ci, les poids copiés n'ont été modifiés qu'à partir de la deuxième epoch.

5.2.4 Test J128 : Augmentation de la base de données

La quantité de données annotées dont nous disposons est relativement faible. Pour rappel, nous ne disposons que de 16 données pour entraîner notre modèle. En conséquence, il pourrait être trop précis sur les données d'entraînement et ne pas bien généraliser la segmentation sur les données de validation dont la quantité est également faible.

Nous avons implémenté une augmentation de la quantité de données de manière artificielle. Pour ce faire, nous avons pris chaque donnée de départ et appliqué une transformation sur celle-ci. Nous avons effectué trois types de transformation ; des retournements, des rotations de 90° ou encore des ajouts de bruit gaussien en appliquant un écart type de 0,1. Chaque type de transformations est effectué une fois sur chaque donnée initiale. Si nous appliquons 5 transformations alors 5 nouvelles données sont créées.

La figure 5.4 permet de vérifier qu'on obtient toujours des images fidèles à la réalité, c'est-à-dire qu'il est toujours possible de reconnaître les différentes parties du coeur. La figure montre trois transformations différentes, nous n'observons pas de différence visuelle pour l'ajout de bruit gaussien car celui-ci est faible.

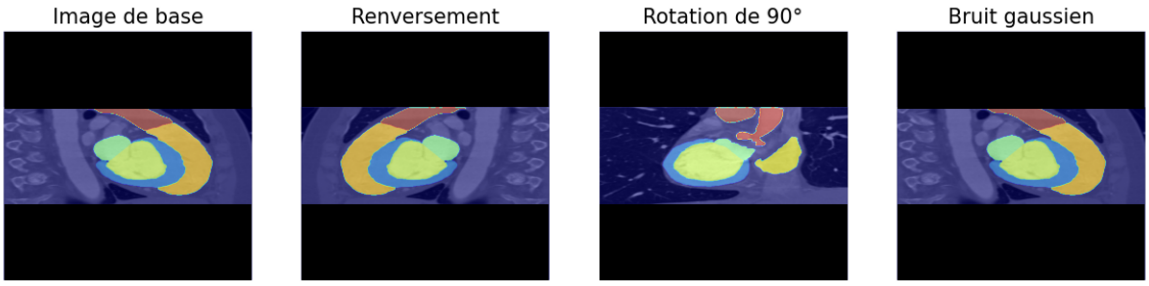


FIGURE 5.4 – Exemple d'images obtenues avec l'augmentation de la base de données. Il y a respectivement de gauche à droite : l'image de base, un exemple de renversement, un exemple de rotation et un exemple d'ajout de bruit gaussien.

Après toutes ces déformations, le nombre de données d'entraînement s'élève à 112 et celui de validation à 28. Grâce à celles-ci de nouveaux tests ont pu être réalisés, en faisant varier divers hyper-paramètres, ces tests se basent sur les mêmes paramètres que les tests D64 à H64 et des tests D128 à F128. Comme pour l'entraînement sur la base de données initiale, les tests ont d'abord été réalisés avec une résolution de 64 $\times$ 64 $\times$ 64 et ensuite avec la résolution supérieure. Des essais avec un modèle pré-entraîné ont également été réalisés.

5.2.5 Test K128 : Standardisation des données

Lors de la création d'un modèle en intelligence artificiel, il est courant de standardiser les données. Nous avons donc standardisé nos données pour tenter d'obtenir un modèle plus général. Si les données ont des différences dans les statistiques d'intensité des pixels, le modèle va peut-être mieux apprendre sur certaines données et aura de moins bonnes performances sur d'autres données. Une première étape a du être réalisée avant de réellement standardiser les données. En effet, comme nous pouvons le voir sur la figure 5.2, il existe un cercle sur les images transversales et les pixels extérieurs à ce cercle ont des valeurs très faibles mais celles-ci ne se trouvent pas réellement dans le coeur. Nous avons donc calculé la valeur minimale des pixels dans une image ne contenant pas ces bords extérieurs. Ensuite, toutes les valeurs en dessous de ce minimum dans l'image contenant les bords extérieurs ont été modifiées. Nous leur avons donné la valeur minimale des pixels de l'image sans les bords extérieurs.

Une fois cette opération effectuée, la standardisation a été réalisée en suivant l'équation 5.1. Le but de la standardisation est d'avoir une moyenne de 0 et une variance de 1.

$$V(i,j) = (V(i,j) - m)/std \tag{5.1}$$

Où V(i,j) correspond à la valeur du pixel à la position (i,j), m correspond à la valeur moyenne des pixels et std représente l'écart type.

Une fois la standardisation effectuée, nous pouvons pré-entraîner notre modèle avec la résolution 64 × 64 × 64 et ensuite l'entraîner avec la résolution 128 × 128 × 128 afin qu'il puisse effectuer la segmentation voulue. Les paramètres d'entraînement sont ceux donnant les meilleurs résultats de ce chapitre, ils se trouvent dans les tableaux 5.19 et 5.23.

Puisque les données d'entraînement ont été modifiées, nous avons voulu voir si les résultats obtenus visuellement et en terme de dice sont différents sur les données de validation.

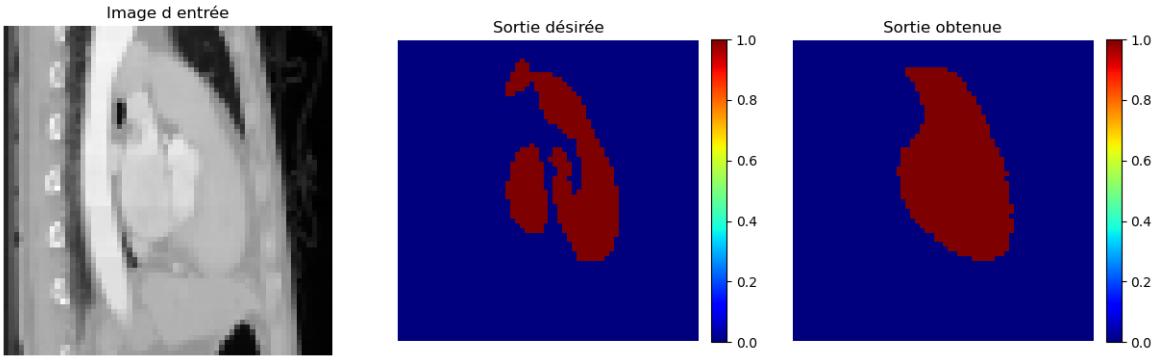
5.3 Résultats et discussion

Cette section permet de présenter les différents résultats obtenus à la suite des tests détaillés à la section précédente. Pour rappel, l'objectif est d'atteindre un dice minimum de 0,7 pour toutes les parties à segmenter.

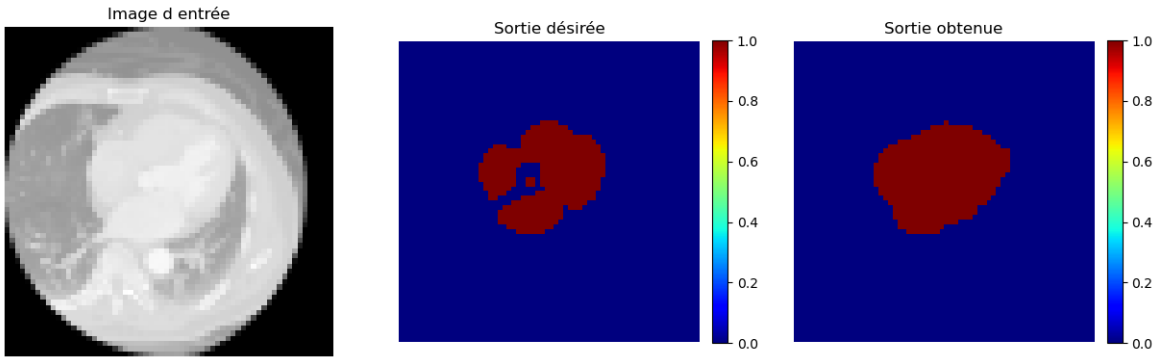
5.3.1 Résultats des tests avec une résolution de 64 × 64 × 64

Résultats du test A64 : Segmentation de deux classes

Ce test s'est avéré concluant avec un dice de validation moyen de 0,86 et un coût final de 0,19. La figure 5.5 montre le résultat visuel selon les vues sagittale et transversale de ce test avec à gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle. Cette figure montre qu'il persiste des différences entre la sortie désirée et celle obtenue. En effet, la sortie obtenue ne prédit qu'une seule tache au lieu de plusieurs petits groupes. Cela est certainement dû à la simplification des données à une seule classe.



(a) Vue sagittale



(b) Vue transversale

FIGURE 5.5 – *Visualisation du résultat obtenu au test A64. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.*

Sur les courbes d'apprentissage à la figure 5.6, nous devinons un début de surapprentissage car aucune amélioration n'est observée sur le dice moyen pour les deux dernières epochs, au contraire il diminue. L'image de gauche montre l'évolution du coût moyen en fonction de l'epoch. Tandis que celle de droite correspond à l'évolution du dice moyen sur les données de validation en fonction de l'epoch.

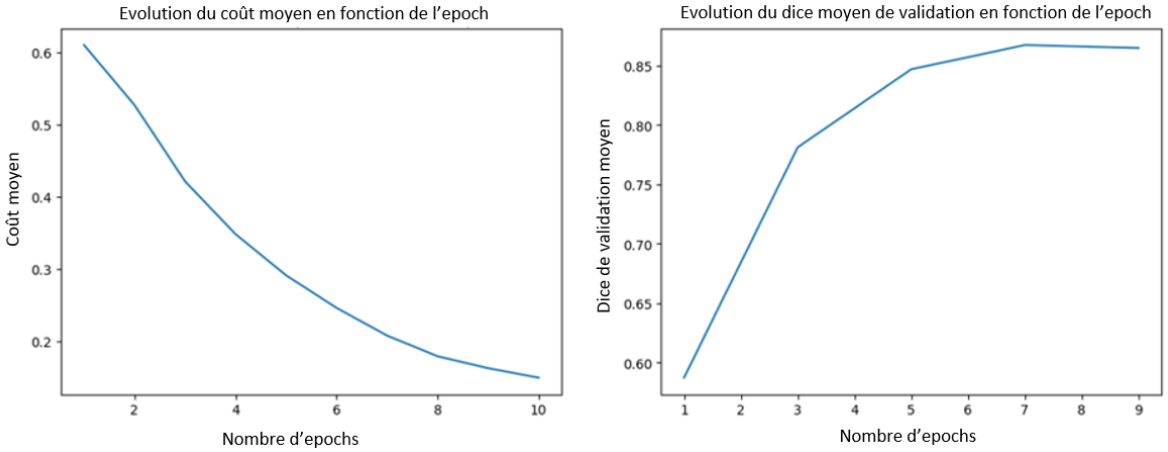
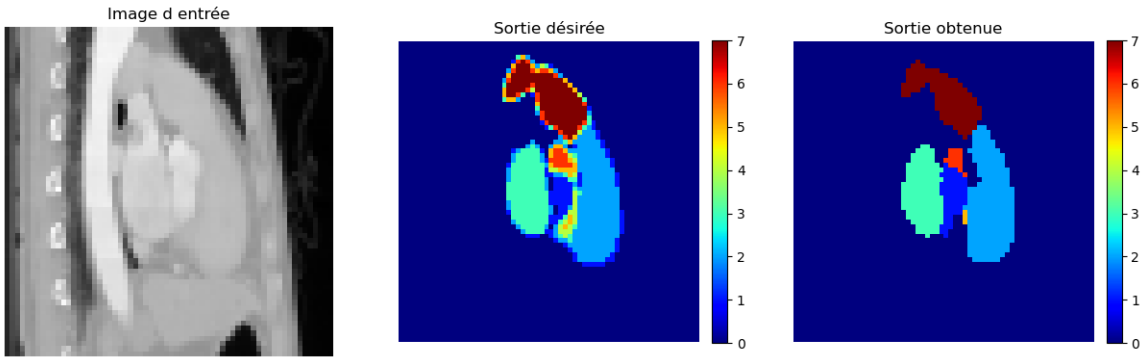


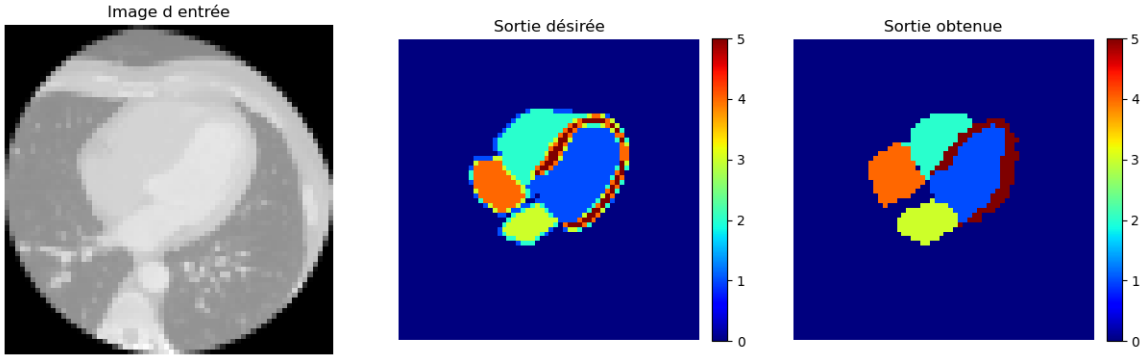
FIGURE 5.6 – *Courbes d'apprentissage obtenues au test A64. A gauche, l'évolution du coût moyen en fonction de l'epoch. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.*

Résultats du test B64 : Première segmentation avec les 7 classes

Dans ce cas, un dice moyen de 0,52 est obtenu avec un coût final de 0,21. Le résultat visuel selon les vues sagittale et transversale se trouve à la figure 5.7. Cette figure montre une séparation entre les différentes classes et non pas une seule forme compacte comme à la figure 5.5. Pour rappel, la correspondance des valeurs allant de 0 à 7 se trouve dans le tableau 5.1.



(a) Vue sagittale



(b) Vue transversale

FIGURE 5.7 – Visualisation du résultat du test B64. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.

Les courbes d'apprentissage sur la figure 5.8 montrent également un début de stagnation des résultats à la fin des epochs. Malgré la différence de dice obtenue entre les deux tests, nous observons une similarité dans le nombre d'epochs nécessaires à l'apprentissage et donc avant la stagnation du modèle. Le test précédent a stagné avec 10 epochs et ce test avec 15 epochs.

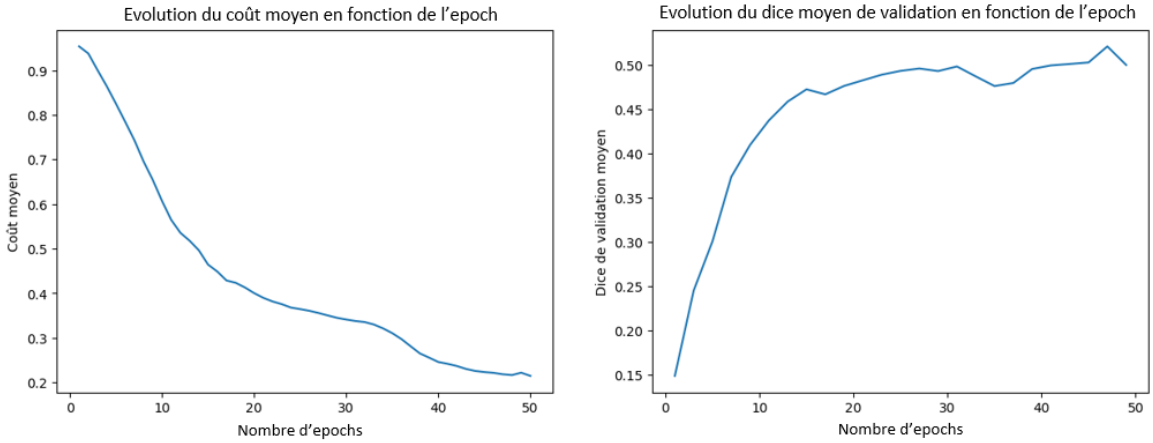


FIGURE 5.8 – Courbes d'apprentissage obtenues au test B64. A gauche, l'évolution du coût moyen en fonction de l'epoch. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.

La figure 5.9 montre l'évolution du dice pour chaque partie du coeur en fonction des epochs. Grâce à ces courbes, nous pouvons observer que certaines parties à segmenter obtiennent de meilleurs résultats comme l'aorte en brun ou l'oreillette droite en rouge. Au contraire, avec cet entraînement, la segmentation de l'artère pulmonaire en rose obtient un dice très faible de 0,3 alors qu'il est compris entre 0,4 et 0,5 pour les autres parties.

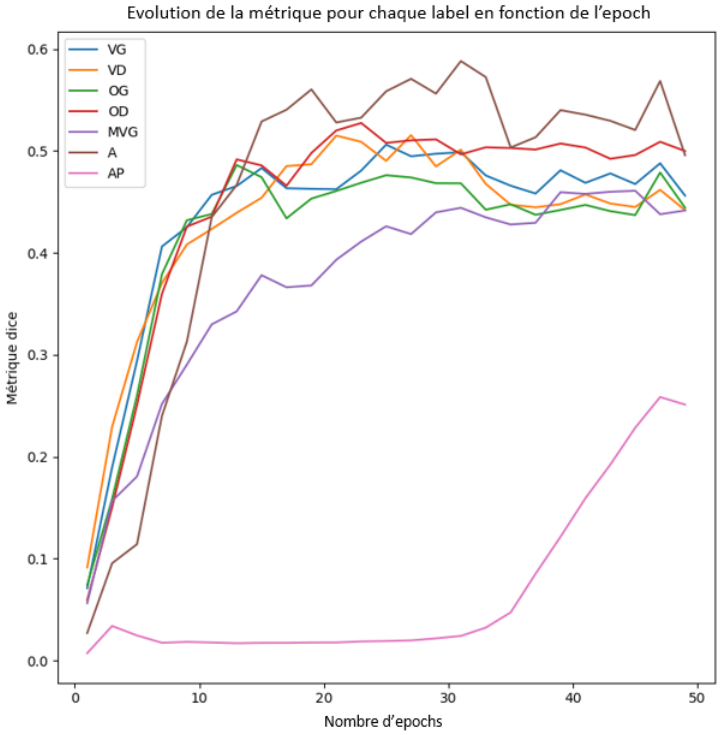


FIGURE 5.9 – Courbes des dices pour chaque partie du coeur au test B64. Chaque ligne correspond à une partie segmentée, la légende fait le lien entre ces parties.

Résultats du test C64 : Choix de la fonction de coût

Le tableau 5.13 montre les résultats de dice obtenus pour chaque test selon les trois fonctions de coût. Nous tenons compte du meilleur dice obtenu au cours des 200 epochs réalisées.

TABLEAU 5.13 – Résultats du test C64

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Dice	0,52	0,56	0,59	0,57	0,44
Focal dice	0,66	0,68	0,70	0,70	0,70
Log cosh dice	0,49	0,51	0,59	0,58	0,46

Nous pouvons voir que les meilleurs résultats, en vert, sont obtenus avec la fonction de coût focal dice. Nous observons également que la fonction produisant les moins bons résultats est la simple fonction de coût dice. La théorie vue au chapitre 3 explique ce résultat. En effet, la fonction focal dice donne toujours de meilleurs résultats que la fonction dice. Nous avons également vu que la fonction de coût dépend des données à traiter, de ce qu'elles contiennent comme information. Cette fonction de coût est donc utilisée pour la suite des tests et même avec une résolution plus élevée car le changement de résolution n'impacte pas le type d'information que les données contiennent [61].

Résultats du test D64 : Influence de la taille du groupe

Un récapitulatif des résultats obtenus est disponible dans le tableau 5.14.

Tableau 5.14 – Résultats du test D64

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Taille = 1	0,68	0,69	0,70	0,70	0,70
Taille = 2	0,66	0,68	0,70	0,70	0,70
Taille = 3	0,63	0,68	0,68	0,69	0,70
Taille = 4	0,59	0,65	0,68	0,70	0,70

Nous remarquons que la taille du groupe a plus d'influence lorsque la vitesse d'apprentissage est faible. En effet, les valeurs de dice obtenues pour une vitesse de 0,0005 varie entre 0,59 en rouge et 0,68 alors que pour une vitesse de 0,005, la valeur de 0,7 en vert est atteinte peu importe la taille du groupe. Nous observons également qu'au plus le groupe est grand, au plus le dice sera faible pour les vitesses les plus basses. Si on se base uniquement sur ce tableau, une taille de groupe de 1 serait choisie pour la suite des tests. Cependant, la différence entre le dice sur les données d'entraînement et celui sur les données de validation est plus élevée pour un groupe de taille 1 par rapport à la taille 2 qui a des résultats presque similaires. Les tests réalisés avec une vitesse d'apprentissage de 0,001, pour des tailles de groupe de 1 et de 2, donnent tous les deux un dice de 0,7. Le dice obtenu sur les données d'entraînement est de 0,88 pour la taille de groupe de 1, tandis qu'il est de 0,84 pour la taille de 2. Cela montre une meilleure généralisation du modèle avec une taille de groupe de 2 alors que le même résultat est atteint sur les données de validation. C'est donc la taille de groupe de 2 qui est choisie pour la suite des tests.

Résultats du test E64 : Influence du nombre de couches

Le tableau 5.15 résume les résultats obtenus au test E64.

Tableau 5.15 – Résultats du test E64

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
2 couches	0,35	0,44	0,58	0,6	0,62
3 couches	0,68	0,69	0,70	0,70	0,70
4 couches	0,6	0,62	0,64	0,65	0,66
5 couches	0,5	0,51	0,56	0,60	0,61

Ce tableau nous permet d'observer que le nombre de couches adéquat est de 3. Si nous observons les colonnes pour chaque vitesse d'apprentissage, le dice le plus élevé est à chaque fois obtenu lorsque le modèle possède trois couches. Comme précédemment expliqué, ce nombre doit être un compromis entre la taille de l'image et sa qualité.

Ce paramètre sera de nouveau analysé pour la résolution supérieure car elle a un impact sur le nombre de couches d'un modèle.

Résultats du test F64 : Influence du dropout

Les différents résultats obtenus sont rassemblés dans le tableau 5.16.

Tableau 5.16 – Résultats du test F64

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Dropout = 0	0,68	0,69	0,70	0,70	0,70
Dropout = 0,1	0,65	0,69	0,71	0,71	0,71
Dropout = 0,2	0,63	0,67	0,71	0,71	0,72
Dropout = 0,3	0,61	0,67	0,72	0,72	0,72
Dropout = 0,4	0,72	0,67	0,59	0,73	0,73

Ces résultats permettent de confirmer l'amélioration des résultats avec un dropout pour des vitesses d'apprentissage plus élevées, à partir de 0,0005. Pour des vitesses plus faibles, une légère diminution survient. Le meilleur dice atteint est de 0,73 avec un dropout de 0,4. Cependant, le dice le plus faible est également atteint avec le dropout de 0,4. Effectivement, avec la vitesse d'apprentissage de 0,0005, le dice augmente avec le dropout mais lorsque celui-ci est de 0,4, le dice chute radicalement à 0,59 alors qu'il est de 0,72 avec un dropout de 0,3. Un dropout de 0,3 est dès lors gardé pour la suite des tests. Celui-ci montre une évolution plus constante des résultats par rapport à un dropout de 0,4.

Résultats du test G64 : Influence du nombre de neurones par couche

Le tableau 5.17 rassemble les résultats obtenus au test G64.

Tableau 5.17 – Résultats du test G64

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Couche = 8 neurones	0,18	0,18	0,43	0,55	0,63
Couche = 16 neurones	0,19	0.32	0,65	0,66	0,69
Couche = 32 neurones	0,42	0,57	0,69	0,70	0,72
Couche = 64 neurones	0,61	0,67	0,72	0,72	0,72

Ce test est effectué afin de vérifier que le nombre de neurones par couche est adéquat en diminuant ce nombre. En regardant les résultats, nous observons que pour chaque vitesse d'apprentissage, le meilleur dice est obtenu lorsqu'il y a 64 neurones dans la plus petite couche. Cela veut dire qu'il faut utiliser ce nombre lorsqu'un dropout de 0,3 est appliqué. Des tests supplémentaires seraient nécessaires pour déterminer le nombre de neurones par couches avec un dropout plus faible. En ce qui concerne la vitesse d'apprentissage, nous remarquons que les meilleurs résultats sont atteints lorsque celle-ci est plus élevée. Nous aurions également pu faire des tests supplémentaires avec des vitesses plus élevées.

Résultat du test H64 : Influence d'un échauffement du modèle

Les résultats obtenus sont rassemblés dans le tableau 5.18.

Tableau 5.18 – Résultats du test H64

Vitesses d'apprentissage maximum	0,00005	0,0001	0,0005	0,001	0,005
Pas d'échauffement	0,61	0,67	0,72	0,72	0,72
Échauffement sur 10 epochs	0,44	0.60	0,67	0,71	0,71
Échauffement sur 25 epochs	0,41	0,60	0,70	0,71	0,72
Échauffement sur 50 epochs	0,42	0,60	0,70	0,71	0,72

Les résultats ne sont pas beaucoup impactés pour de hautes vitesses d'apprentissage. Cependant pour des vitesses plus faibles, le dice diminue lorsque de l'échauffement est présent. Ces résultats ne sont pas tout à fait en accord avec la théorie supposée. Nous aurions du effectuer des tests avec des vitesses plus élevées. Dans ce cas, la présence d'un échauffement aurait peut être donné de meilleurs résultats.

Puisque que le dice est toujours supérieur ou similaire lorsqu'aucun échauffement est appliqué, les paramètres par défaut des tests resteront les mêmes pour la suite des tests, c'est-a-dire sans échauffement.

Meilleure combinaison de paramètres avec la résolution 64 × 64 × 64

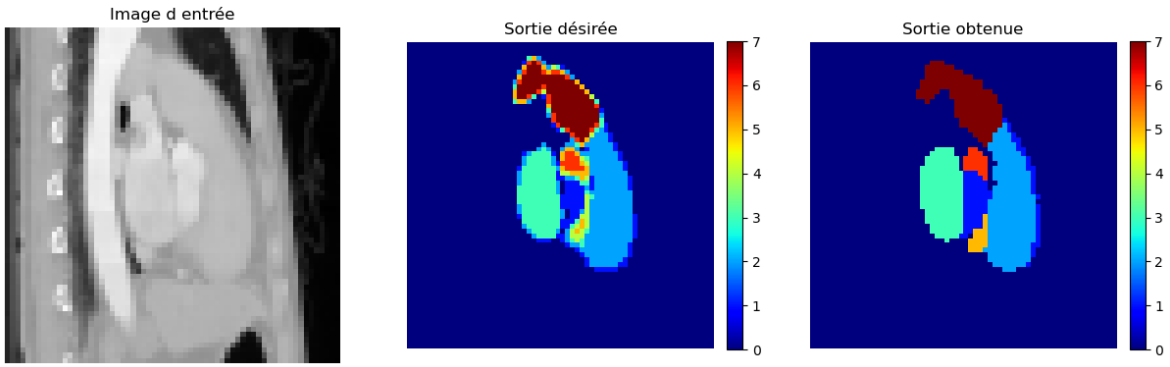
Une fois tous nos hyper-paramètres testés, il faut analyser les résultats pour déterminer la meilleure combinaison de paramètres afin d'avoir des résultats optimisés.

Si nous observons toutes les valeurs de dice obtenues, la meilleure est de 0,72. Cependant, cette valeur a été obtenue pour plusieurs combinaisons de paramètres dans les tests F64, G64 et H64. Pour choisir correctement la meilleure d'entre-elles, nous avons décidé de garder celle produisant la plus petite différence entre la valeur de dice sur les données d'entraînement et sur les données de validation. Ainsi nous gardons le modèle le plus général. Les paramètres retenus sont repris dans le tableau 5.19.

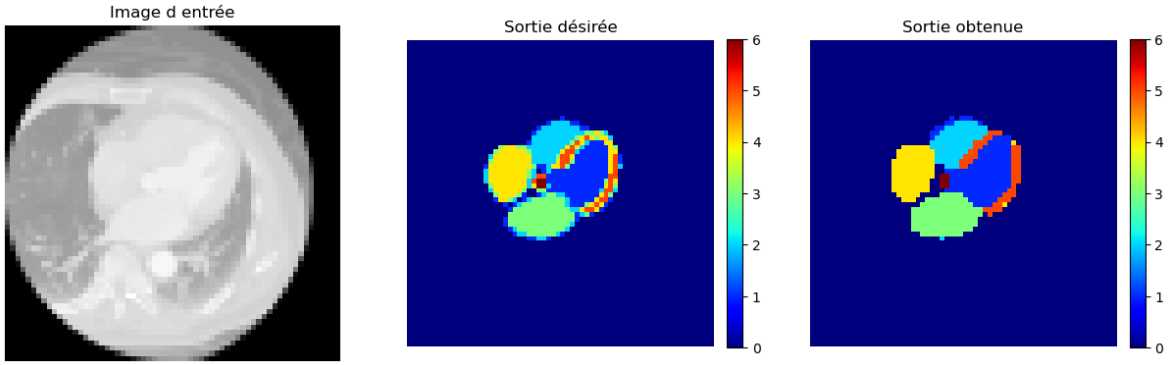
Tableau 5.19 – Meilleure combinaison de paramètres pour la résolution 64 × 64 × 64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	0,0005 et constante
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

La figure 5.10 montre le résultat visuel selon les vues sagittale et transversale de cette combinaison de paramètres. Cette figure permet de voir que les détails de la segmentation et plus particulièrement des contours de chaque partie du coeur sont plus proches de ce qui est attendu par rapport à ce qu'on pouvait observer sur la figure 5.7.



(a) Vue sagittale



(b) Vue transversale

FIGURE 5.10 – *Visualisation du résultat avec la meilleure combinaison de paramètres pour la résolution 64 × 64 × 64. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.*

Les courbes d'apprentissages présentées à la figure 5.11 montrent que peu de sur-apprentissage est atteint malgré l'aplatissement des courbes à partir de 75 epochs car le coût continue de diminuer et le dice d'augmenter sur les données de validation.

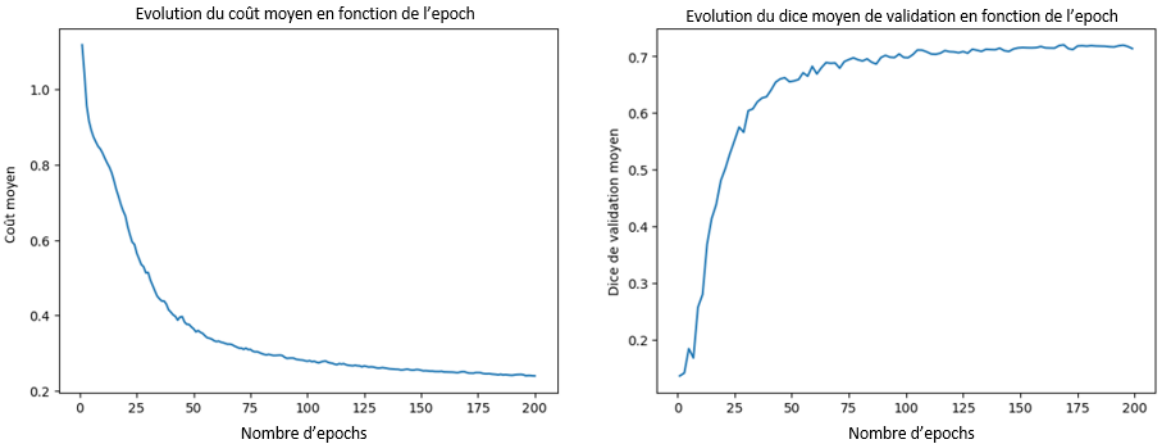


FIGURE 5.11 – *Courbes d'apprentissage obtenues avec la meilleure combinaison de paramètres pour la résolution 64 × 64 × 64. A gauche, l'évolution du coût moyen en fonction de l'epoch. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.*

Le dice atteint sur les données de validation est de 0,72 tandis que que celui atteint sur les données d'entraînement est de 0,77. Bien que l'objectif d'un dice minimum de 0,7 est atteint, il faut vérifier que cela est également le cas pour chaque partie du coeur à segmenter.

La figure 5.12 montre l'évolution de chacun de ces dices en fonction de l'epoch. Sur ce graphe, nous observons que deux parties uniquement atteignent au minimum un dice de 0,7. Il s'agit de l'aorte qui avait également le meilleur dice au test B64. Cependant la deuxième partie est l'artère pulmonaire qui avait le dice le plus faible au test B64. Concernant cette partie, nous remarquons également que son dice est de 0 sur les 20 premières epochs. Puisque seulement 50 epochs étaient réalisées au test B64, cela peut expliquer la grande différence de performance sur cette partie du coeur. Pour ce qui est des autres parties, elles atteignent toutes une valeur de plus ou moins 0,6, ce qui n'est pas suffisant.

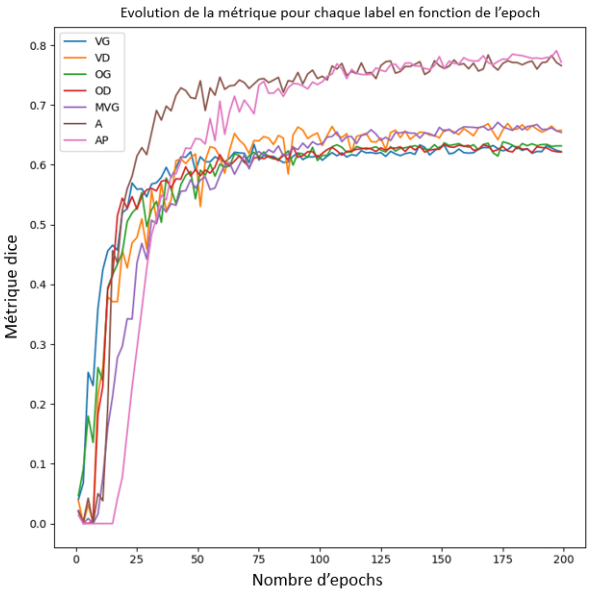


FIGURE 5.12 – Courbes des dices pour chaque partie du coeur avec la meilleure combinaison de paramètres pour la résolution 64 × 64 × 64. Chaque ligne correspond à une partie segmentée, la légende fait le lien entre ces parties.

Nous pouvons comparer ces résultats avec ceux exposés dans le chapitre 4. Nous avions montré les dices obtenus pour six articles différents. Chacun d'eux obtenait un résultat supérieur au dice de 0,72 que nous atteignons. Cependant, si nous observons les dices obtenus pour chaque partie du coeur, nous remarquons que dans la moitié des articles présentés, les dices obtenus pour l'artère pulmonaire et l'aorte sont inférieurs à 0,70. Avec notre modèle, ces dices atteignent 0,80.

5.3.2 Résultats des tests avec une résolution de 128 × 128 × 128

Résultats du test D128 : Influence de la taille du groupe

Les différentes valeurs de dice obtenues pour les différentes combinaisons de taille du groupe et de vitesse d'apprentissage se trouvent dans le tableau 5.20.

TABLEAU 5.20 – Résultats du test D128

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Taille = 1	0,64	0,64	0,67	0,66	0,61
Taille = 2	0,60	0,62	0,66	0,66	0,62
Taille = 3	0,58	0,61	0,63	0,64	0,60
Taille = 4	0,56	0,61	0,64	0,64	0,60

Comme pour la résolution de 64 × 64 × 64 évaluée au test D64, les résultats nous montrent un meilleur dice pour une taille de groupe de 1. Cependant, nous pouvons faire les mêmes observations quant au sur-apprentissage. Si l'on compare le dice obtenu avec une taille de 1 ou de 2 avec une vitesse d'apprentissage de 0,001, ils sont tous les deux de 0,66. Cependant, le dice sur les données d'entraînement pour la taille de 1 est de 0,71 alors que celui pour la taille de 2 est de 0,68. Encore une fois, nous observons que le modèle avec une taille de 2 est plus général. Cette observation se rapporte bien à la théorie expliquée dans la méthode.

Nous pouvons également remarquer qu'avec cette résolution, les meilleurs dices sont obtenus avec la vitesse d'apprentissage de 0,0005. Cette observation n'est pas la même que celle faite avec la résolution inférieure.

Résultats du test E128 : Influence du nombre de couches

Les résultats obtenus se trouvent dans le tableau 5.21.

Tableau 5.21 – Résultats du test E128

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
2 couches	0,18	0,2	0,3	0,32	0,28
3 couches	0,40	0,52	0,59	0,61	0,58
4 couches	0,60	0,62	0,66	0,66	0,62
5 couches	0,45	0,53	0,56	0,53	0,52

Grâce à ces résultats, nous remarquons que le nombre de couches adéquat pour ce modèle est quatre. Nous savons que le nombre de couches nécessaires dépend principalement de la qualité de l'image et de sa taille. Pour la résolution 64 × 64 × 64, le test E64 a permis de déterminer le nombre de couches à trois pour ce modèle. En passant à la résolution 128 × 128 × 128, bien que la qualité de l'image s'est améliorée, sa taille est huit fois plus grande. Par conséquent, le nombre de couches nécessaires est plus élevé.

Résultats du test F128 : Influence du dropout

Les résultats sont rassemblés dans le tableau 5.22.

Tableau 5.22 – Résultats du test F128

Vitesses d'apprentissage	0,00005	0,0001	0,0005	0,001	0,005
Dropout = 0,0	0,53	0,56	0,58	0,56	0,56
Dropout = 0,1	0.58	0,62	0,63	0,64	0,58
Dropout = 0,2	0,60	0,63	0,65	0,65	0,60
Dropout = 0,3	0,60	0,62	0,66	0,66	0,62

Ces résultats confirment l'utilisation d'un dropout de 0,3. En effet, peu importe la vitesse d'apprentissage, la valeur du dice augmente lors que le dropout augmente aussi. Cette observation est différente de celle faites avec la résolution inférieure. Effectivement, lors du test F64, pour de faibles vitesses d'apprentissage, le dice diminuait légèrement lorsque le dropout augmentait. Nous remarquons également que ce n'est pas la plus grande vitesse d'apprentissage qui produit le meilleur résultat mais bien les vitesses de 0,0005 et 0,001.

Meilleure combinaison de paramètres avec la résolution 128 × 128 × 128

Après l'analyse de ces résultats, nous pouvons identifier les hyper-paramètres donnant le meilleur dice pour la résolution 128 × 128 × 128. Ceux-ci sont disponibles dans le tableau 5.23. Un dice de validation de 0,66 est atteint, celui d'entraînement est de 0,68, ce qui montre peu de sur-apprentissage et un modèle assez général. Le coût à la dernière epoch est de 0,11. Nous pouvons remarquer que les combinaisons des meilleurs paramètres pour les deux résolutions sont les mêmes mis à part le nombre de couches qui est plus élevé pour la plus grande résolution.

Tableau 5.23 – Meilleure combinaison de paramètres avec la résolution de 128 × 128 × 128

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	4
Nombre d'epochs	200
Vitesse d'apprentissage	0,0005 et constante
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

Nous pouvons observer le résultat visuel obtenu en utilisant cette combinaison d'hyper-paramètres à la figure 5.13. Cette figure comprend différentes vues et les couches représentées sont celles du centre de masse du patient. Les figures mettent bien en avant que le résultat de dice final est inférieur à celui obtenu pour la figure 5.10. En effet, les contours des différentes parties du coeur ne sont pas bien délimités malgré la présence des différentes couleurs censées se trouver dans les images. On peut également soulever l'absence de continuité dans les différentes formes.

(a) Vue sagittale

(b) Vue transversale

FIGURE 5.13 – *Visualisation du résultat avec la meilleure combinaison de paramètres avec la résolution de 128 × 128 × 128. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.*

Les courbes d'apprentissage relatives à ce test sont présentes à la figure 5.14. On peut y observer un ralentissement de l'apprentissage à partir de 75 epochs, cependant le modèle continue à apprendre presque jusqu'à la fin des epochs.

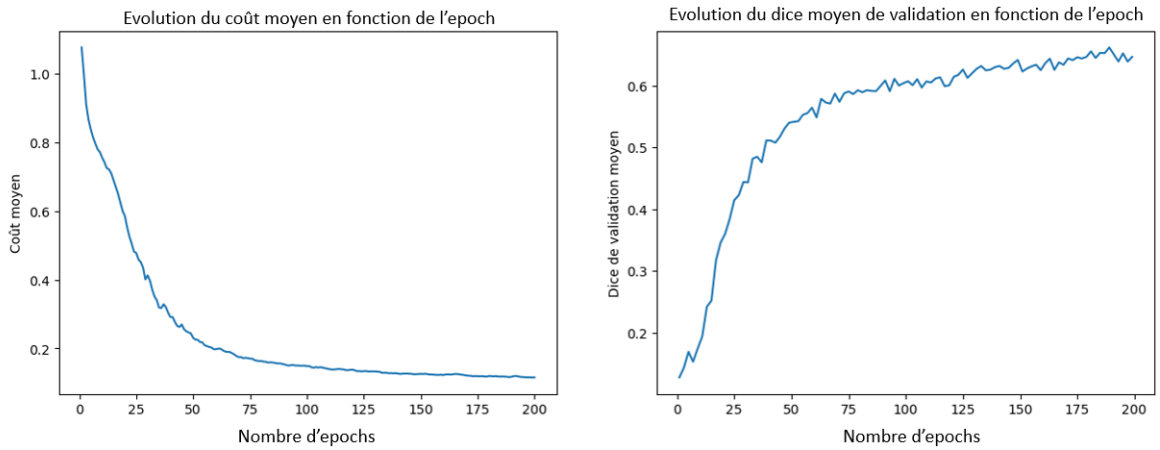


FIGURE 5.14 – *Courbes d'apprentissage obtenues avec la meilleure combinaison de paramètres avec la résolution 128 × 128 × 128. A gauche, l'évolution du coût moyen en fonction de l'epoch. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.*

Pour rappel, l'objectif est d'atteindre un dice de minimum 0,7. Ce dice n'est pas atteint en moyenne. La figure 5.15 permet de voir si certaines parties atteignent ce dice. Encore une fois l'aorte obtient le dice le plus élevé, celui-ci est supérieur à 0,7. Le second meilleur dice est celui de l'artère pulmonaire qui atteint difficilement le 0,7. Cependant, cela permet de confirmer l'hypothèse faite précédemment. Le dice de l'artère pulmonaire est encore de 0 sur les 20 premières epochs. La présence d'un dice très faible pour cette partie au test B64 est du à un manque d'epoch.

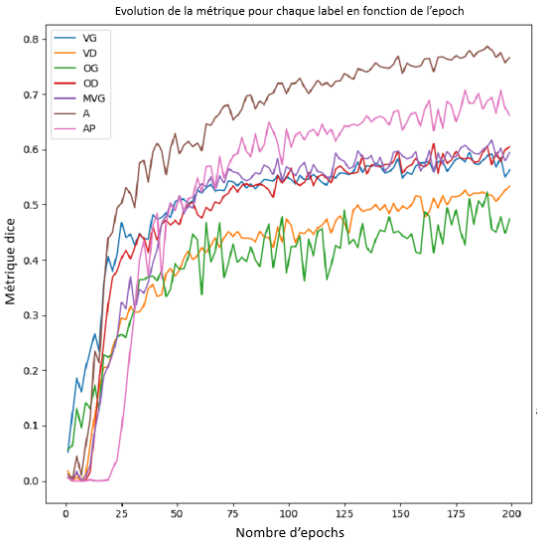


FIGURE 5.15 – Courbes d'apprentissage obtenues pour chaque classe avec la meilleure combinaison de paramètres pour la résolution 128 × 128 × 128. Chaque ligne correspond à une partie segmentée, la légende fait le lien entre ces parties.

Encore une fois, nous pouvons comparer ces résultats à ceux présentés au chapitre 4. Les mêmes observations peuvent être faites quant au dice moyen sur le coeur entier, notre dice de 0,66 est inférieur à ceux obtenus dans les six articles. L'aorte et l'artère pulmonaire restent encore une exception dans ces résultats puisqu'elles obtiennent respectivement un dice de 0,80 et 0,72.

5.3.3 Résultats du test I128 : Modèle pré-entraîné

Grâce à cette technique, le dice de 0,66 obtenu précédemment est passé à 0,7, ce qui est l'objectif posé au départ. La figure 5.16 montre le résultat visuel obtenu sur le même patient que pour le test précédent. Nous y observons que la segmentation est meilleure. En effet, les contours sont plus proches de ce qui est attendu, les différentes formes présentent un peu plus de continuité que sur les résultats précédents.

(a) Vue sagittale

(b) Vue transversale

FIGURE 5.16 – *Visualisation du résultat obtenu en utilisant un modèle pré-entraîné. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.*

Les courbes d'apprentissage montrent bien que l'objectif en terme de dice moyen est atteint à la figure 5.17. Bien que celles-ci atteignent un plateau, le dice continue d'augmenter. Cependant, le résultat visuel montre toujours des imperfections dans la segmentation.

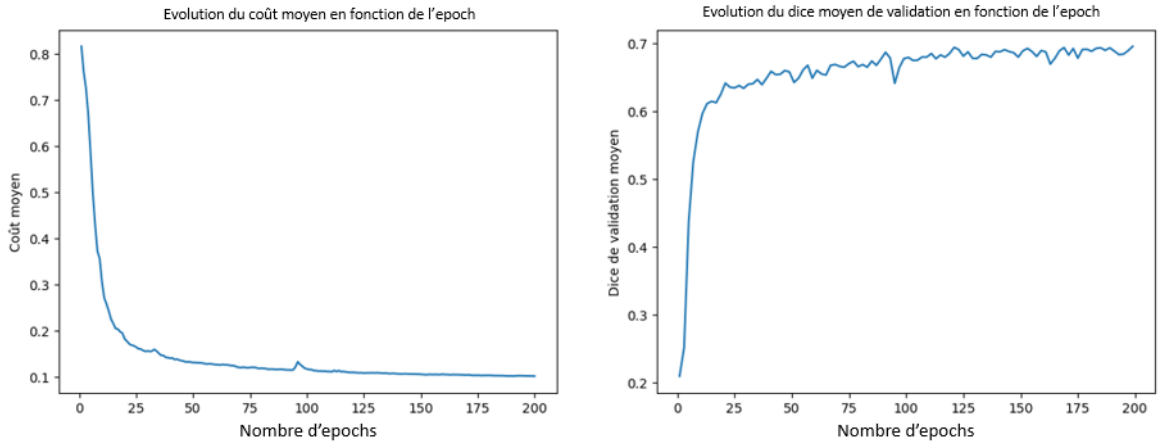


FIGURE 5.17 – *Courbes d'apprentissage obtenues en utilisant un modèle pré-entraîné. A gauche, l'évolution du coût moyen en fonction de l'epoch. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.*

Encore une fois nous avons présenté le dice moyen mais nous analysons également le dice pour chaque partie que nous segmentons à la figure 5.18. Cela permet de mieux comprendre le résultat visuel obtenu. Le dice final atteint pour les 5 grosses parties du coeur, c'est-à-dire les deux ventricules, les deux oreillettes et le myocarde du ventricule gauche, est environ de 0,6, ce qui est encore insuffisant. Les mêmes résultats avaient été obtenus avec la résolution de 64 × 64 × 64. Nous pouvons à nouveau observer un dice plus élevé pour l'artère pulmonaire et l'aorte. Ces parties obtiennent respectivement un dice de 0,73 et 0,82. Si ces parties obtiennent toujours de meilleurs résultats, c'est que leur identification est plus aisée peut être grâce à leur forme qui diffère des autres formes présentes sur les images.

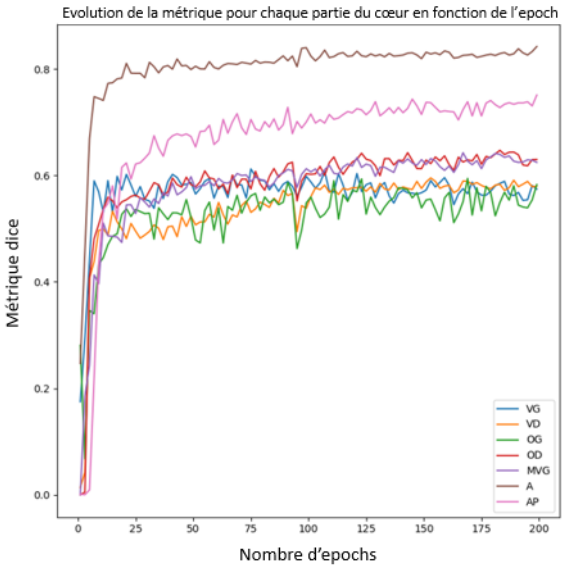


FIGURE 5.18 – Courbes des dices obtenues pour chaque partie du coeur avec un modèle pré-entraîné. Chaque courbe représente l'évolution du dice lors de l'apprentissage pour une partie du coeur. On observe que certaines parties comme l'aorte ou l'artère pulmonaire sont plus faciles à segmenter.

Ce résultat est vraiment meilleur en sachant que les volumes de l'aorte et de l'artère pulmonaire sont plus faibles que celui des cavités du coeur. En effet, au plus la partie à segmenter est petite, au plus une erreur de segmentation a de l'impact sur le dice.

5.3.4 Résultats du test J128 : Augmentation de la base de données

Les tests suivants réalisés dans ce chapitre sont basés sur l'augmentation de la quantité de données expliquée à la section précédente.

Malheureusement, tous les tests ont obtenu un dice inférieur à ce qui a été atteint sans l'augmentation de la base de données. En effet, avec la résolution 64 × 64 × 64, le dice maximum atteint est de 0,69 contrairement à 0,72 précédemment. La combinaison de paramètres utilisée se trouve dans le tableau 5.24.

TABLEAU 5.24 – Paramètres utilisés pour entraîner le modèle avec l'augmentation de la base de données et une résolution de 64 × 64 × 64

Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	3
Nombre d'epochs	200
Vitesse d'apprentissage	0.001
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

Pour ce qui est de la résolution 128 × 128 × 128, un dice maximum de 0,48 est atteint alors que sans augmentation de la quantité de données, un dice de 0,66 était obtenu. Les paramètres nécessaires pour arriver à ce résultat se trouvent dans le tableau 5.25.

Tableau 5.25 – Paramètres utilisés pour entraîner le modèle avec l'augmentation de la base de données et une résolution de 128 × 128 × 128

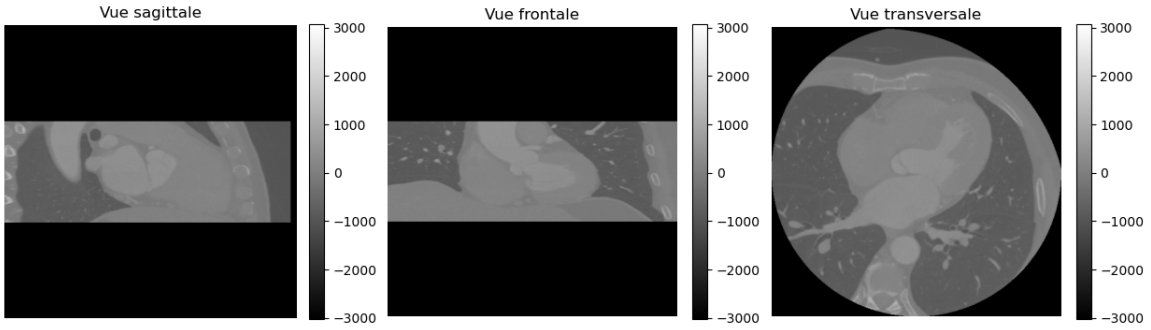
Paramètres	Valeurs
Taille des groupes	2
Nombre de couches	4
Nombre d'epochs	200
Vitesse d'apprentissage	0.0001
Nombre de neurones dans la plus petite couche	64
Dropout	0,3
Échauffement	Non

Nous avons également testé de pré-entraîner un modèle avec la résolution 64 × 64 × 64. Ensuite, nous utilisons les valeurs des poids pour initialiser les couches correspondantes du modèle avec la résolution supérieure. Cependant, les résultats obtenus restent toujours inférieurs à ceux atteints sans l'augmentation de la quantité de données. En effet, le dice maximum obtenu avec l'augmentation est de 0,51.

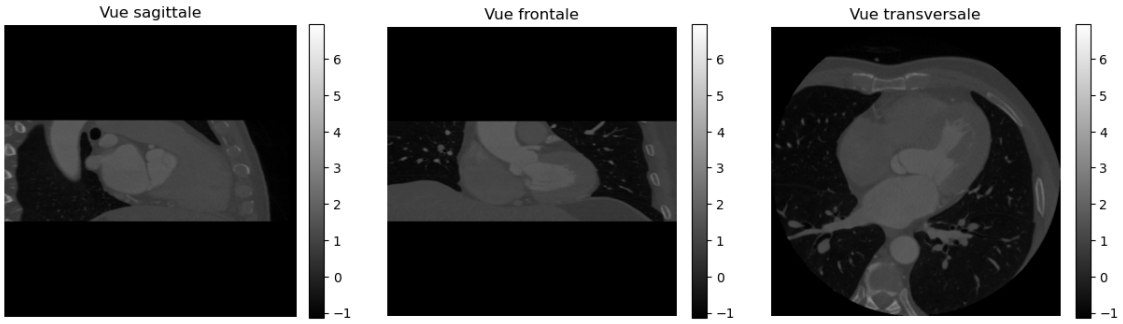
Nous pensons que les résultats sont inférieurs car la quantité de données créées artificiellement est trop importante par rapport à la quantité de données initiales. Nous avons dès lors testé de diminuer le nombre de données artificielles mais cela n'a pas eu d'impact réel sur les résultats. Ce faible résultat pourrait être amélioré avec plus de données initiales et une meilleure augmentation sur base de celles-ci.

5.3.5 Résultats du test K128 : Standardisation des données

Une fois la standardisation effectuée, nous observons une différence majeure comme nous pouvons le voir à la figure 5.19. En effet, la plage de valeurs des pixels est réduite, elle est maintenant comprise entre -1,15 et 6,95. La moyenne des valeurs est de 0 et la variance vaut 1.



(a) Données non standardisées



(b) Données standardisées

FIGURE 5.19 – Exemple de données d'entraînement standardisées ou non. Trois vues sont utilisées pour visualiser les données. Images tirées de la base de données du Multi-Modality whole heart segmentation challenge [35]

Ensuite, nous avons récolté de nouveaux résultats sur les données de validation afin de vérifier si la standardisation a un impact sur la performance générale du modèle.

Les résultats obtenus montrent que le dice moyen final obtenu avec la résolution $128 \times 128 \times 128$ est à présent de 0,69. Cette légère diminution peut être due à une meilleure généralisation du modèle. Le modèle précédent était peut être trop entraîné et donc trop précis sur certaines données. En conséquence, le modèle plus général a une perte de performance sur ces données.

La figure 5.20 montre la même allure que pour le même modèle sur des données non standardisées. A ce stade, nous ne pouvons pas dire si le modèle généralise mieux avec la standardisation.

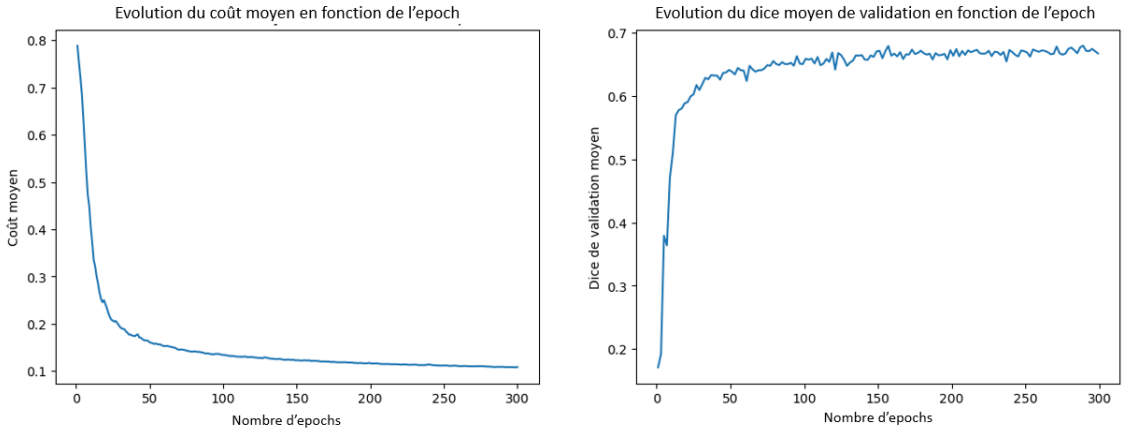


FIGURE 5.20 – Courbe d'apprentissage obtenues en utilisant les données standardisées. A droite, l'évolution du dice moyen sur les données de validation en fonction de l'epoch.

La figure 5.21 nous montre l'évolution du dice moyen pour les différentes parties du coeur. A nouveau, l'aorte présente le meilleur dice. L'artère pulmonaire obtient un dice légèrement inférieur avec la standardisation, il passe de 0,75 à 0,69. Les autres cavités du coeur semble atteindre un dice similaire sauf pour l'oreillette gauche avec un dice inférieur à 0,5. Encore une fois, ce graphe n'exprime aucune information quant à la généralisation du modèle.

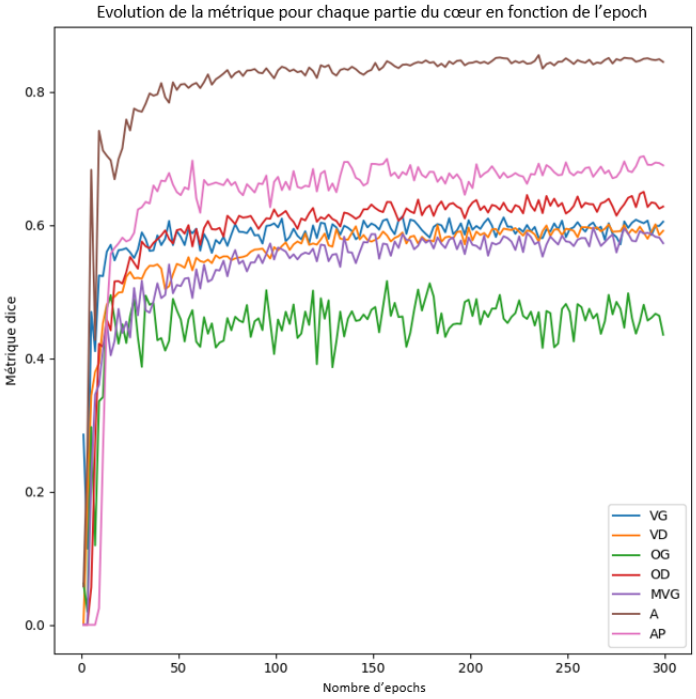


FIGURE 5.21 – Courbes des dices obtenues pour chaque partie du coeur avec des données standardisées. Chaque courbe représente l'évolution du dice lors de l'apprentissage pour une partie du coeur. On observe que certaines parties comme l'aorte ou l'artère pulmonaire sont plus faciles à segmenter.

Une autre manière pour vérifier si le modèle généralise mieux est de visualiser les images. Si la segmentation visuelle est meilleure sur une donnée mais que le dice moyen sur les quatre données est inférieure, alors le modèle généralise mieux. Cela est visible à la figure 5.23, le dice moyen est plus faible et pourtant le résultat visuel sur cette donnée est meilleur par rapport à la figure 5.16.

(a) Vue sagittale

(a) Vue transversale

FIGURE 5.23 – Visualisation du résultat obtenu en utilisant un modèle pré-entraîné et des données standardisées. A gauche l'image à segmenter, au milieu la segmentation attendue et à droite, la segmentation prédite par le modèle.

Avec ce modèle plus général, le dice obtenu pour l'aorte dépasse toujours celui de la plupart des articles du chapitre 4. Cependant, le résultat pour l'artère pulmonaire est dans ces cas inférieure à 0,70 et donc inférieur à celui des articles sauf pour un d'entre eux.

5.4 Résumé

Dans ce chapitre, nous avons exposé le travail réalisé pour obtenir un modèle capable de segmenter sept parties différentes du coeur : les ventricules gauche et droit, les oreillettes gauche et droite, le myocarde du ventricule gauche, l'aorte et l'artère pulmonaire. Nous avons travaillé avec différentes résolutions : 64 × 64 × 64 et 128 × 128 × 128. Avec la résolution 64 × 64 × 64, le meilleur dice moyen obtenu est de 0,72 ce qui dépasse l'objectif de 0,7. Pour la résolution 128 × 128 × 128, le dice maximum atteint est de 0,66 sans utilisation de modèle pré-entraîné, ce résultat est inférieur à l'objectif. Cependant un dice de 0,7 est atteint avec cette résolution si nous utilisons un modèle pré-entraîné en résolution 64 × 64 × 64. L'utilisation d'une augmentation artificielle de la quantité de données n'a pas abouti à de meilleurs résultats. Pour finir, nous avons standardisé nos données afin d'obtenir un modèle plus général. Dans ce cas, un dice de 0,69 est atteint ce qui est légèrement inférieur à l'objectif.

Bien que le dice objectif de 0,7 soit atteint dans certains cas, il s'agit du dice moyen. Lorsque nous le décomposons pour chaque partie à segmenter, nous remarquons que deux parties seulement atteignent ce dice. Ces parties sont l'aorte et l'artère pulmonaire. Les autres parties obtiennent un dice d'environ 0,6. Ces résultats sont malheureusement inférieurs à ceux présentés dans le chapitre 4 à l'exception de l'aorte et de l'artère pulmonaire. Différentes pistes d'amélioration sont envisageables, elles seront exposées dans le chapitre 7.

Chapitre 6

Évaluation du modèle sur de nouvelles données

Ce chapitre porte sur la dernière partie de ce mémoire, c'est-à-dire l'évaluation de notre modèle sur les données reçues aux *cliniques universitaires Saint Luc*. Il met en avant les différences entre les deux bases de données utilisées et comment nous avons tenté de combler celles-ci. Finalement, nous présentons les résultats obtenus tout le long du processus d'évaluation ainsi que la segmentation finale réalisée sur les données de Saint Luc.

6.1 Matériels

Pour cette partie, nous avons besoin du meilleur modèle créé avec les données standardisées du chapitre 5 pour segmenter le coeur. De plus, nous utilisons des données reçues aux *cliniques universitaires Saint Luc* sur lesquelles nous voulons effectuer la segmentation. Nous appellerons cette base de données SL. Le nombre de données s'élève à 48. Celles-ci ont été prises sur des patients ayant une embolie pulmonaire. Ces dernières ne comportent aucune annotation concernant la segmentation des cavités cardiaques. C'est pourquoi l'évaluation de la segmentation se fait uniquement sur base des résultats visuels. Les données ont également une dimension de 512 × 512 × C où la valeur de C dépend des patients. Un exemple de ces données est visible selon trois vues différentes à la figure 6.1.

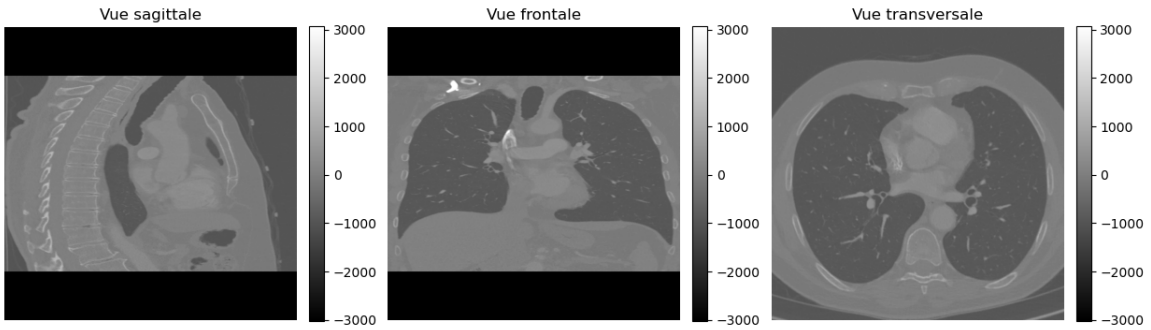


Figure 6.1 – Exemple de données à segmenter. Trois vues sont utilisées pour visualiser les données. Images tirées de la base SL.

La base de données d'entraînement est standardisée, ce qui n'est pas le cas pour la base SL. Cela engendre une grande différente dans les statistiques d'intensité des pixels entre les deux bases de données. Nous remarquons également que la première base de données est beaucoup plus centrée sur le coeur que la base SL.

6.2 Méthodes

6.2.1 Cadrage des données

La première étape réalisée pour cette dernière partie du travail est d'adapter le champ de vision des images. En effet, la base de données d'entraînement est plus centrée sur le coeur que celle à segmenter. Les figures 5.19 et 6.1 montrent bien une différence dans les organes pris en compte dans l'image. Avant de segmenter la nouvelle base de données, il est donc nécessaire de centrer la base SL sur le coeur également. Au plus les images représentent la même partie du corps que les données d'entraînement, au plus le modèle sera apte à faire une bonne segmentation. Cette étape a été réalisée sur base visuelle uniquement.

6.2.2 Standardisation des données

Puisque les données utilisées pour entrainer le modèle sont standardisées, nous devons également le faire pour la base de données SL. Pour rappel, le but de cette standardisation est d'avoir une moyenne de 0 et une variance de 1 dans la base de données. Nous voulons également que la plage de valeurs des pixels soient similaires à celle de l'autre base de données standardisées. La base SL a été standardisée avec la moyenne et l'écart type calculés sur les données d'entraînement.

6.2.3 Segmentation finale sur la base de données SL

La dernière étape est de tester le modèle entraîné sur les données standardisées sur la base de données SL. Ces données sont totalement inconnues pour le modèle. Ce test permet de vérifier que celui-ci est capable de généraliser ses performances sur de nouvelles données. Il faut rappeler que l'évaluation des résultats ne peut se faire que de manière visuelle car aucune annotation n'est disponible pour la base de données SL.

6.3 Résultats et discussion

6.3.1 Résultats du cadrage des données

Nous avons uniquement cadré les données de la base SL. La nouvelle dimension de ces images est de $350 \times 350 \times 140$. Avec celle-ci, nous obtenons des images assez similaires dans les deux bases de données. La figure 6.2 montre un résultat recueilli grâce à cette technique.

Comme nous pouvons le voir, les organes présents en dessous du coeur ont disparu lorsque nous observons les vues sagittale et frontale. En ce qui concerne la vue transversale, nous remarquons que différentes informations présentes autour du coeur sont ôtées des données.

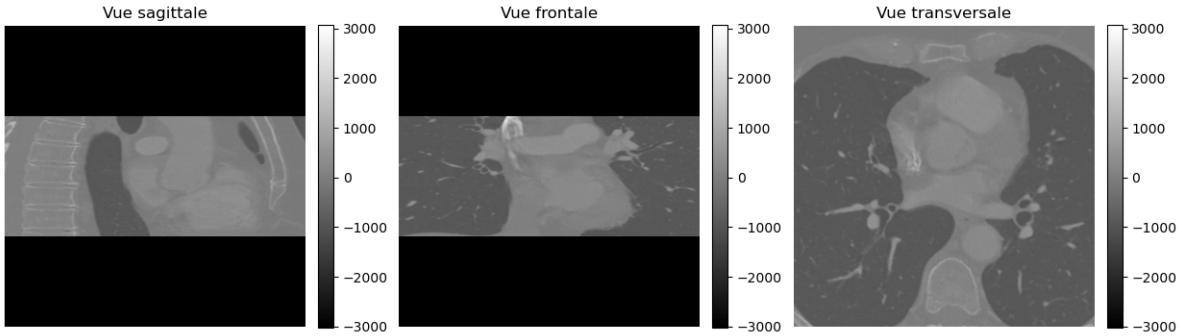


FIGURE 6.2 – Exemple de la base SL cadrée. Les images ont été centrées sur le coeur. Elles ont une dimension de 350 × 350 × 140. Trois vues sont utilisées pour visualiser ces données. Images tirées et cadrées de la base SL.

6.3.2 Résultats de la standardisation des données

La standardisation a été appliquée aux données de la base SL. Les statistiques obtenues montrent une moyenne proche de 0 et une variance s'approchant de 1. La figure 6.3 montre les images récoltées grâce à cette méthode.

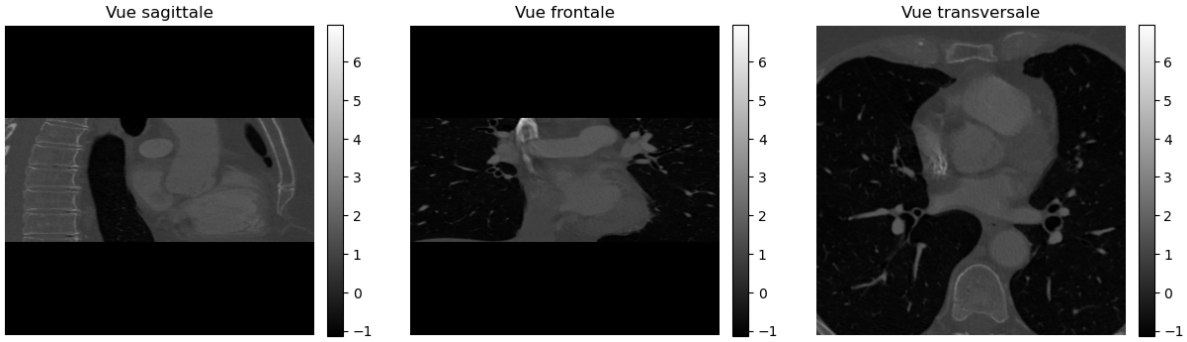


FIGURE 6.3 – Exemple de la base SL cadrée et standardisée. Trois vues sont utilisées pour visualiser ces données. Images tirées, cadrées et standardisées depuis la base de données SL.

Si nous comparons les figures 5.19 et 6.3, les valeurs des pixels correspondent pour les mêmes parties des images, par exemple les poumons ont une valeur inférieure à -1 dans les deux bases de données. La plage de valeur pour la base de données SL est comprise entre -1,14 et 6,83 , ce qui est relativement proche de la base de données d'entraînement standardisée.

6.3.3 Résultats de la segmentation finale sur la base de données SL

L'analyse des résultats sur la base SL ne peut se faire que de manière visuelle. En effet, aucune annotation n'est disponible, nous ne pouvons pas calculer de dice. La figure 6.4 montre le résultat obtenu lors de la segmentation sur les données de la base SL cadrées et standardisées. Celle-ci montre que la segmentation n'est pas parfaite. En effet, certaines parties du coeur ne sont pas correctement segmentées puisque certaines couleurs sont un peu mélangées.

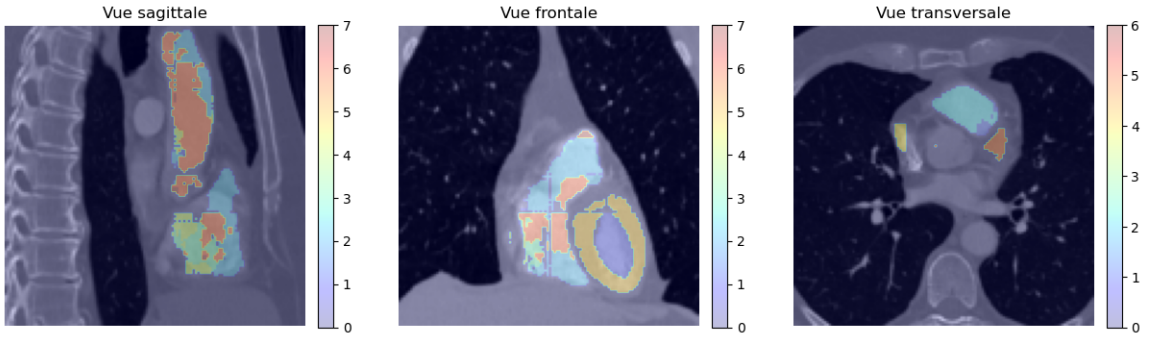


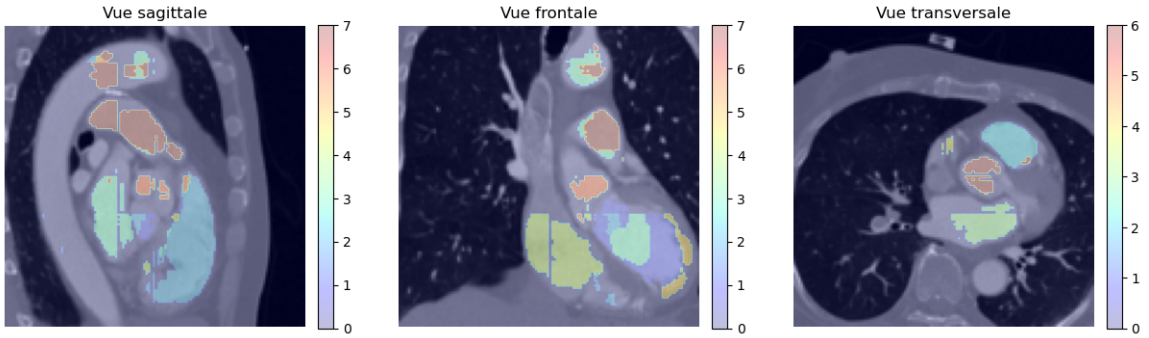
FIGURE 6.4 – Exemple 1 de segmentation sur les données de la base SL cadrées et standardisées. Trois vues sont utilisées pour visualiser ces données. La figure montre des segmentations présentes uniquement dans le coeur.

Puisque nous disposons de plusieurs données des *cliniques universitaires Saint Luc*, nous avons pu tester plusieurs fois notre modèle. La figure 6.5 présentent d'autres exemples de segmentation obtenues. Comme nous pouvons le voir sur ces images, il existe parfois des segmentations hors du coeur, cependant celles-ci restent minimes.

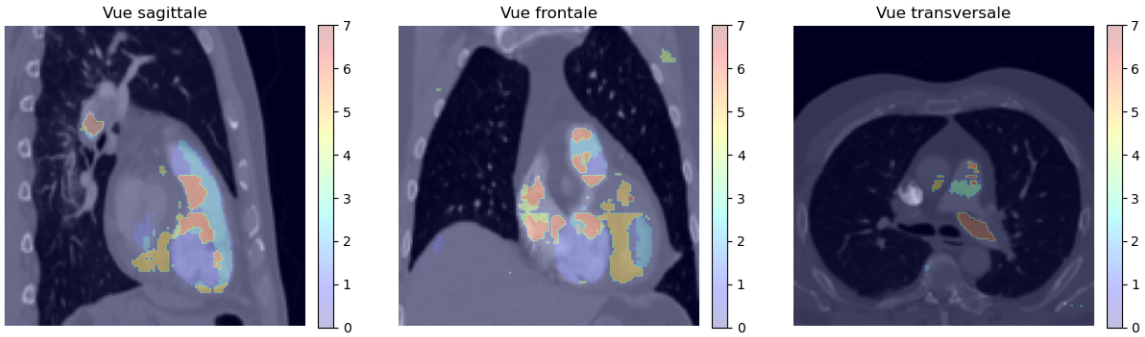
La segmentation montre des mélanges dans les couleurs et non pas des formes continues de la même couleur. La vue frontale de l'exemple 2 expose un mélange des couleurs bleu et rouge, ce mélange se retrouve également à l'exemple 1 et 4 ou encore sur les vues sagittales des exemples 1, 2 et 3. Ces couleurs correspondent à l'aorte et au ventricule droit. Pourtant, l'aorte est une des parties les mieux segmentées sur les données de validation.

Ces différents exemples montrent que le ventricule gauche et son myocarde semblent être relativement bien segmentés. Nous observons sur les différentes images le myocarde, dont la valeur vaut 5, qui entoure le ventricule dont la valeur est 1.

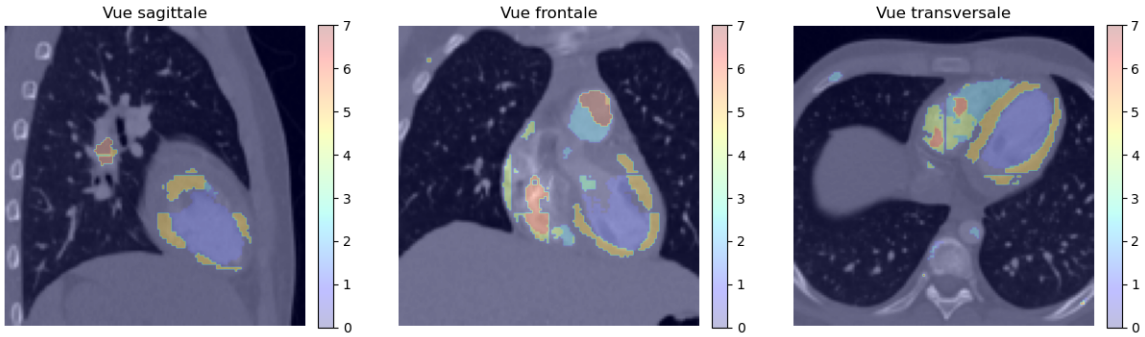
Pour finir, nous pensons que ces résultats sont encourageants mais présentent certains défauts, le plus important étant les couleurs mélangées. Différentes pistes d'amélioration sont envisageables, elles seront exposées dans le chapitre 7 de ce travail.



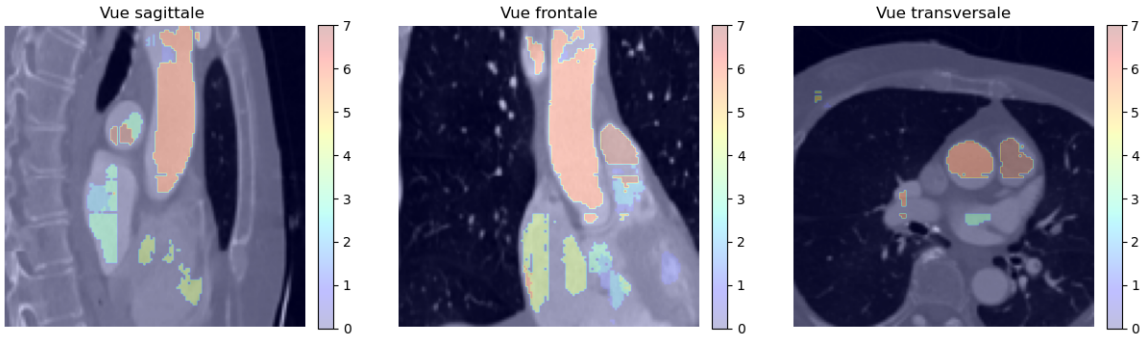
(a) Exemple 2



(b) Exemple 3



(c) Exemple 4



(d) Exemple 5

FIGURE 6.5 – *Trois exemples supplémentaires de segmentation sur les données de la base SL cadrées, standardisées et segmentées. Trois vues sont utilisées pour visualiser ces données.*

6.4 Résumé

Le but de ce chapitre est d'évaluer le modèle créé au chapitre précédent avec de nouvelles données reçues aux *cliniques universitaires Saint Luc*.

Les données d'entraînement et celles que l'on veut segmenter présentaient des différences sur plusieurs points. La première base de données est plus centrée sur le coeur que la deuxième. Nous avons dès lors cadré celle-ci sur le coeur. Les données d'entraînement sont standardisées. Nous avons donc également standardisé la deuxième base de données.

Finalement, nous avons réalisé la segmentation sur les données reçues aux *cliniques universitaires Saint Luc*. Les résultats obtenus semblent encourageants même si des défauts persistent. Nous avons remarqué que les parties les mieux segmentées sur les données de validation diffèrent des parties les mieux segmentées sur la deuxième base de données.

Chapitre 7

Pistes d'amélioration

Les différents résultats obtenus tout au long de ce travail sont prometteurs mais ne sont pas parfaits. C'est pourquoi ce chapitre propose différentes pistes d'amélioration envisageables pour produire de meilleurs résultats.

7.1 Amélioration de la segmentation du coeur

Pour rappel, le dice final atteint à la fin du chapitre 5 est de 0,69. Le but de cette étape est de créer un modèle permettant la segmentation des différentes parties du coeur. Malheureusement, une valeur de dice de 0,7 n'est pas atteinte pour toutes les parties du coeur que l'on veut segmenter. Pour améliorer ces résultats, nous pourrions tester différentes solutions. Nous pourrions essayer certaines techniques utilisées dans l'état de l'art puisqu'elles présentent de meilleurs résultats.

Première segmentation du coeur en entier

Par exemple, en faisant une première segmentation du coeur en entier pour ensuite différencier chaque partie du coeur segmenté. Cela permettrait de réduire la zone de segmentation. La deuxième partie du modèle segmenterait alors chaque partie du coeur dans la zone réduite.

Utilisation de combinaisons de fonctions de coût

Une autre amélioration serait de tenter différentes fonctions de coût hybrides comme cela a été fait dans le troisième article présenté [49]. En effet, nous n'avons testé que trois fonctions de coût séparément dans ce travail. Avec cette technique, notre fonction de coût final serait une combinaison d'autres fonction : $C = a \cdot C1 + b \cdot C2 + c \cdot C3$. Dans cette équation, les paramètres a, b et c sont des constantes pour donner de l'importance aux différentes fonctions de coût C1, C2 et C3. Ces fonctions seraient choisies pour essayer d'équilibrer certaines mauvaises segmentations. Par exemple une fonction de coût qui aide à différencier des classes ayant des formes similaires comme c'est le cas pour les ventricules ou les oreillettes.

Changement de résolution

Nous pourrions également essayer de travailler avec la résolution initiale de 512 × 512 × C où la valeur de C dépend du patient, puisque l'objectif est d'avoir une segmentation la plus précise possible. En effet, lorsque la dimension est réduite, il y a une certaine perte d'information et de précision. La segmentation attendue sur la figure 5.16 ne montre pas de délimitation nette entre les différentes parties du coeur. Si on regarde la forme bleue claire et la forme jaune, il existe une intersection entre les deux qui donne la couleur vert clair. Le résultat obtenu de cette segmentation montre que le modèle essaie de reproduire cette intersection qui en réalité n'existe pas sur les données non redimensionnées comme sur la figure 5.1. Le modèle utilisé avec de telles données aurait un nombre de couches plus élevé que celui utilisé avec la résolution de 128 × 128 × 128. En effet, comme cela a pu être observé à la section 5.3 pour passer de la résolution 64 × 64 × 64 à la résolution supérieure, le nombre de couche est passé de 3 à 4 car l'effet de la taille des images est plus important que la qualité. Nous avons observé une perte de dice en passant de la résolution 64 × 64 × 64 à 128 × 128 × 128. Cependant, nous espérons n'avoir qu'une faible diminution du dice, voir une augmentation en utilisant la résolution initiale de 512 × 512 × C car il n'y aurait pas de perte d'informations. Cette idée pourrait être combinée à un cadrage des images dés le départ. Le but serait de centrer les données au maximum sur le coeur, demander moins de ressources et supprimer l'impact des organes environnants.

Réduction du nombre de parties à segmenter

En fonction des besoins réels pour définir la gravité d'une embolie, toutes les classes n'ont peut-être pas besoin d'être segmentées. Dans ce cas, nous pourrions mettre la valeur des pixels de ces classes à la même valeur que l'arrière plan. Cela permettrait de simplifier la tache à accomplir. Nous pourrions dès lors diminuer le nombre de couches ou le nombre de neurones par couche dans notre modèle.

Utilisation d'un modèle par partie à segmenter

Finalement, une autre possibilité serait de segmenter les différentes parties du coeur séparément. Nous aurions donc un modèle par classe à segmenter. Cela diminuerait la difficulté de la tâche que le modèle doit effectuer. Une tâche plus simple permettrait peut être aussi de diminuer le nombre de neurones nécessaires par couche ou même le nombres de couches et au final avoir plus de modèles mais plus petits.

7.2 Amélioration de la segmentation de la base de données SL

En ce qui concerne la deuxième partie de ce travail, c'est-à-dire l'évaluation du modèle sur les données reçues des *cliniques universitaire Saint Luc*, les résultats visuels obtenus ne sont pas parfaits. Dans ce cas, ils pourraient également être améliorés. Pour obtenir ces résultats, nous avons dû cadrer les données et ensuite les standardiser.

Augmentation de la base de données

Nous pourrions tenter de généraliser encore plus le modèle en appliquant une augmentation de la base de données. Cette fois, ce ne sont pas des rotations ou des symétries qui seraient appliquées. L'augmentation serait ciblée sur la valeur des pixels en additionnant ou en multipliant les valeurs initiales avec des valeurs aléatoires. Cela généraliserait le modèle sur des statistiques d'intensité de pixels plus larges.

Meilleure évaluation des résultats

Une meilleure évaluation des résultats serait envisageable si nous avions des données supplémentaires. En effet, nous avons connaissance de documents contenant une série d'informations sur chaque patient dont nous avons reçu les images aux *cliniques universitaires Saint Luc*. Ceux-ci contiennent notamment les volumes des ventricules. Nous pourrions calculer le volume de chaque partie segmentée avec notre modèle. Ensuite, nous pourrions comparer les valeurs de volume pour quantifier les performances du modèle sur les données des *cliniques universitaires Saint Luc*.

7.3 Pistes d'amélioration générales

Ces différentes pistes d'amélioration peuvent être combinées afin d'obtenir les meilleurs résultats possibles. Si le premier modèle est amélioré, les résultats du deuxième modèle seront également meilleurs.

Autre technique pour calculer le dice

Il faut aussi noter que tous les modèles ont été entraînés sur les mêmes 16 données et validés sur les mêmes 4 données. Par manque de données, nous avons également éliminés l'ensemble de test, tous les dices finaux sont donc calculés sur les données de validation. Nous pourrions utiliser la méthode de validation croisée. Cette technique consiste à entraîner 20 fois le modèle avec 19 données et le valider sur une seule donnée en changeant à chaque fois la donnée de validation. Pour finir, un dice moyen est calculé sur base des 20 dices calculés. Cette méthode permettrait d'obtenir un dice moins bruité.

Changement de méthode d'apprentissage

Une solution plus radicale pour améliorer notre modèle serait de changer la méthode d'apprentissage du modèle et d'utiliser un apprentissage auto-supervisé. Dans ce cas, l'apprentissage serait réalisé directement grâce à la base de données SL et le modèle serait peaufiné à l'aide des données annotées du challenge MM-WHS.

7.4 Stratification des embolies

L'un des objectifs de ce mémoire était également de pouvoir stratifier la gravité d'une embolie pulmonaire. Pour réaliser cette tâche, nous aurions du avoir des données cliniques supplémentaires en plus des images que nous avons reçues. Grâce à celles-ci, nous aurions pu chercher des corrélations entre l'état du patient à court terme et différents biomarqueurs. Ces biomarqueurs pourraient être le volume ou le diamètre des cavités cardiaques. Malheureusement ces données n'ont pas été disponibles, cette partie présente alors une possible continuité à ce travail.

Chapitre 8

Conclusion

L'objectif de ce travail est de permettre la segmentation automatique des cavités cardiaques sur base d'images en trois dimensions prises par tomodensitométrie. Sur base de cette segmentation, certains biomarqueurs comme le volume ou le diamètre des cavités du coeur peuvent être calculés. Ceux-ci pourraient être utilisés afin de stratifier la gravité d'une embolie pulmonaire. Pour atteindre cet objectif, deux étapes ont été réalisées. La première est la création d'un modèle d'apprentissage profond pour effectuer la segmentation. La seconde étant l'évaluation du modèle sur les données que nous souhaitons segmenter.

Le modèle créé est plus particulièrement un U-Net en trois dimensions. Pour la création de celui-ci, nous avons du faire appel à des concepts bien connus de l'apprentissage automatique comme l'utilisation d'un modèle pré-entraîné ou l'augmentation artificielle de la base de données. Afin de valider le modèle, nous avons pour objectif d'obtenir un dice minimum de 0,7. Les premiers tests réalisés ont été faits en réduisant la dimension des données à $64 \times 64 \times 64$. Avec cette résolution, un dice de 0,72 est obtenu. Ensuite, des tests ont été effectués avec des données ayant une résolution de $128 \times 128 \times 128$. Le dice maximum est de 0,66.

Le meilleur dice avec cette résolution est obtenu en utilisant un modèle pré-entraîné. Nous avons utilisé les poids calculés dans le meilleur modèle avec la résolution de $64 \times 64 \times 64$ pour initialiser les couches du modèle avec la résolution supérieure. Avec cette technique un dice final de 0,7 a été obtenu. Malheureusement, il s'agit du dice moyen et cette valeur n'est pas atteinte pour toutes les parties du coeur que nous souhaitons segmenter. En effet, l'aorte et l'artère pulmonaire obtiennent la plupart du temps un dice supérieur à 0,7 mais les autres parties du coeur ne dépassent pas 0,65. Bien que les résultats obtenus n'atteignent pas réellement l'objectif fixé, les différentes étapes de la création ont permis de comprendre et d'analyser l'influence d'un grand nombre de paramètres.

Nous avons également tenté d'augmenter la quantité de données en réalisant des rotations ou des symétries des données initiales. Cependant, cette méthode n'a pas permis d'améliorer les résultats. Cela peut certainement s'expliquer par un trop grand nombre de données modifiées par rapport au nombre de données réelles. Une plus grande quantité de données dés le départ, par exemple 100 au lieu de 20 donnerait déjà de meilleurs résultats sans modifier le modèle en lui-même. Cela permettrait certainement d'obtenir un dice de 0,7 minimum pour toutes les parties du coeur et donc de dépasser le 0,65.

Les différents résultats obtenus peuvent être comparés à ce qui a déjà été réalisé. Nous avons présenté six travaux semblables à celui-ci au début de ce mémoire. Ils obtiennent généralement de meilleurs résultats sauf pour la segmentation de l'aorte et de l'artère pulmonaire. Il faut rappeler que ces travaux utilisaient des techniques supplémentaires comme une pré-segmentation du coeur en entier, la combinaison de plusieurs types d'images ou encore l'utilisation d'un modèle par classe à segmenter. Ces techniques sont des pistes d'amélioration possibles à ce mémoire. Nous en avons utilisé d'autres pour tenter d'améliorer nos résultats comme l'utilisation d'un modèle pré-entraîné.

La deuxième partie de ce travail nous a permis d'évaluer le modèle créé dans la première partie en segmentant les données reçues des *cliniques universitaires Saint Luc*. Pour réaliser cette tâche, nous avons modifié le cadrage et standardisé les statistiques d'intensité des pixels sur les images des deux bases de données utilisées. Nous avons gardé le paramétrage du modèle précédent et nous l'avons entraîné avec les données modifiées. Cela a eu pour conséquence une stagnation du dice obtenu à 0,69. Les segmentations obtenues sur les données de Saint Luc restent bien centrées sur le coeur et semblent segmenter correctement certaines parties de celui-ci. La diminution du dice de 0,7 à 0,69 montre une meilleure généralisation de notre modèle car la valeur de dice a diminué pour certains patients mais augmenté pour d'autres.

L'objectif final de ce mémoire était de pouvoir stratifier la gravité d'une embolie pulmonaire. Malheureusement, les données nécessaires pour effectuer cette dernière tâche n'ont pas été disponibles à temps. Une continuité possible à ce mémoire pourrait permettre d'analyser la suite des données reçues. Celles-ci contiennent des informations cliniques sur les différents patients, elles permettraient de trouver des corrélations entre la gravité d'une embolie et certains biomarqueurs. Ceux-ci pouvant être calculés sur base de la segmentation des cavités cardiaques.

Finalement, nous pouvons conclure que ce travail a permis d'avancer dans la technique de segmentation des cavités cardiaques. Des résultats encourageants ont été produits quant à l'utilisation d'un U-Net en trois dimensions pour réaliser cette tâche. Le modèle créé a pu être évalué sur de nouvelles données, cela implique aussi qu'il pourrait être également utilisé pour des données venant d'autres appareils d'acquisition. Pour terminer, ce mémoire ne constitue qu'un début de recherche puisqu'il possède des possibilités d'évolution.

Références

[1] Franck Verschuren et Frédéric Thys. *Communiqué de presse : L'embolie pulmonaire peut se soigner à domicile*. 2011. URL : https://www.saintluc.be/sites/default/files/2011-embolie-pulmonaire.pdf.

[2] *Assurance maladie : Personnes prises en charge pour embolie pulmonaire aiguë en 2020*. URL : https://www.ameli.fr/sites/default/files/2020_fiche_embolie-pulmonaire-aigue_0.pdf (visité le 11/04/2023).

[3] Guy Meyer et Olivier Sanchez. « Thrombose veineuse profonde et embolie pulmonaire ». In : *La revue du praticien* 59 (20 mars 2009), p. 393-404.

[4] Victor F. Tapson. « Acute pulmonary embolism ». In : *The New England Journal of Medicine* 358.10 (6 mars 2008), p. 1037-1052. DOI : 10.1056/NEJMra072753.

[5] *Physiopathologie de l'embolie pulmonaire*. URL : http://www.pneumocourlancy.fr/popup/EP_schema.html (visité le 11/04/2023).

[6] Tanya Dutta, William H. Frishman et Wilbert S. Aronow. « Echocardiography in the Evaluation of Pulmonary Embolism ». In : *Cardiology in Review* 25.6 (2017), p. 309-314. DOI : 10.1097/CRD.0000000000000158.

[7] F Jardin. « Le ventricule droit dans l'embolie pulmonaire ». In : *Réanimation* 10.2 (1er mars 2001), p. 225-231. DOI : 10.1016/S1164-6756(01)00098-6.

[8] *Qu'est-ce que l'atténuation des rayons X - Définition*. Radiation Dosimetry. 16 juill. 2020. URL : https://www.radiation-dosimetry.org/fr/quest-ce-que-lattenuation-des-rayons-x-definition/ (visité le 12/04/2023).

[9] S Kasperl et al. « Simulation-Based Planning of Optimal Conditions for Industrial Computed Tomography ». In : *International Symposium on Digital Industrial Radiology and Computed Tomography* (2011).

[10] *Radiography*. Radiology Key. 16 mai 2021. URL : https://radiologykey.com/radiography-3/ (visité le 16/05/2023).

[11] Jiang Hsieh. « Introduction ». In : *Computed Tomography : Principles, Design, Artifacts, and Recent Advances*. SPIE Press, 2003.

[12] F Welkenhuyzen et al. « Introduction ». In : *Industrial computer tomography for dimensional metrology : Overview of influence factors and improvement strategies*. 1er jan. 2009.

[13] Angela Cantatore et Pavel Müller. *Introduction to computed tomography*. Report. Kgs.Lyngby : DTU Mechanical Engineering, 2011.

[14] Greet Kerckhofs. *Cours Medical Imaging LGBIO2050 - UCLouvain*. 2021.

[15] *Philips - IQon Spectral CT La couleur s'invite dans vos images scanner*. Philips. URL : https://www.philips.be/fr/fr/healthcare/product/HCNOCTN284/iqon-spectral-ct-la-couleur-sinvite-dans-vos-images-scanner (visité le 14/04/2023).

[16] *Déroulement de l'examen échographique*. Medicine Key. URL : https://clemedicine.com/6-deroulement-de-lexamen-echographique/ (visité le 23/04/2023).

[17] Dinggang SHEN, Guorong WU et Heung-Il SUK. « Deep Learning in Medical Image Analysis ». In : *Annual Review of Biomedical Engineering* 19.1 (2017), p. 221-248. DOI : 10.1146/annurev-bioeng-071516-044442.

[18] Yann LECUN, Yoshua BENGIO et Geoffrey HINTON. « Deep learning ». In : *Nature* 521.7553 (mai 2015), p. 436-444. DOI : 10.1038/nature14539.

[19] Zhi-Hua ZHOU. « Introduction ». In : *Machine Learning*. Springer Nature, 20 août 2021.

[20] Matthieu CORD et Pádraig CUNNINGHAM. « Introduction ». In : *Machine Learning Techniques for Multimedia : Case Studies on Organization and Retrieval*. 1er jan. 2008. DOI : 10.1007/978-3-540-75171-7.

[21] Geoffrey HINTON et Terrence J. SEJNOWSKI. « Unsupervised learning ». In : *Unsupervised Learning : Foundations of Neural Computation*. 24 mai 1999. DOI : 10.7551/mitpress/7011.001.0001.

[22] Mohamed Farouk Abdel HADY et Friedhelm SCHWENKER. « Semi-supervised Learning ». In : *Handbook on Neural Information Processing*. 2013. DOI : 10.1007/978-3-642-36657-4_7.

[23] Qizhe XIE et al. « Self-Training With Noisy Student Improves ImageNet Classification ». In : 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, juin 2020, p. 10684-10695. DOI : 10.1109/CVPR42600.2020.01070.

[24] Shih-Cheng HUANG et al. « Self-supervised learning for medical image classification : a systematic review and implementation guidelines ». In : *npj Digital Medicine* 6.1 (26 avr. 2023), p. 1-16. DOI : 10.1038/s41746-023-00811-0.

[25] Chi-hau CHEN. « Introduction ». In : *Computer Vision In Medical Imaging*. World Scientific, 18 nov. 2013.

[26] Keiron O'SHEA et Ryan NASH. *An Introduction to Convolutional Neural Networks*. 2 déc. 2015. DOI : 10.48550/arXiv.1511.08458.

[27] Yann LECUN et Y. BENGIO. « Convolutional Networks for Images, Speech, and Time-Series ». In : *The Handbook of Brain Theory and Neural Networks* (1er jan. 1995).

[28] John H. MURPHY. « An Overview of Convolutional Neural Network Architectures for Deep Learning ». In : *Microway Inc* (2016), p. 1-22.

[29] *Filtres de convolution*. Mathinfo. URL : https://mathinfo.alwaysdata.net/2016/11/filtres-de-convolution/ (visité le 15/05/2023).

[30] *Réseaux convolutifs (CNN) : comment ça marche ?* Blent.ai. URL : https://blent.ai/blog/a/cnn-comment-ca-marche (visité le 14/04/2023).

[31] *Convolutional Neural Networks (CNN) Introduction*. Belajar Pembelajaran Mesin Indonesia. 7 mars 2018. URL : https://fr.scribd.com/document/475925758/Student-Notes-Convolutional-Neural-Networks-CNN-Introduction-Belajar-Pembelajaran-Mesin-Indonesia (visité le 14/04/2023).

[32] Brahim AIT SKOURT, Abdelhamid EL HASSANI et Aicha MAJDA. « Lung CT Image Segmentation Using Deep Neural Networks ». In : *Procedia Computer Science* 127 (1er jan. 2018), p. 109-113. DOI : 10.1016/j.procs.2018.01.104.

[33] Nikolaos Zioulis et al. « Hybrid Skip : A Biologically Inspired Skip Connection for the UNet Architecture ». In : *IEEE Access* 10 (2022), p. 53928-53939. doi : 10.1109/ACCESS.2022.3175864.

[34] Olaf Ronneberger, Philipp Fischer et Thomas Brox. *U-Net : Convolutional Networks for Biomedical Image Segmentation*. 18 mai 2015. doi : 10.48550/arXiv.1505.04597.

[35] Özgün Çiçek et al. *3D U-Net : Learning Dense Volumetric Segmentation from Sparse Annotation*. 21 juin 2016. doi : 10.48550/arXiv.1606.06650.

[36] *Perceptron in Machine Learning*. Javatpoint. url : https://www.javatpoint.com/perceptron-in-machine-learning (visité le 03/04/2023).

[37] John Lee et Michel Verleysen. *Cours Machine Learning LELEC2870 - UCLouvain*. 2022.

[38] Fabio Cappabianco et al. « A General and Balanced Region-Based Metric for Evaluating Medical Image Segmentation Algorithms ». In : 2019 IEEE International Conference on Image Processing (ICIP). 1er sept. 2019, p. 1525-1529. doi : 10.1109/ICIP.2019.8803083.

[39] T. Chai et R. R. Draxler. *Root mean square error (RMSE) or mean absolute error (MAE) ?* Numerical Methods, 28 fév. 2014. doi : 10.5194/gmdd-7-1525-2014.

[40] N. Sebe, M.S. Lew et D.P. Huijsmans. « Toward improved ranking metrics ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.10 (oct. 2000), p. 1132-1143. doi : 10.1109/34.879793.

[41] Rongjian Zhao et al. « Rethinking Dice Loss for Medical Image Segmentation ». In : 2020 IEEE International Conference on Data Mining (ICDM). Nov. 2020. doi : 10.1109/ICDM50108.2020.00094.

[42] Ken C. L. Wong et al. « 3D Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes ». In : *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, p. 612-619. doi : 10.1007/978-3-030-00931-1_70.

[43] Shruti Jadon. « A survey of loss functions for semantic segmentation ». In : 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). Oct. 2020. doi : 10.1109/CIBCB48159.2020.9277638.

[44] Diederik P. Kingma et Jimmy Ba. *Adam : A Method for Stochastic Optimization*. 29 jan. 2017. doi : 10.48550/arXiv.1412.6980.

[45] Xiahai Zhuang et Juan Shen. « Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI ». In : *Medical Image Analysis* 31 (juill. 2016), p. 77-87. doi : 10.1016/j.media.2016.02.006.

[46] Xiahai Zhuang. « Multivariate mixture model for myocardium segmentation combining multi-source images ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.12 (1er déc. 2019), p. 2933-2946. doi : 10.1109/TPAMI.2018.2869576.

[47] Majd Zreik et al. « Automatic segmentation of the left ventricle in cardiac CT angiography using convolutional neural networks ». In : 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI 2016). Avr. 2016. doi : 10.1109/ISBI.2016.7493206.

[48] Chunliang WANG et Örjan SMEDBY. « Automatic Whole Heart Segmentation Using Deep Learning and Shape Context ». In : *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. 2018, p. 242-249. DOI : 10.1007/978-3-319-75541-0_26.

[49] Xin YANG et al. « Hybrid Loss Guided Convolutional Networks for Whole Heart Parsing ». In : *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. 2018, p. 215-223. DOI : 10.1007/978-3-319-75541-0_23.

[50] Akifumi YOSHIDA et al. « Automated heart segmentation using U-Net in pediatric cardiac CT ». In : *Measurement : Sensors* 18 (1er déc. 2021), p. 100127. DOI : 10.1016/j.measen.2021.100127.

[51] Qianqian TONG et al. « 3D Deeply-Supervised U-Net Based Whole Heart Segmentation ». In : *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. Springer International Publishing, 2018, p. 224-232. DOI : 10.1007/978-3-319-75541-0_24.

[52] Marija HABIJAN et al. « Whole Heart Segmentation from CT images Using 3D U-Net architecture ». In : 2019 International Conference on Systems, Signals and Image Processing (IWSSIP). Juin 2019, p. 121-126. DOI : 10.1109/IWSSIP.2019.8787253.

[53] Stewart GAEDE et al. « An evaluation of an automated 4D-CT contour propagation tool to define an internal gross tumour volume for lung cancer radiotherapy ». In : *Radiotherapy and Oncology* 101.2 (1er nov. 2011), p. 322-328. ISSN : 0167-8140. DOI : 10.1016/j.radonc.2011.08.036.

[54] *MONAI - Home*. URL : https://monai.io/ (visité le 02/04/2023).

[55] Huiyan JIANG et al. « A review of deep learning-based multiple-lesion recognition from medical images : classification, detection and segmentation ». In : *Computers in Biology and Medicine* 157 (mai 2023), p. 106726. DOI : 10.1016/j.compbiomed.2023.106726.

[56] Ibrahem KANDEL et Mauro CASTELLI. « The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset ». In : *ICT Express* 6.4 (1er déc. 2020), p. 312-315. DOI : 10.1016/j.icte.2020.04.010. (Visité le 18/05/2023).

[57] Samuel DODGE et Lina KARAM. « Understanding how image quality affects deep neural networks ». In : 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX). Juin 2016, p. 1-6. DOI : 10.1109/QoMEX.2016.7498955.

[58] Sungheon PARK et Nojun KWAK. « Analysis on the Dropout Effect in Convolutional Neural Networks ». In : *Computer Vision – ACCV 2016*. Springer International Publishing, 2017, p. 189-204. DOI : 10.1007/978-3-319-54184-6_12.

[59] Akhilesh GOTMARE et al. *A Closer Look at Deep Learning Heuristics : Learning rate restarts, Warmup and Distillation*. 29 oct. 2018. DOI : 10.48550/arXiv.1810.13243.

[60] Rich CARUANA. « Multitask Learning ». In : *Machine Learning* 28 (1er juill. 1997). ISSN : 978-1-4613-7527-2. DOI : 10.1023/A:1007379606734.

[61] Carole H. Sudre et al. « Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations ». In : *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer International Publishing, 2017, p. 240-248. doi : 10.1007/978-3-319-67558-9_28.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**