

Segmentation cardiaque en IRM

Meryem Sikouky

Décembre 2023

Abstract

Avec l'émergence de l'apprentissage profond ces dernières années, l'utilisation de nouveaux outils tels que les réseaux neuronaux convolutifs pour extraire des informations des données est devenue une option privilégiée. Ceci est particulièrement vrai dans le domaine de l'imagerie médicale, où la segmentation des images joue un rôle crucial pour des tâches telles que la détermination du type de maladie et la localisation des lésions, domaines où l'apprentissage profond s'avère particulièrement efficace. Parmi les structures de modèles profonds, U-Net se distingue par ses excellents résultats dans la segmentation des images médicales. Dans ce contexte, l'analyse et l'interprétation des images de résonance magnétique cardiaque (IRM cardiaque) sont souvent chronophages. La segmentation automatisée des structures cardiaques peut réduire le temps nécessaire à l'analyse des images. Dans cet article, nous mettons en œuvre l'architecture U-Net [1] en partant de zéro en utilisant les blocs de base de Keras et évaluons sa performance sur un ensemble de données d'images cardiaques dans le but est de pouvoir séparer l'épicarde et l'endocarde.

1 Introduction

Le réseau de neurones convolutif (CNN) a eu un impact considérable sur le traitement des images, améliorant nettement la précision des tâches telles que la classification d'images et la reconnaissance d'objets. Cependant, il existe un autre type de tâches nécessitant la localisation pour mettre en évidence une caractéristique ou un objet spécifique dans une image, ce qu'on appelle la segmentation d'image. Un modèle de segmentation d'image attribue une étiquette à tous les pixels d'une image pour localiser des objets à l'intérieur de celle-ci. La segmentation d'image trouve son application dans de nombreux domaines tels que le transport, la conception de logements, la détection d'objets et l'imagerie médicale, ce dernier étant notre principal centre d'intérêt.

Une tâche importante et complexe pour les chercheurs biomédicaux et les radiologues consiste à reconnaître des caractéristiques dans une image médicale, comme une cellule cancéreuse ou une cavité dentaire. L'interprétation d'une image biomédicale exige une expertise spécialisée. En pratique, cela peut s'avérer difficile même pour un spécialiste expérimenté. Par conséquent, disposer d'un modèle qui souligne les points critiques sur une image médicale peut être très utile pour le diagnostic et le traitement des maladies. Un tel modèle doit être très précis, car il affecte directement le processus de traitement des patients. D'autre part, le modèle proposé doit atteindre une haute performance avec des ensembles de données relativement petits. L'annotation d'images médicales nécessitant des spécialistes qualifiés, il est coûteux de fournir un grand nombre d'échantillons. Il est courant de n'avoir que quelques centaines d'images comme ensemble d'entraînement pour un modèle de segmentation.

Plusieurs tentatives ont été faites pour relever ces

défis. Ciresan et al. [2] ont formé un réseau dans une configuration de fenêtre glissante pour prédire l'étiquette de classe de chaque pixel en fournissant une région locale (patch) autour de ce pixel en entrée. Ce modèle est localisé et augmente le nombre de données d'entraînement en utilisant des patchs d'images. Cependant, cette stratégie présente deux inconvénients : le modèle est lent, et il y a un compromis entre la localisation et la taille des patchs. D'autres travaux ont proposé une sortie de classificateur qui prend en compte les entrées de plusieurs couches, ce qui pourrait améliorer l'approche de Ciresan et al. pour obtenir une bonne localisation et l'utilisation simultanée du contexte (patches).

Cependant, parmi les modèles proposés pour la segmentation d'image, UNet a eu un impact significatif sur ce type de tâche, en générant un modèle qui est rapide et localisé tout en utilisant l'image entière comme entrée. Même plusieurs années après, UNet reste le fondement de nombreux nouveaux modèles de segmentation tels que UNet++, attention UNet, etc. Ces innovations lui permettent d'atteindre une haute précision même avec de petits ensembles d'entraînement. Le réseau UNet convient à une majorité des types de problèmes de segmentation biomédicale et permet la segmentation d'images de taille arbitrairement grande. Et c'est ce réseau qui sera l'objet de notre étude en l'appliquant sur la segmentation de l'endocarde et l'épicarde issus des images IRM cardiaques.

2 Techniques de segmentation des images médicales

Avant de plonger dans les détails complexes de l'architecture UNet, nous allons présenter dans cette section un aperçu de quelques des techniques classiques de

segmentation d’images . Nous notons:

Segmentation Basée sur les Seuils (Thresholding) : Cette méthode simple et rapide segmente l’image en fixant un seuil T . Les pixels en un point x, y au-dessus de ce seuil sont considérés comme faisant partie d’un objet, tandis que les autres sont considérés comme l’arrière-plan. Elle est efficace pour les images avec un fort contraste entre l’objet et l’arrière-plan.

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (1)$$

Segmentation Basée sur les Régions (Region-Based Segmentation) : Cette technique inclut des méthodes comme la croissance de régions (region growing) et la division de régions (region splitting). Elle consiste à regrouper les pixels voisins ayant des propriétés similaires, comme l’intensité ou la texture.

Segmentation par Contour (Edge Detection) : Cette méthode détecte les bords ou les contours des objets dans une image, en utilisant des détecteurs de bords comme le filtre de Sobel ou de Canny. Elle est particulièrement utile pour identifier les limites des structures ou des organes.

Segmentation par Clustering : Cette méthode regroupe les pixels en clusters en fonction de leurs caractéristiques. Les algorithmes comme K-means ou la segmentation basée sur les mélanges gaussiens (GMM) sont fréquemment utilisés.

Segmentation watershed: Elle est inspirée du concept géographique de bassin versant. Dans cette technique, l’image est interprétée comme un relief où l’intensité de chaque pixel représente une altitude. Le processus commence par l’identification des points les plus bas de l’image, suivie d’une inondation virtuelle qui remplit progressivement les creux. Cette méthode est particulièrement utile pour séparer des objets en contact ou très proches dans l’image, comme dans le cas de cellules se chevauchant en imagerie médicale.

Chacune de ces techniques a ses avantages et inconvénients, et le choix dépend souvent de la nature spécifique de l’image médicale et des objectifs de segmentation. Il existe aussi des méthodes hybrides qui combinent différentes approches, comme la segmentation basée sur les contours et le clustering, pour tirer parti des forces de chaque méthode. Cependant ces techniques présentent souvent la limitation de dépendre de règles fixes ou de paramètres prédéfinis.

En revanche, l’avantage principal des dernières techniques de la segmentation utilisant le deep learning, par rapport aux méthodes classiques, réside dans sa capacité à apprendre directement à partir des données. Il permet au modèle d’identifier et d’adapter ses critères de segmentation en fonction des caractéristiques complexes et variées des images. Les modèles de deep learning, comme les CNN et U-Net, sont particulièrement efficaces pour traiter des images aux motifs subtiles et complexes, typiques de l’imagerie médicale. Ils peuvent apprendre

à reconnaître des motifs et des structures spécifiques, même dans des conditions où le contraste est faible ou les structures sont irrégulières, ce qui est souvent difficile pour les méthodes conventionnelles. En outre, ces modèles sont capables de gérer de grands volumes de données et de s’améliorer continuellement avec l’ajout de nouvelles données, offrant ainsi une précision et une robustesse accrues. Ils sont également plus adaptés à la segmentation de structures anatomiques variées et peuvent être entraînés pour s’adapter à différents types d’images médicales, ce qui les rend polyvalents et efficaces dans diverses applications médicales.

3 Matériels et Méthodes

3.1 Dataset

Dans ce travail, nous avons comme objet de segmenter, en utilisant l’architecture UNET, l’endocarde et l’épicarde issus des coupes de ciné IRM 2D petit axe. Les coupes de cœur en "petit axe" sont des vues spécifiques obtenues lors d’une imagerie par résonance magnétique (IRM) ou échocardiographie du cœur. Elles fournissent une vue transversale du cœur, perpendiculaire à l’axe longitudinal de celui-ci, et sont cruciales pour l’évaluation détaillée de la structure cardiaque et de la fonction ventriculaire.

Pour cela, nous avons des images 3D+t de la base de données de 30 patients qui ont été converties en coupes .png avec comme nom du fichier: nn_zz_tt où:

- **nn**: Le numéro du patient.
- **zz**: Le niveau de coupe normalisé entre 0 et 1. Dans les coupes de cœur en petit axe, il y a typiquement trois niveaux principaux : apical, médian (ou mi-ventriculaire) et basale comme montré dans la figure 1:
 - *Coupe Basale "Base"*: Située près de la base du cœur, cette coupe englobe les valvules atrioventriculaires (mitrale et tricuspide), ainsi que les oreillettes. C’est le niveau le plus proche des veines pulmonaires et de l’aorte. Dans cette coupe, on évalue souvent la fonction des valves et la structure des oreillettes.
 - *Coupe Médiane "Mid"* (Mi-Ventriculaire): Cette coupe tranche le cœur approximativement au milieu des ventricules. Elle est utilisée pour évaluer l’épaisseur et le mouvement de la paroi ventriculaire, ainsi que la masse et le volume ventriculaires. C’est un niveau clé pour l’évaluation de la fonction systolique du cœur.
 - *Coupe Apicale "Apex"*: Située près de l’apex du cœur, cette coupe montre les parties terminales des ventricules. Les coupes apicales sont particulièrement importantes pour détecter des anomalies régionales de la contraction cardiaque, souvent associées à des conditions telles que l’infarctus du myocarde.

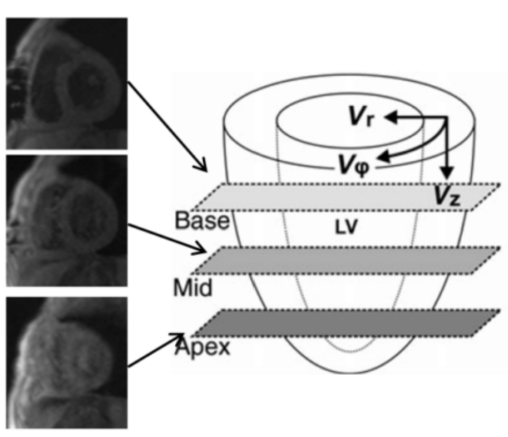


Figure 1: Niveaux de coupes en petit axe du coeur

- **tt**: Le temps dans le cycle cardiaque normalisé entre 0 et 1. Le cycle cardiaque est le processus par lequel le cœur se contracte et se relâche pour pomper le sang à travers le corps et les poumons. Il se compose de deux phases principales : la systole et la diastole.
-*Systole*: C'est la phase de contraction du cœur. Elle commence par la contraction des oreillettes, poussant le sang dans les ventricules (systole auriculaire), suivie de la contraction des ventricules qui propulsent le sang vers les poumons et le reste du corps (systole ventriculaire).
-*Diastole*: C'est la phase de relaxation du cœur. Les ventricules se détendent, se remplissant de sang en provenance des oreillettes (diastole ventriculaire), puis les oreillettes se remplissent de sang venant des veines (diastole auriculaire).

3.2 Architecture UNET

Afin de pouvoir ségmenter l'épicarde et l'endocarde dans les images IRM cardiaques, nous avons implémenté dans ce travail l'architecture UNET proposée dans [1]. Elle s'agit d'un type de réseau de neurones convolutif spécialement conçu pour la tâche de segmentation d'images, très prisé dans le domaine de l'imagerie médicale. Sa structure se distingue par une forme en U, d'où son nom. La figure 2 schématise le principe de fonctionnement de cet algorithme.

L'architecture Unet est décomposée comme suit:

- **Le chemin de contraction, ou la partie d'encodage**, suit l'architecture typique d'un réseau de neurones convolutif. Il est composé de plusieurs couches convolutives, chacune suivie d'une fonction d'activation, généralement la ReLU (Rectified Linear Unit). Après chaque couche ou groupe de couches convolutives, une opération de pooling, souvent un max-pooling, est effectuée pour réduire la dimension spatiale de l'image. Cette réduction de taille à chaque étape aide à capturer le contexte global de l'image, c'est-à-dire les caractéristiques générales et larges.

- **Le chemin d'expansion, ou la partie de décodage**, le processus inverse est effectué pour reconstruire l'image segmentée à partir des caractéristiques encodées. Ce chemin comprend des couches de convolution transposée, ou up-convolution, pour augmenter progressivement la dimensionnalité spatiale de l'image. À chaque étape de cette partie, les caractéristiques extraites du chemin de contraction correspondant sont ajoutées (concaténées). Cette technique, connue sous le nom de "**copie et recadrage**" (**Skip connections**), est essentielle pour restaurer les informations spatiales perdues pendant le pooling. Des couches convolutives suivent chaque étape d'up-convolution pour affiner davantage les caractéristiques.
- La couche finale de l'U-Net est généralement une couche convolutive avec une fonction d'activation adaptée, comme la softmax pour les tâches de classification multi-classes. Cette couche produit la carte de segmentation finale, attribuant à chaque pixel une catégorie cible.

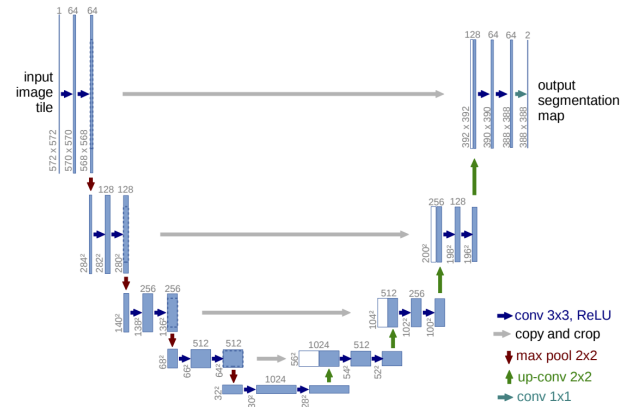


Figure 2: Architecture Unet

L'aspect le plus distinctif de l'U-Net est ses connexions de copie et recadrage entre les chemins de contraction et d'expansion. Ces connexions permettent de conserver des informations de localisation précises, cruciales pour une segmentation de haute précision. La combinaison de la capture du contexte global de l'image dans le chemin de contraction et la préservation des détails spatiaux dans le chemin d'expansion rend l'U-Net particulièrement efficace. Cette efficacité est maintenue même avec un nombre limité de données d'entraînement, ce qui est souvent le cas dans les applications d'imagerie médicale.

3.3 Implémentation de l'architecture UNET

Pour implémenter l'architecture en question, nous avons utilisé la bibliothèque Python Keras. Il sert d'interface pour TensorFlow, offrant une manière plus simple et plus

intuitive de construire des modèles de deep learning, tout en permettant aux utilisateurs de tirer parti de la robustesse, de la performance et de la flexibilité de TensorFlow.

3.3.1 Création et structuration des données

Au départ, nous disposons de deux dossiers:

- **Frames:** contenant les images IRM qui feront l'objet de notre segmentation.
- **Masks:** contenant les masques qui permettent d'isoler des régions spécifiques ou des structures d'intérêt dans une image. Ils sont utilisés comme "vérité terrain (Ground Truth)" dans la formation de modèles de segmentation. Ils indiquent au modèle les zones exactes à identifier et à séparer dans les images. En comparant les prédictions du modèle aux masques de référence, on peut évaluer la précision de la segmentation, ce qui est crucial pour vérifier la performance du modèle. Nous distinguons de sous dossiers des masques: Celui correspondant à l'endocarde et celui correspondant à l'épicarde.

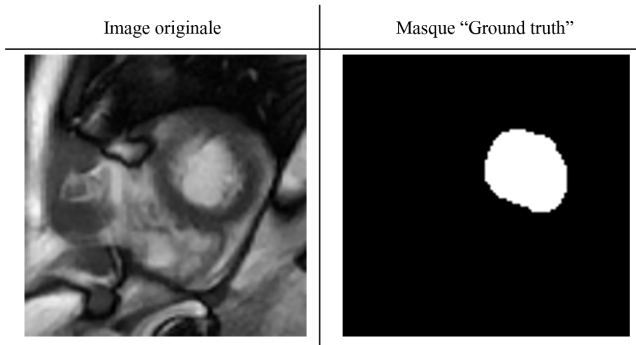


Figure 3: Image 1_0.266667_0.034 et son masque d'endocarde

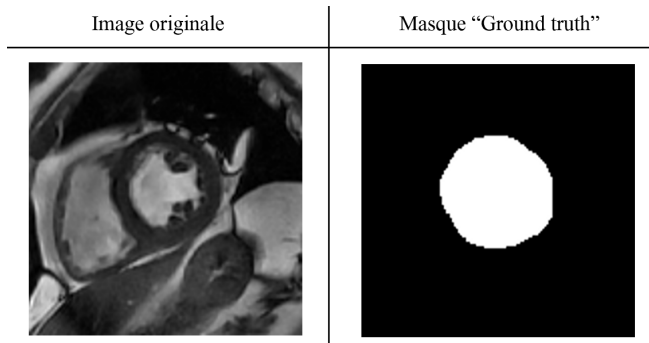


Figure 4: Image 27_0.500000_0.793103 et son masque d'épicarde

Pour préparer cet ensemble de données d'images et de masques pour la segmentation d'images, nous avons suivi les étapes suivantes:

- La configuration initiale comprend la définition des chemins d'accès aux images et aux masques, ainsi que la spécification des critères pour sélectionner les masques basés sur le numéro de patient, les niveaux de coupe (z) et les intervalles de temps (t) afin de s'assurer de la bonne correspondance entre les images et les masques. La taille standard des images est fixée à 128x128 pixels. Des dossiers sont créés pour organiser les images et les masques en ensembles d'entraînement (train), de validation (val) et de test.
- Ensuite, nous procédons à la lecture de tous les noms de fichiers d'images et de masques. Les masques sont sélectionnés en fonction des critères prédéfinis de niveau de coupe et de temps, en utilisant des expressions régulières pour extraire ces valeurs des noms de fichiers. Les masques sélectionnés sont ensuite mélangés aléatoirement et répartis en ensembles pour **l'entraînement (70 %)**, **la validation (20 %)** et **le test (10 %)**.
- Pour chaque ensemble, nous avons créé des dossiers correspondants pour y stocker les images et les masques. Les fonctions *add_frames* et *add_masks* servent à lire, redimensionner et sauvegarder les images et les masques dans les dossiers appropriés, standardisant leur taille à 128x128 pixels.

Ce processus est typique d'une étape de prétraitement en apprentissage profond, où les données sont organisées et préparées avant d'être utilisées pour entraîner un modèle de segmentation, tel que U-Net.

3.3.2 Augmentation des données

Avant de commencer l'entraînement d'un modèle en deep learning, il est courant et souvent bénéfique d'effectuer une augmentation des données. Elle consiste à appliquer une série de transformations géométriques et photométriques sur les images originales pour créer des variantes supplémentaires. Nous notons les raisons clés pour lesquelles nous avons choisi d'effectuer l'augmentation des données avant l'entraînement :

- **Augmentation de la Diversité des Données :** L'augmentation crée des variations des images d'entraînement et leur masque correspondant, ce qui aide le modèle à apprendre à généraliser à partir d'un ensemble plus diversifié d'exemples.
- **Prévention de l'Overfitting :** Dans les cas où les données d'entraînement sont limitées, l'augmentation des données peut aider à prévenir l'overfitting, car le modèle est moins susceptible d'apprendre par cœur les détails spécifiques des images d'entraînement.
- **Amélioration de la Robustesse du Modèle :** En entraînant le modèle sur des images qui ont subi diverses transformations (comme des rotations, des changements d'échelle, des translations,

etc.), on améliore sa capacité à reconnaître des patterns sous différents angles, tailles et conditions d'éclairage.

• Simulation de Conditions Réelles :

L'augmentation peut aider à simuler des variations qui se produisent dans les données réelles mais qui ne sont pas suffisamment représentées dans l'ensemble d'entraînement initial.

ImageDataGenerator de Keras, une classe utilisée pour augmenter les images et préparer les données pour l'entraînement de modèles de deep learning. Il génère avec ces paramètres un flux d'images où chaque image est légèrement modifiée (décalée, zoomée, normalisée, tournée) par rapport aux images originales. Il permet aussi de normaliser les pixels des images. Chaque pixel est divisé par 255, de sorte que les valeurs des pixels sont normalisées pour se situer dans l'intervalle $[0, 1]$. Cette étape est importante pour que le modèle puisse mieux traiter les valeurs d'entrée. Nous notons dans la figure 5 un exemple des ces opérations:

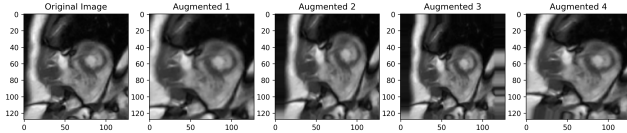


Figure 5: Exemple d'augmentation d'image

3.3.3 Entraînement "Training" et Optimisation

Durant l'entraînement, nous avons séparé les données en **Batch** et **Epochs**. Elles sont divisées en petits lots (batches) de taille $BATCH_SIZE=64$ pour l'entraînement. Cette approche permet de réduire la charge mémoire et d'améliorer la stabilité de l'entraînement en mettant à jour les poids du modèle après chaque lot plutôt qu'après l'ensemble du jeu de données. Un epoch représente une itération complète sur l'ensemble des données d'entraînement. L'utilisation de plusieurs epochs (définie par $NO_OF_EPOCHS = 100$) permet au modèle de voir les données plusieurs fois, améliorant ainsi l'apprentissage.

Pour effectuer cet entraînement, nous avons choisi comme paramètres:

1. Fonction de perte

Nous avons choisi la fonction de perte "binary crossentropy" (entropie croisée binaire) qui est une mesure couramment utilisée pour évaluer la performance d'un modèle de classification binaire. Dans le contexte de la segmentation d'images, où chaque pixel est classé indépendamment comme appartenant à l'une des deux classes (par exemple, objet ou arrière-plan), la formule de la binary cross-

entropy est la suivante :

$$BCE = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \quad (2)$$

où:

- N est le nombre total de pixels dans toutes les images du lot.
- y_i est la véritable étiquette du i -ème pixel/observation (0 ou 1).
- p_i est la probabilité prédite par le modèle pour le i -ème pixel/observation d'appartenir à la classe 1 (classe objet).

Cette fonction calcule la perte pour chaque pixel en prenant en compte la probabilité prédite et la compare à sa véritable étiquette. Pour un pixel donné :

- Si la véritable étiquette est 1 (l'objet), la perte est $-\log(p_i)$, qui augmente à mesure que p_i diminue.
- Si la véritable étiquette est 0 (l'arrière-plan), la perte est $-\log(1 - p_i)$, qui augmente à mesure que p_i augmente.

La perte totale est la moyenne de la perte calculée pour chaque pixel, fournissant une mesure globale de la performance du modèle sur l'ensemble du lot ou de l'ensemble de données. Cette fonction est particulièrement efficace pour les problèmes où il y a une distinction claire entre l'objet et l'arrière-plan, et elle fonctionne bien lorsque les classes (objet et arrière-plan) sont plus ou moins équilibrées dans l'image. Mais elle a comme limitation le traitement de chaque pixel individuellement et ne prend pas en compte les relations spatiales entre les pixels, ce qui peut être un inconvénient pour certaines tâches de segmentation complexes.

2. Optimiseur

Nous avons choisi comme algorithme d'optimisation **ADAM (Adaptive Moment Estimation)** qui ajuste les poids du réseau pour minimiser la fonction de perte. Il combine les avantages des deux autres optimiseurs populaires, AdaGrad et RMSProp. Cet algorithme ajuste les poids du réseau en utilisant non seulement les gradients du premier ordre (comme dans la descente de gradient classique), mais aussi en prenant en compte les seconds moments des gradients (c'est-à-dire, le taux de changement des gradients).

Les poids du modèle sont mis à jour comme suit:

$$\theta_{t+1} = \theta_t - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (3)$$

où:

- θ_{t+1} sont les poids du modèle mis à jour.
- θ_t sont les poids actuels du modèle.
- α est le taux d'apprentissage.
- m_t le moment de premier ordre des gradients.

- v_t le moment de second ordre des gradients.
- ϵ un petit nombre pour éviter la division par zéro.

Cet optimiseur a comme avantage l'ajustement du taux d'apprentissage pour chaque poids du modèle grâce à l'estimation des seconds moments. D'une autre part, il fonctionne bien même lorsque les gradients sont irréguliers ou bruités. Cependant, il peut nécessiter plus de mémoire que des optimiseurs plus simples en raison du stockage des moments pour chaque poids.

3. Métriques

Le coefficient Dice, également connu sous le nom de Dice Similarity Coefficient (DSC), est une métrique utilisée pour quantifier la similarité entre deux ensembles de données. Dans le contexte de la segmentation d'images, il sert à mesurer la similarité entre les zones segmentées par le modèle (prédictions) et les véritables zones (vérité terrain). Il est calculé comme suit:

$$DSC = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (4)$$

où:

- X représente l'ensemble des pixels de la vérité terrain (masque réel).
- Y représente l'ensemble des pixels de la prédiction de segmentation.
- $|X \cap Y|$ est le nombre de pixels communs entre la vérité terrain et la prédiction (c'est-à-dire les vrais positifs).
- $|X|$ et $|Y|$ sont les nombres totaux de pixels dans les ensembles de la vérité terrain et de la prédiction respectivement.

Le coefficient Dice varie entre 0 et 1, où 1 indique une correspondance parfaite entre la prédiction et la vérité terrain, et 0 indique aucune correspondance. Le coefficient est particulièrement utile pour évaluer la précision de la segmentation, car il est très sensible aux vrais positifs. Il prend en compte à la fois les faux positifs (erreurs de segmentation) et les faux négatifs (zones manquées). Cependant, dans les cas où la taille des régions varie considérablement, le coefficient Dice pourrait favoriser les grandes régions.

En résumé, pendant l'entraînement, les poids du modèle sont ajustés en fonction de la fonction de perte calculée sur les données d'entraînement. La performance du modèle est également évaluée sur un ensemble de validation, qui n'est pas utilisé pour l'entraînement mais sert à surveiller et à prévenir l'overfitting. Nous avons utilisé aussi des callbacks *ModelCheckpoint* et *EarlyStopping* pour enregistrer les meilleurs poids (basés sur la perte de validation) et arrêter l'entraînement si le modèle ne s'améliore plus sur l'ensemble de validation (la performance sur l'ensemble de validation ne s'améliore pas pendant un nombre défini d'epochs (patience)).

4 Résultats et discussion

4.1 Résultats d'entraînement et de validation

Afin d'illustrer et de comprendre la performance du modèle tout au long du processus d'entraînement et de validation, nous allons illustrer des résultats obtenus de l'outil de visualisation **Tensorboard**. Plus précisément, nous nous intéressons des graphes représentant l'évolution de la fonction de perte et le coefficient Dice pour les deux cas de figures de segmentation à savoir l'endocarde et l'épicarde.

- **Pour l'endocarde:**

- **Fonction de perte**(Validation en bleu, Training en noir dans la figure 6): Le loss de validation et d'entraînement diminue au fil des epochs, indiquant que le modèle s'adapte bien aux données. La courbe de validation montre que le modèle généralise bien puisque la perte de validation ne commence pas à augmenter à mesure que le loss d'entraînement diminue, ce qui aurait indiqué un surajustement (overfitting).

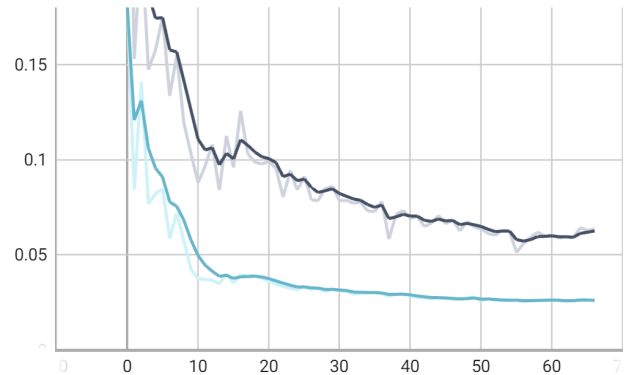


Figure 6: Evolution de la fonction de perte pour l'endocarde

- **Dice Coefficient**(Validation en bleu, Training en noir dans la figure 7): Le score Dice pour l'entraînement augmente de manière significative au début, ce qui indique que le modèle apprend rapidement à segmenter l'endocarde correctement. Le score de validation suit une tendance similaire, bien qu'il soit généralement plus haut que celui de l'entraînement, ce qui est assez atypique. Cela pourrait indiquer que les données de validation sont plus simples à segmenter ou que le modèle généralise bien. La convergence du score de validation et d'entraînement vers la fin des epochs suggère que le modèle atteint une performance stable.

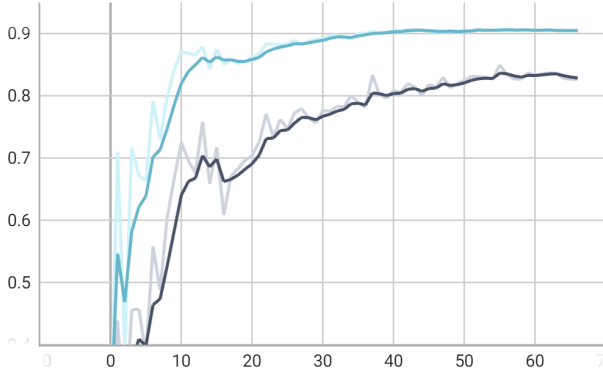


Figure 7: Evolution du coefficient de Dice pour l'endocarde

- Pour l'épicarde:

- **Fonction de perte**(Validation en orange, Training en rose dans la figure 8): comme pour l'endocarde, la fonction de perte pour l'épicarde diminue également, suggérant que le modèle apprend de manière appropriée. La variabilité dans le loss de validation est plus notable ici, ce qui pourrait indiquer des problèmes de généralisation ou que les données de validation contiennent des exemples plus variés ou difficiles.

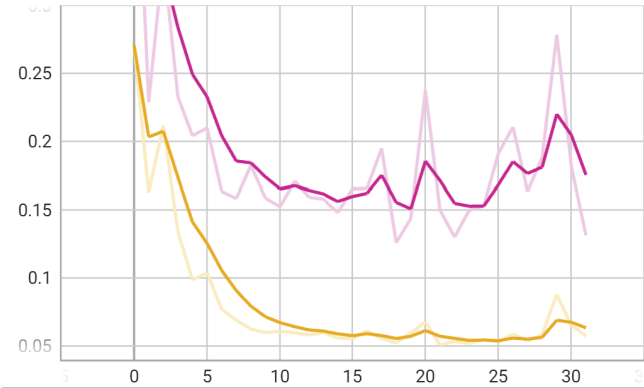


Figure 8: Evolution de la fonction de perte pour l'épicarde

- **Dice Coefficient**(Validation en orange, Training en rose dans la figure 9): L'augmentation du Dice suggère que la qualité de la segmentation s'améliore. Cependant, la variabilité dans le score de validation et l'absence d'une tendance à la convergence avec le score d'entraînement pourraient indiquer que le modèle n'est pas aussi stable ou que les hyperparamètres nécessitent un ajustement.

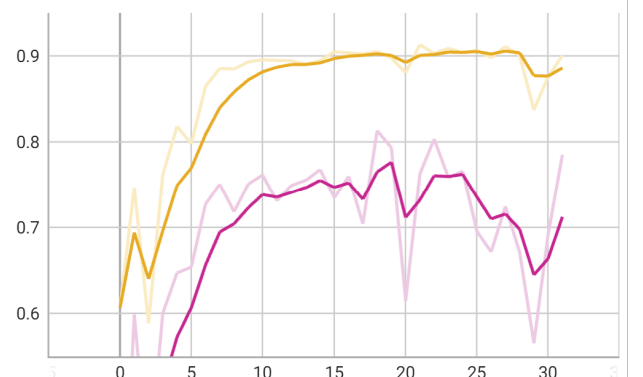


Figure 9: Evolution du coefficient de Dice pour l'épicarde

En résumé, pour l'endocarde, les deux métriques indiquent que le modèle apprend et généralise bien, mais l'anomalie où le Dice de validation est supérieur au Dice d'entraînement mérite attention. Cela pourrait être dû à une sélection de données de validation qui ne sont pas suffisamment difficiles ou à une évaluation sur des cas particulièrement bien segmentés par le modèle. Pour l'épicarde, alors que les tendances générales sont positives, la plus grande variabilité dans les scores de validation pour les deux métriques indique que le modèle pourrait bénéficier d'une meilleure régularisation ou d'une validation croisée pour assurer une généralisation robuste.

4.2 Résultats de la prédiction

4.2.1 Métriques utilisées

Afin d'évaluer la performance de notre modèles, nous allons calculer la moyenne des trois métriques suivants:

- **Average Surface Distance (ASD)**: Elle mesure la distance moyenne entre les surfaces des objets segmentés dans les images prédites et les images de référence (ground truth). La formule de l'ASD est:

$$ASD = \frac{1}{|S_1| + |S_2|} \left(\sum_{x \in S_1} \min_{y \in S_2} d(x, y) + \sum_{y \in S_2} \min_{x \in S_1} d(y, x) \right) \quad (5)$$

où:

- S_1 est l'ensemble des points de la surface de l'objet segmenté dans la prédiction.
- S_2 est l'ensemble des points de la surface de l'objet segmenté dans la référence.
- $d(x, y)$ est la distance euclidienne entre les points x et y .
- **Hausdorff Distance (HD)**: une métrique utilisée pour quantifier à quel point deux ensembles de points (typiquement, les contours de la segmentation dans les images médicales) sont éloignés

l'un de l'autre. La Hausdorff Distance est particulièrement utile pour mesurer la plus grande des distances de tous les points d'un ensemble à l'ensemble le plus proche dans l'autre ensemble. Cette distance entre deux ensembles de points A et B est définie comme suit:

$$HD(A, B) = \max\left\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(b, a)\right\} \quad (6)$$

où:

-*sup* représente le supremum (le plus grand élément) d'un ensemble.

-*inf* représente l'infimum (le plus petit élément) d'un ensemble.

-*d(a, b)* est la distance (généralement euclidienne) entre les points *a* et *b*.

En général, *A* représente les points sur le contour (dans le cas de la segmentation 2D) ou sur la surface (dans le cas de la segmentation 3D) de la région segmentée par le modèle de machine learning ou l'algorithme de segmentation. Ces points sont issus de la segmentation prédite par le modèle. *B* représente les points sur le contour ou la surface de la région segmentée dans l'image de référence "ground truth". Cette référence est typiquement établie par des experts humains ou par un autre moyen considéré comme véridique et précis.

- **Dice Coefficient:** Même définition présentée dans la section 3.

Chacune de ces métriques a ses propres utilisations et doit être choisie en fonction de l'aspect spécifique de la performance de segmentation que nous souhaitons évaluer. Par exemple, le Dice Coefficient est excellent pour une évaluation générale de la similitude, tandis que la Mean Distance et la Hausdorff Distance sont plus précises pour évaluer la précision géométrique des contours segmentés.

4.2.2 Analyse des prédictions

En évaluant les trois métriques décrites dans la partie précédente, nous obtenons les résultats présentés dans le tableau ci-dessous pour les deux catégories de segmentation discutées endocarde (Endo) et épicarde (Epi).

Dataset	Dice	Mean Distance	Hausdorff distance
Endo			
Train	92.4	3.01	9.2
Validation	92.2	3.02	9.32
Test	92.6	3.86	12.56
Epi			
Train	93.3	4.39	14.46
Validation	92.9	4.68	15.1
Test	92.5	5.12	18.97

Nous interprétons les résultats comme suit:

Endocarde

- **Coefficient de Dice:** Nous notons des scores de Dice élevés autour de 92% pour les ensembles de formation (Train), de validation et de test ce qui indique une excellente correspondance globale entre les prédictions de segmentation et la vérité terrain. La cohérence des scores de Dice à travers les ensembles suggère que l'architecture du modèle généralise bien et ne présente pas d'overfitting.
- **Mean distance:** Des valeurs de Mean Distance basses (autour de 3) pour l'entraînement et la validation indiquent que la distance moyenne entre les contours prédits et réels est faible, ce qui est bon. Cependant, il y a une augmentation notable de cette distance dans l'ensemble de test, passant à 3.86, ce qui peut indiquer une légère baisse de la précision des contours dans des cas non vus pendant l'entraînement.
- **Hausdorff distance:** Les valeurs de Hausdorff Distance sont relativement faibles pour l'entraînement et la validation mais augmentent de manière significative dans l'ensemble de test, passant à 12.56. Cela suggère que bien que la plupart des prédictions soient précises, il y a des cas extrêmes où les erreurs de segmentation sont plus importantes.

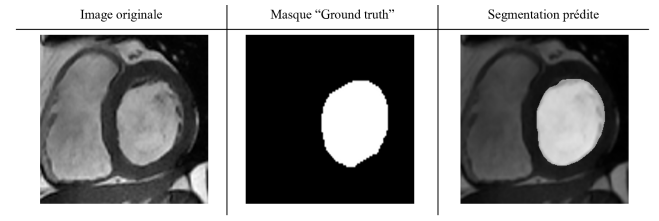


Figure 10: Segmentation d'endocarde prédite pour l'image 2.0.714286.0.827586

Epicarde

- **Coefficient de Dice:** Des scores de Dice légèrement plus élevés pour l'ensemble d'entraînement par rapport à l'endocarde, mais une légère diminution dans les ensembles de validation et de test. Cela indique que l'architecture est très légèrement plus performante sur l'épicarde pour l'entraînement, mais cette performance ne se maintient pas aussi bien lors de la généralisation.
- **Mean distance:** Des valeurs de Mean Distance plus élevées en comparaison avec l'endocarde, et qui augmentent entre l'entraînement et le test. Cela indique que les contours prédits pour l'épicarde sont en moyenne plus éloignés de la vérité terrain que pour l'endocarde, et cette imprécision s'accroît pour les données de test.

- **Hausdorff distance:** Des valeurs plus élevées que pour l'endocarde dans tous les ensembles de données, avec une augmentation particulièrement marquée dans l'ensemble de test, jusqu'à 18.97. Cela montre que les erreurs maximales de segmentation sont plus prononcées pour l'épicarde, particulièrement sur les données de test, ce qui pourrait indiquer des erreurs de segmentation plus extrêmes sur des cas difficiles ou atypiques.

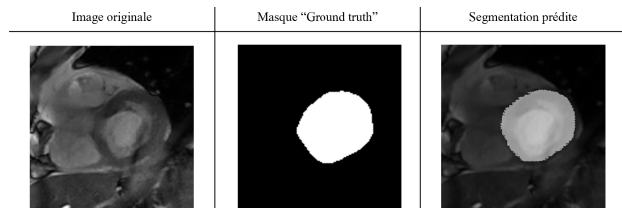


Figure 11: Segmentation d'épicarde prédite pour l'image 5.0.714286.0.448

Les scores élevés de Dice pour les deux ensembles de données à travers les différentes étapes montrent que l'architecture semble fonctionner globalement. Cependant, la hausse des valeurs de la distance moyenne et de la distance hausdorffale sur l'ensemble de test pour les deux ensembles de données pourrait indiquer des problèmes de généralisation, en particulier pour les prédictions d'épicarde. Ces résultats pourraient suggérer de revoir l'architecture du modèle ou les méthodes de régularisation pour améliorer la généralisation, en particulier pour les cas plus complexes ou les données qui ne répondent pas aux conditions d'entraînement standard.

Nous pouvons par la suite envisager d'introduire d'autres techniques d'augmentation de données pour renforcer la robustesse du modèle, d'enrichir l'ensemble de données avec des exemples représentatifs des erreurs de test et d'appliquer des techniques de régularisation pour prévenir le surajustement afin d'améliorer la performance de cette architecture de segmentation. De plus, nous pourrions optimiser davantage les hyperparamètres et envisager l'utilisation d'un apprentissage par transfert, qui pourrait également être avantageux. Une meilleure généralisation peut également être obtenue en explorant des architectures de réseau plus complexes, en intégrant des mécanismes d'attention et en utilisant des techniques d'ensemble. Finalement, il peut être utile de cibler spécifiquement les domaines qui nécessitent une amélioration en modifiant la fonction de perte pour se concentrer sur des métriques importantes telles que la distance moyenne ou la distance Hausdorff.

5 Conclusion

L'article présente une étude détaillée sur l'application de l'architecture U-Net pour la segmentation cardiaque en

IRM, en mettant l'accent sur une identification précise d'endocarde et d'épicarde. Les résultats montrent la capacité de l'U-Net à produire des segmentations de haute qualité, avec des scores de coefficient Dice élevés obtenus sur les ensembles de données de formation, la validation et les tests. Les mesures de la moyenne et de la distance de Hausdorff indiquent des performances prometteuses, bien que certaines limites aient été observées dans la généralisation du modèle pour tester les données, en particulier pour la segmentation épicardique.

Les résultats soulignent l'importance de la sélection rigoureuse des données et de la régularisation appropriée lors de la formation de modèles d'apprentissage profond pour la segmentation de l'image médicale. Ils soulignent également la nécessité d'explorer en permanence les architectures de réseaux neuronaux et les stratégies d'amélioration, telles que l'augmentation des données et l'optimisation de des hyperparamètres, pour améliorer les performances du modèle.

Pour améliorer ces résultats, plusieurs voies peuvent être explorées, y compris l'intégration de techniques d'apprentissage avancées telles que les réseaux neuronaux convolutifs ou GAN (Generative Adversarial Networks) pour une génération de données de formation plus raffinée. Le transfert de l'apprentissage à partir de modèles pré-entraînés sur des ensembles de données similaires pourrait également améliorer la robustesse et la précision de la segmentation, en particulier dans les cas où les données de formation sont limitées.

De nouvelles technologies telles que l'apprentissage fédéré et la segmentation semi-supervisée offrent des possibilités passionnantes pour surmonter les contraintes et les limitations liées à la confidentialité des données liées aux petites tailles d'échantillons. Ces approches permettent à des modèles d'être formés sur des ensembles de données distribués sans accès direct aux données elles-mêmes, offrant un moyen d'améliorer la segmentation tout en respectant la confidentialité des données des patients.

En conclusion, l'efficacité de l'architecture U-Net dans la segmentation cardiaque en MRI constitue la base de futures recherches visant à améliorer les outils de diagnostic et les soins aux patients en cardiologie.

References

- [1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Springer*, 2015.
- [2] Dan Ciresan, Alessandro Giusti, Luca Gamardella, and Jurgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems*, 2012.