



## ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

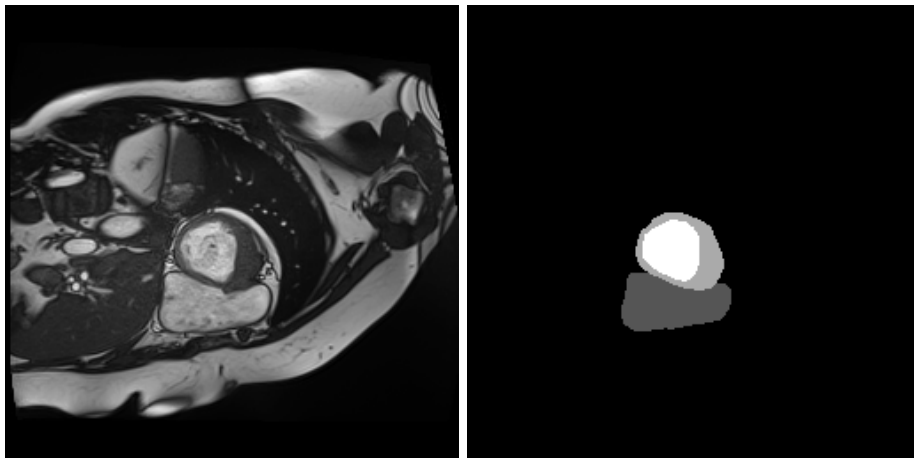
MTI865 - APPRENTISSAGE PROFOND POUR LA VISION PAR ORDINATEUR

---

### Challenge

Segmentation sémantique

---



LÉA GRIMA (GRIL26369901)  
RAPHAËL LARGEAU (LARR92290101)  
TRISTAN LOUKIANENKO (LOUT72330101)  
SIMON-OLIVIER DUGUAY (DUGS22129805)

23 DECEMBRE 2023

## Abstract

Aujourd'hui, la segmentation sémantique a de plus en plus d'applications, notamment dans le secteur médical. En effet, l'analyse d'imagerie médicale demande une grande expertise et beaucoup de temps. C'est pourquoi de nombreux articles dans la littérature proposent de créer des modèles permettant de réaliser ces tâches efficacement et avec une excellente précision, dans l'objectif de faire gagner du temps aux spécialistes. Dans la suite de ces travaux, nous proposons un modèle permettant de segmenter différentes parties du cœur. Notre approche essaie de trouver un bon compromis entre l'utilisation d'augmentation des données, d'apprentissage semi-supervisé et d'autres méthodes d'apprentissage profond et de vision par ordinateur. Nous utilisons à notre avantage une architecture de UNet++, très performante pour les tâches de segmentation, avec des poids pré-entraînés pour l'encodeur. Notre approche offre un résultat environ 10% plus précis qu'une architecture U-Net de base, selon la métrique de Dice. Bien que notre modèle éprouve éprouve une plus grande difficulté pour prédire une classe en particulier, nous obtenons un bon performance globale.

## 1 Introduction

Dans le contexte du domaine médical, la segmentation d'images a son importance. Elle aide les médecins dans le diagnostic et le suivi de certaines pathologies. Nous pouvons penser aux tumeurs pour lesquelles il est possible de calculer le volume et suivre l'évolution (grossissement ou rétrécissement). Cependant, le traitement des données d'imagerie médicale amène plusieurs problématiques. Premièrement, la quantité de données et leur diversité rendent la tâche de segmentation plus compliquée que lorsque la base de données est plus homogène. De plus, l'étiquetage des données est très chronophage pour les experts du domaine et donc la plupart du temps les données ne sont pas étiquetées. Il est donc nécessaire de mettre en place un algorithme d'apprentissage qui prenne en compte cette absence d'étiquettes et qui puisse tout de

même donner de bonnes performances.

Dans le cadre du challenge, les données à notre disposition sont des images médicales et l'objectif est de segmenter le cœur. Parmi elles, une petite partie est fournie avec la segmentation de l'organe cardiaque et une autre part, plus grande, n'a pas d'étiquettes.

Dans le présent rapport, nous proposons donc une architecture de réseau de neurones permettant la mise en place d'un apprentissage semi-supervisé sur les données qui nous sont fournies. Un état de l'art est donné avant la présentation de notre méthode. Ensuite, une évaluation du modèle et de ses performances est faite grâce à différentes métriques. Avant de conclure, une courte discussion est proposée pour mettre en lumière certains de nos résultats.

## 2 État de l'art

### 2.1 Modèles

Pour la sélection de notre modèle, nous avons examiné d'autres modèles performant en segmentation en général ou bien en segmentation médicale précisément. Notre idée était de partir d'un modèle plus récent ayant déjà été utilisé dans des projets de recherche et ayant prouvé de bonnes performances, puis les adapter à la tâche donnée. Il était aussi pertinent de trouver un modèle ayant des poids pré-entraînés disponibles en ligne.

À partir du U-Net [6] de base, nous avons élargi les recherches vers d'autres versions de cette architecture populaire, soit le UNet++ [9] ainsi que le U<sup>2</sup>-Net [5]. Bien que le U<sup>2</sup>-Net semble plus récent et offre de meilleures performances, la taille de son architecture le rend difficile à utiliser et à entraîner avec les ressources nous étant disponibles.

Nous avons aussi exploré des architectures de modèle utilisées et conceptualisées pour de la segmentation médicale, comme le MANet [3].

Finalement, nous avons visité des architectures autres que Unet qui ne s'appliquent pas forcément au domaine médical, mais pour la segmentation en général, soit le FPN [4].

Bien que plusieurs options auraient bien performées pour cette tâche, nous nous sommes

arrêtés sur le UNet++. Autant pour sa performance que pour sa taille, nous permettant de mieux nous adapter et de pouvoir tester d'autres variables.

## 2.2 Fonctions de coût

Par ailleurs, nous avons cherché les différentes fonctions de coût appropriées à notre problème. Nous avons donc pris l'entropie-croisée qui fonctionne généralement bien sur les problèmes d'apprentissage profond et notamment sur les problèmes de segmentation sémantique.

Nous avons aussi sélectionné l'entropie croisée pondérée puisque la surface des 3 classes que nous cherchons à détecter et du fond sont inégalement réparties dans les images. En effet, le fond représente la majorité de la surface des images et les classes ne recouvrent pas non plus la même surface entre elles. Dans ce contexte, l'entropie croisée pondérée pourrait permettre réduire les inégalités entre les classes.

De plus, puisque nous cherchons à réaliser de la segmentation sémantique, nous avons donc choisi d'étudier la fonction de perte Dice [1], qui est particulièrement adaptée à ce type de problème, puisqu'elle évalue si la superposition entre vérité terrain et prédiction est bonne ou mauvaise.

Enfin, vérifier l'impact d'une contrainte de circularité minimisant la distance à un centroïde nous est apparu comme approprié sur les classes 2 et 3. En effet, puisqu'elles sont généralement réparties en forme de cercle (classe 2) et de disque (classe 3).

## 3 Méthode

### 3.1 Modèle sélectionné

Nous avons choisi l'architecture UNet++, offrant un équilibre entre la taille du modèle et sa bonne performance. Les blocs encodeurs de notre modèle sont des blocs ResNet [2], plus précisément des blocs resnet34, pré-entraînés avec sur imagenet. Les blocs ResNet offrent une bonne flexibilité niveau taille et poids de pré-entraînement, nous permettant d'essayer plusieurs combinaisons possibles. Ce sont aussi

des encodeurs souvent utilisés pour des tâches de segmentation.

Après plusieurs tests et considérations, nous avons opté pour 0.5 de dropout, ReLu comme fonction d'activation et Adam comme optimisateur. Cela nous aide à limiter le sur-apprentissage tout autant que la disparition du gradient.

### 3.2 Pré-traitement

Pour pré-traiter les données dans le cadre de l'apprentissage profond, il est commun d'appliquer une normalisation sur les données. La normalisation consiste à retirer la moyenne et diviser par l'écart-type chacune des données. Cette normalisation peut se faire de plusieurs manières différentes : globalement ou plus localement. Notons  $\mu$  la moyenne sur l'ensemble des données et  $\sigma$  l'écart-type associé. Si l'on considère une image  $I$  et que l'on note  $p_i$  ses pixels, on obtient la formule suivante :

$$\forall i \in I, p'_i = \frac{p_i - \mu_I}{\sigma_I}$$

Cette méthode est utile pour mettre les données à la même échelle et réduire l'importance des valeurs aberrantes. Par conséquent, les algorithmes d'apprentissage sont aidés, convergents plus rapidement et leurs performances sont donc améliorées en théorie.

Pour l'application locale, elle se fait avant le début de l'apprentissage. Les données d'apprentissage sont divisées en lots (*batch*), c'est-à-dire en sous-ensembles, contenant 16 images chacun. Il est alors possible d'effectuer une normalisation par lots en calculant les moyennes et écarts-types de chaque sous-ensemble puis en appliquant la formule ci-dessus.

Ce pré-traitement n'a cependant pas été gardé dans la version finale de notre algorithme, car les performances n'étaient pas si différentes sans normalisation et cela ajoutait du temps d'exécution lors de l'entraînement du modèle.

### 3.3 Augmentation des données

De par l'essence même de notre problème non-supervisé et pour pallier le manque de données annotées tout en améliorant la généralisation de notre modèle. Nous avons eu recours à

une stratégie d'augmentation de données. Nous avons donc réalisé une série de transformations aléatoires sur chaque image du jeu d'entraînement et de validation de la base annotée. Les transformations étaient les suivantes :

- Rotation de l'image entre  $-20^\circ$  et  $+20^\circ$ ;
- Translation de l'image de 20 pixels sur les axes X et Y (positif ou négatif) ;
- Application d'un bruit "Salt and Pepper" sur 1% de l'image.

Cette technique permet de simuler la potentielle variabilité entre les différents scans des différents patients. Nous avons décidé aussi d'ajouter une transformation incluant du bruit pour simuler ce qu'il peut arriver dans la réalité clinique, c'est à dire des scans plus défectueux que d'autres, contenant une anomalie ou du bruit résultant d'un phénomène aléatoire. La base de données annotée a donc doublé de volume contenant les images annotées initiales et leur "double" contenant aléatoirement une ou plusieurs de ces transformations. Évidemment, ces transformations étaient aussi réalisées sur les masques de segmentation, excepté le bruit, par souci de clarté, qui aurait pu se révéler comme une complexification inutile de notre problème.

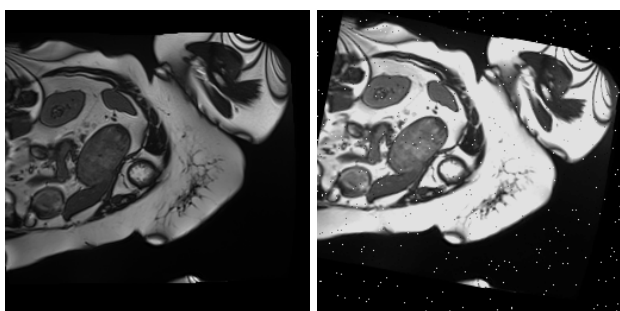


FIGURE 1 – Exemple d'une image vierge et de sa version augmentée

Cependant, même si théoriquement cette technique aurait dû permettre d'améliorer significativement la généralisation et la robustesse de notre modèle, les améliorations dans les résultats n'étaient pas assez notables pour passer plus de temps sur cette étape pré-entraînement.

### 3.4 Semi-supervisé

Pour améliorer nos chances d'avoir un modèle performant, il était important de tirer avantage de toutes les images qui nous étaient données. Puisque nous avons un grand nombre de données non étiquetées et un petit nombre de données étiquetées, nous avons opté pour la création de pseudo-labels, soit de l'apprentissage semi-supervisé.

Après avoir entraîné un modèle de départ, uniquement avec les données étiquetées, jusqu'à obtenir une performance raisonnablement bonne. Nous avons utilisé ce modèle pour créer des pseudo-labels à partir des données non étiquetées. Après avoir adapté notre Dataloader, nous étions prêts à utiliser les données et leurs nouvelles étiquettes.

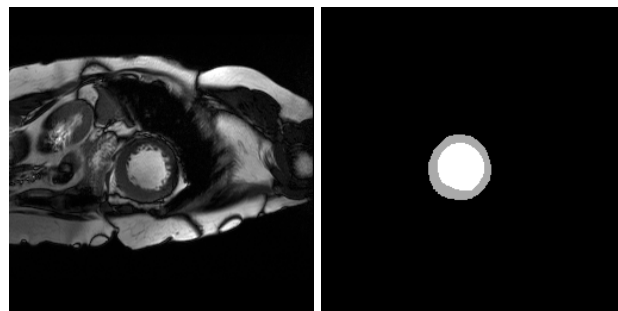


FIGURE 2 – Image non-étiquetée et sa pseudo-étiquette générée

Bien que cette technique de génération de pseudo-étiquettes soit efficace et a prouvé son bon fonctionnement dans notre cas, il existe méthodes plus sophistiquées et plus performantes permettant d'obtenir de meilleures pseudo-étiquettes. Faute de temps et de ressources, nous sommes restés avec cette méthode plutôt basique.

### 3.5 Fonctions de coût

Dans l'objectif d'optimiser l'entraînement de notre modèle, nous avons comparé les fonctions de coût et contraintes suivantes :

- entropie-croisée,
- entropie-croisée pondérée,
- dice,
- contrainte de circularité.

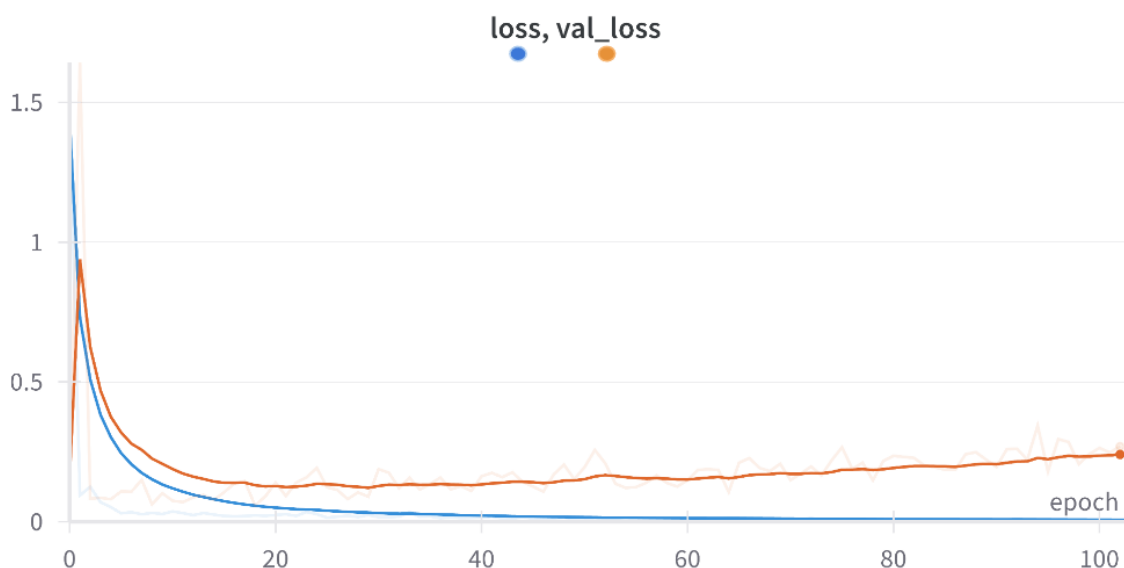


FIGURE 3 – Perte d'entraînement et de validation

Pour cela, nous avons procédé avec ces fonctions seules ou en associant l'entropie-croisée (pondérée ou pas) avec la Dice ou la contrainte de circularité. Nous avons remarqué que l'entropie-croisée, pondérée avec les poids :  $[0,05; 0,9; 0,8; 0,7]$  correspondant respectivement aux classes : [fond, 1, 2 et 3], est bien plus performante que l'entropie-croisée classique. Par ailleurs, il n'y a pas de différence majeure avec ou sans la fonction Dice. Enfin, la contrainte de circularité rendait de moins bons résultats. Il est possible que cela soit dû à une mauvaise implémentation de notre part de cette contrainte.

De cette manière, nous avons choisi d'utiliser comme fonction de coût uniquement l'entropie-croisée pondérée avec les poids :  $[0,05; 0,9; 0,8; 0,7]$ . Nous ne nous sommes pas servis de la Dice par souci de simplicité, puisqu'elle n'apportait pas de bénéfice visible. Enfin, nous n'avons pas utilisé la contrainte de circularité qui est contre-productive.

### 3.6 Entraînement

Les encodeurs étant déjà pré-entraînés, l'apprentissage de notre modèle était plutôt rapide. L'entraînement a principalement été fait sur les serveurs de Saturn Cloud, ainsi que sur une

carte graphique Nvidia GTX 1070.

Nous voyons donc, dans la figure 3 un apprentissage très rapide, où la fonction de coût la plus basse est atteinte vers la 20-30-ème epoch. Cela est dû au fait que nous utilisons des encodeurs avec des poids pré-entraînés.

Les encodeurs ont donc déjà des liens entre ses neurones pertinents pour l'extraction des caractéristiques d'images. Même si ce ne sont pas des encodeurs entraînés avec des images médicales, les poids sont bien mieux initialisés qu'à l'aléatoire et donc s'ajustent bien plus rapidement.

Il faut donc simplement adapter ces encodeurs et entraîner les décodeurs à la sous-tâche sur laquelle on les entraîne.

Dans notre cas, nous sauvegardons uniquement les poids ayant eu la valeur de coût de validation la plus basse. Donc malgré le sur-entraînement qui semble se produire à partir de la 40e epoch, c'est uniquement le meilleur modèle qui est sauvegardé et qui sera utilisé ensuite.

### 3.7 Post-traitement

Pour traiter les images après la segmentation, nous avons décidé d'utiliser des techniques de morphologie mathématique, en particulier les

opérations d'ouverture et de fermeture pour essayer d'améliorer la cohérence des segmentations.

En effet, l'ouverture, étant le résultat d'une érosion suivie d'une dilatation, permet d'éliminer les petits détails indésirables (considérés comme du bruit sur le masque) dans les régions segmentées. La fermeture, quant à elle, est le résultat d'une dilatation suivie d'une érosion permettant de connecter les discontinuités des masques. Nous avons utilisé ces deux opérations particulièrement, car elles sont régulièrement utilisées en segmentation pour améliorer la forme des objets segmentés et atténuer les artefacts résultant de la segmentation du modèle.

Cependant, malgré la pertinence de ces techniques dans d'autres contextes [8], nous avons constaté que leur application à nos images segmentées n'a pas nécessairement conduit à une amélioration des résultats. Ceci peut être dû à la complexité des structures cardiaques ou à la trop grande variabilité de la forme des masques dépendant du patient. Nous avons donc fait le choix de nous concentrer sur d'autres aspects plus importants de l'approche pour le temps qu'il nous était donné de respecter.

## 4 Résultats

### 4.1 Métriques d'évaluation

Pour évaluer notre modèle, nous nous sommes servis de 3 métriques différentes et complémentaires qui sont :

- la Dice,
- la distance d'Hausdorff (HD),
- la distance moyenne entre les surfaces (ASD).

La métrique de Dice, qui évalue la superposition entre la surface prédite  $\hat{Y}$  et la vraie surface  $Y$  des classes une à une (elle n'est pas calculée sur le fond). Cet indicateur est défini entre 0 et 1, la meilleure valeur étant 1 et représentant une superposition parfaite des deux surfaces. À l'inverse une valeur de 0 correspond à deux sur-

faces ne se superposant pas.

$$Dice(Y, \hat{Y}) = \frac{|Y \cap \hat{Y}|}{|Y| \cup |\hat{Y}|}$$

La distance d'Hausdorff dépend des distances maximales entre les surfaces  $Y$  et  $\hat{Y}$  pour chaque classe. Cette distance doit être minimisée, la valeur optimale étant 0. Elle est exprimée par formule suivante :

$$H(Y, \hat{Y}) = \max(h(Y, \hat{Y}), h(\hat{Y}, Y))$$

$$\text{Avec, } h(Y, \hat{Y}) = \max_{y \in Y} \min_{\hat{y} \in \hat{Y}} \|y - \hat{y}\|$$

Enfin, la distance moyenne entre les surfaces  $Y$  et  $\hat{Y}$  de chaque classe, que nous cherchons à minimiser jusqu'à 0 idéalement. Elle est définie selon :

$$ASD(Y, \hat{Y}) = \frac{1}{|Y| + |\hat{Y}|} \left( \sum_{i=1}^{|Y|} d(p, Y) + \sum_{i=1}^{|\hat{Y}|} d(p, \hat{Y}) \right)$$

$$\text{Avec, } d(p, Y) = \min_{y \in Y} \|p - y\|$$

### 4.2 Tableaux des résultats

Pour pouvoir obtenir nos résultats, nous avons utilisé le jeu de données de validation à notre disposition. Bien que cela ne soit pas parfait, cette méthode permet de pouvoir comparer les performances de notre modèle avec d'autres sur exactement les mêmes données. De cette manière, nous réduisons le risque de faire du sur-apprentissage puisque les modèles n'apprennent pas directement à partir du jeu de validation.

Nous avons donc entraîné un Unet de base sur les données jusqu'à obtenir les meilleures performances possibles (voir 1). Cela nous permet de montrer les performances atteignables avec un modèle simple et de mesurer les améliorations apportées par notre modèle.

Métrique	Dice	HD	ASD
C1	0.49	31.37	10.06
C2	0.61	39.96	6.92
C3	0.79	19.63	6.21
Moyenne	0.63	30.32	7.73

TABLE 1 – Résultats avec un Unet de base

Ensuite, nous avons entraîné notre modèle avec exactement les mêmes données d'apprentissage et de validation. Comme avec le modèle précédent, nous avons arrêté l'apprentissage au moment où nous avons les meilleures performances (voir 2).

Métrique	Dice	HD	ASD
C1	0.57	23.67	6.13
C2	0.76	23.46	3.60
C3	0.87	12.29	4.38
Moyenne	<b>0.736</b>	<b>20.14</b>	<b>4.38</b>

TABLE 2 – Résultats avec notre modèle

## 5 Discussions

### 5.1 Analyse des résultats

Nous pouvons tout de suite observer qu'une amélioration remarquable a lieu quand on compare nos résultats avec ceux du modèle U-Net de base, notamment avec une amélioration de près de 10% sur la métrique de Dice.

En terme de performance, nous remarquons qu'une classe se démarque et est moins bien détectée que les autres. C'est la classe 1 qui, même lorsqu'une partie du cœur est visible sur les images d'entrée, peut ne pas être présente. De plus, c'est la classe dont la forme de sa surface varie le plus dans les données. De ce fait, le modèle a plus de mal à trouver cette classe lorsqu'elle représente très peu de pixels et donc à la segmenter correctement. Nous pouvons voir qu'autant pour un UNet de base ou pour notre modèle utilisant l'architecture UNet++, les métriques sont moins bonnes pour la classe 1 dans les tableaux de la section 4.2 ci-dessus. On peut par exemple remarquer que la Dice ne dépasse pas 60% pour la classe 1 alors qu'elle dépasse 75% pour les 2 autres ce qui fait diminuer la Dice moyenne et donc la performance mesurée.

### 5.2 Faux positifs

Nous pouvons également parler des faux positifs (FP) obtenus lors de l'exécution de notre modèle. Plusieurs fois, l'algorithme

nous a renvoyé une segmentation cardiaque alors que le cœur n'était même pas présent sur l'image d'entrée (voir figure 4).



FIGURE 4 – Exemple de faux positif : une image, la vérité terrain et la prédiction

Il arrive aussi souvent que le modèle prédise la classe 1 alors qu'elle n'est pas visible sur l'image (voir figure 5).

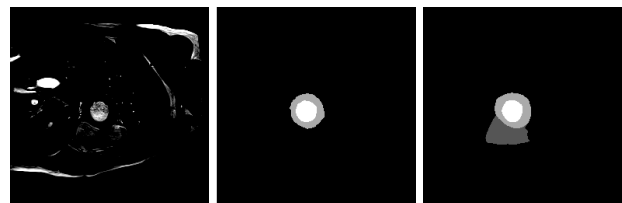


FIGURE 5 – Autre exemple de faux positif : une image, la vérité terrain et la prédiction

Étant entraîné pour détecter 3 classes en plus du fond, le modèle a parfois segmenté des formes qui ne sont en réalité pas liées à l'organe ciblé. Pour de tels cas, les métriques utilisées pour la performance tendent vers 0 pour la Dice (aucune superposition des ensembles réels et reconstruits) ou vers de grandes valeurs pour HD et ASD. Malgré le fait que les performances soient affectées par les FPs, il est pertinent de noter que dans le domaine médical ils sont préférables aux faux négatifs. En effet, un médecin va préférer examiner un patient qui a été estimé malade sans l'être plutôt que de laisser un patient ayant une tumeur rentrer chez lui sans consultation suite à une erreur du modèle. De plus, un médecin vérifiera toujours les segmentations produites, elles ne remplacent son expertise, mais lui permettent plutôt de gagner du temps en vérifiant et corrigeant un travail presque terminé.

## 6 Conclusion

### 6.1 Résumé

En conclusion, le modèle que nous avons produit est assez prometteur pour la segmentation des différentes parties du cœur. Cependant, nous avons fait face à plusieurs défis qui nous ont empêchés de l'améliorer. Ce challenge nous a permis de créer un modèle jetant les bases pour un modèle plus poussé en segmentation d'images médicales.

### 6.2 Limitations

Dans un premier temps, nous avons rencontré des difficultés liées à la puissance de calcul nécessaire pour entraîner notre modèle, nous conduisant à utiliser la plateforme Saturn Cloud. L'idéal aurait évidemment été de disposer de nos propres ressources de calcul dédiées pour avoir des apprentissages plus rapides.

Aussi, le temps limité réservé à ce challenge a été une contrainte majeure, nous restreignant à peu (voire pas du tout) explorer certains des leviers d'optimisation possibles.

### 6.3 Améliorations possibles

Plusieurs pistes améliorations restent inexplorées par notre équipe, nous allons dresser la liste que nous pensons utile à l'amélioration de notre présent modèle.

Une extension du processus d'augmentation de données, par exemple sur la base de données non étiquetée après la génération de pseudo-étiquettes, aurait permis de renforcer la généralisation de notre modèle. Aussi, l'utilisation d'autres techniques d'augmentation de données (telles que l'élastic distortion ou les transformations non linéaires) aurait pu permettre d'enrichir la variabilité dans les données et donc de simuler d'autant plus la réalité des différents scans.

Nous aurions pu aussi optimiser les différents hyper-paramètres de notre modèle avec des méthodes de validation croisée ou d'optimisation bayésienne. Une évaluation plus rigoureuse de ces paramètres aurait pu permettre une meilleure généralisation.

L'amélioration du processus de création des pseudo-étiquettes aurait pu être réfléchi, notamment en utilisant des algorithmes plus approfondis tels que des méthodes basées sur les générateurs antagonistes (GAN) ou encore sur Teacher/Student. Des pseudo-étiquettes plus qualitatives auraient nécessairement amélioré la robustesse de notre modèle.

Enfin, nous aurions pu revoir complètement notre approche en utilisant une architecture de base [UNet++] complètement différente et nous diriger vers les Transformers [7] par exemple, avec le mécanisme d'attention. Un benchmark des différents modèles nous aurait permis de savoir lequel aurait été le plus prometteur pour se pencher sur l'optimisation de ce dernier.



## Bibliographie

- [1] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3) :297–302, 1945.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [3] Kasun Hettihewa, Thananop Kobchaisawat, Natthaporn Tanpowpong, and Thanarat H. Chali-dabhongse. Manet : a multi-attention network for automatic liver tumor segmentation in computed tomography (ct) imaging. *Scientific Reports*, 13(1) :20098, Nov 2023.
- [4] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [5] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaiane, and Martin Jagersand. U2-net : Going deeper with nested u-structure for salient object detection. *Pattern Recognition*, 106 :107404, 2020.
- [6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [7] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M. Patel. Medical transformer : Gated axial-attention for medical image segmentation. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 36–46, Cham, 2021. Springer International Publishing.
- [8] D. Vargas-Vazquez, J. Crespo, V. Maojo, and I.R. Terol-Villalobos. Medical image segmentation using openings and closings with reconstruction criteria. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 3, pages III–981, 2003.
- [9] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++ : A nested u-net architecture for medical image segmentation. In Danail Stoyanov, Zeike Taylor, Gustavo Carneiro, Tanveer Syeda-Mahmood, Anne Martel, Lena Maier-Hein, João Manuel R.S. Tavares, Andrew Bradley, João Paulo Papa, Vasileios Belagiannis, Jacinto C. Nascimento, Zhi Lu, Sailesh Conjeti, Mehdi Moradi, Hayit Greenspan, and Anant Madabhushi, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11, Cham, 2018. Springer International Publishing.