Segmentation et suivi temporel automatiques des cavités cardiaques en IRM dynamique

Nicolas Portal

Segmentation et suivi temporel automatiques des cavités cardiaques en IRM dynamique

Présentée par
Nicolas Portal

Thèse de doctorat en informatique

Ecole doctorale Sciences Mécaniques, Acoustique, Electronique et Robotique de Paris (SMAER)

Dirigée par Catherine Achard et co-encadrée par Thomas Dietenbeck

Sorbonne Université
Institut des Systèmes Intelligents et de Robotique (ISIR)
Laboratoire d'Imagerie Biomédicale (LIB)

Présentée et soutenue publiquement le 5 novembre 2024

Composition du jury :

Sylvie Treuillet	Professeure des universités, Université d'Orléans	Président du Jury
Olivier Bernard	Professeur des universités, INSA Lyon	Rapporteur
Fabrice Meriaudeau	Professeur des universités, Université de Bourgogne	Rapporteur
Emilie Pery	Maitresse de conférences, Université Clermont Auvergne	Examinatrice
Nicolas Thome	Professeur des universités, Sorbonne Université	Examinateur
Thomas Dietenbeck	Maitre de conférences, Sorbonne Université	Co-directeur de thèse
Catherine Achard	Professeure des universités, Sorbonne Université	Directrice de thèse

Table des matières

Première partie

Introduction générale

Chapitre 1

Introduction

1.1 Contexte

D'après l'Organisation Mondiale de la Santé (OMS), en 2019, 17.9 millions de personnes sont décédées du fait d'une maladie cardiovasculaire. Cela représente 32% du total des décès dans le monde. De ce fait, les maladies cardiovasculaires constituent un enjeu majeur du 21ème siècle nécessitant la mise en place de mesures de contrôle et de prévention. Parmi ces décès, 85% furent la conséquence d'Accidents Vasculaires Cérébraux (AVC) ou d'infarctus du myocarde ("crise cardiaque"). En étudiant la forme des principales cavités cardiaques, ainsi que leur déformation au cours du temps, il est possible de diagnostiquer ces maladies plus facilement et d'agir plus tôt pour éviter leur aggravation. En particulier, il est intéressant d'analyser la manière dont le muscle cardiaque (myocarde) se contracte et se relâche au cours du cycle cardiaque, car certaines anomalies de contraction peuvent se manifester plus précocement au cours de la maladie que d'autres marqueurs communément utilisés en routine clinique. Néanmoins, actuellement, les logiciels cliniques qui permettent d'étudier cette déformation myocardique au cours du temps ne sont pas entièrement automatisés. En effet, ils nécessitent une initialisation manuelle des contours des cavités cardiaques par les médecins, rendant l'analyse complexe et longue. Par ailleurs, ces logiciels ne traitent souvent que le ventricule gauche alors que les autres cavités peuvent également présenter des irrégularités.

Le projet européen MAESTRIA regroupant 18 partenaires provenant de 9 pays propose de s'attaquer à ce problème et vise à développer de nouvelles approches pour la détection de la cardiomyopathie atriale afin d'améliorer la gestion des soins et d'identifier de nouvelles pistes thérapeutiques pour la médecine personnalisée des AVC. Plus particulièrement, MAESTRIA vise à mettre au point une plateforme pour faciliter le diagnostic de la fibrillation atriale (FA) et des AVC. Cette plateforme doit regrouper des outils permettant d'extraire des informations cliniques à partir de données de différentes formes : biopsie, ECG, imagerie CT, imagerie IRM et échocardiographie. L'utilisation conjointe de ces outils doit permettre d'évaluer le risque de développement de la fibrillation atriale ainsi que de ses conséquences (infarctus, AVC, ...) chez le patient de manière plus précise. Les cardiologues disposeront ainsi d'une plateforme proposant un diagnostic rapide et multifactoriel pour des patients ayant une fibrillation atriale.

Les travaux de cette thèse prennent place au sein du "Work Package 1" du projet

MAESTRIA qui vise à développer, optimiser et valider cliniquement de nouveaux paramètres d'imagerie cardiaque permettant de prédire la cardiomyopathie atriale. Ainsi, nous travaillons sur des images acquises par Imagerie par Résonance Magnétique (IRM), disposant d'un bon contraste pour les tissus mous tout en bénéficiant d'une bonne résolution spatiale. Il s'agit d'une méthode d'acquisition qui n'expose pas les patients à des radiations, ce qui la rend plus sûre pour certains sujets à risque comme les enfants ou femmes enceintes. Ces images permettent aux médecins de clairement visualiser les deux ventricules ainsi que leurs oreillettes et les muscles papillaires. Plusieurs images sont acquises selon plusieurs coupes du cœur permettant de remonter au volume 3D. De plus, ces volumes sont acquis à plusieurs instants du cycle cardiaque de sorte que ces données comportent également une dimension temporelle (3D + t). À partir de ces données, il est possible de calculer à la fois des paramètres cliniques liés à la taille et la forme des structures cardiaques, mais aussi d'obtenir des informations quant à la manière dont ces structures se déforment au cours du temps. Ainsi, l'analyse de données IRM facilite à la fois la détection d'anomalies morphologiques et contractiles. En particulier, il est possible d'extraire des paramètres quantitatifs tels que le volume des ventricules, le volume d'éjection, la fraction d'éjection, la masse du myocarde ou le strain myocardique, paramètres facilitant le diagnostic de nombreuses maladies cardiaques.

1.2 Objectif

Dans ce contexte, l'équipe Imagerie Cardio-Vasculaire (ICV) du LIB a mis au point un logiciel, nommé CardioTrack, qui permet l'analyse semi-automatique de la déformation de toutes les cavités cardiaques à partir de données IRM. Ce logiciel permet d'effectuer le suivi de points de contours des structures cardiaques au sein de séquences d'images IRM (feature tracking) et donc, de remonter à la segmentation des structures d'intérêts. Il permet de réduire radicalement le temps consacré à la segmentation des séquences IRM en comparaison d'un traitement manuel. Grâce à son efficacité, ce logiciel est actuellement utilisé pour traiter de larges cohortes de patients. Néanmoins, ce programme repose sur une initialisation manuelle des contours des structures cardiaques en diastole. Par conséquent, il serait intéressant de pouvoir complètement automatiser le traitement des données IRM en remplaçant la phase d'initialisation manuelle par une segmentation automatique réalisée par apprentissage. Les performances de cet algorithme pourraient également être comparées avec les résultats de suivi du logiciel. Par la suite, nous espérons améliorer l'estimation des paramètres cliniques mentionnés précédemment.

Les travaux effectués durant cette thèse pour analyser ces données s'appuient sur l'utilisation de méthodes d'apprentissage. Ces méthodes optimisent un critère préalablement choisi de façon à accomplir une tâche donnée. Le terme d'apprentissage ou de "Machine Learning" (ML) provient de la manière d'optimiser ce critère, assimilable à un processus d'apprentissage. En effet, il est préalablement nécessaire de collecter une grande quantité de données qui sera séparée en deux groupes. Le premier groupe est utilisé par l'algorithme pour optimiser le critère, il s'agit de la phase d'entrainement. Le second groupe est utilisé une fois le modèle entrainé, pour évaluer les performances sur des données non utilisées pour l'entrainement. Cette

seconde étape correspond à la phase de test qui mesure la capacité du modèle à généraliser ce qui a été appris sur de nouvelles données. Pour accroitre l'efficacité de la phase d'entrainement, il est possible de fournir des annotations associées aux données que le modèle va chercher à prédire, il s'agit d'un apprentissage supervisé. Parmi ces méthodes d'apprentissage, les recherches menées dans le cadre de cette thèse utilisent des techniques dites d'apprentissage profond ou Deep Learning (DL) reposant sur l'utilisation de réseaux de neurones profonds. L'un des principaux atouts de ces réseaux de neurones profonds est leur capacité à extraire par eux-mêmes des "caractéristiques" (features en anglais) relatives aux données et de les utiliser pour optimiser au mieux le critère prédéfini. De ce fait, contrairement à d'autres algorithmes d'apprentissage (SVM par exemple), il n'est pas nécessaire de trouver par soi-même les caractéristiques les plus intéressantes pour résoudre le problème. Les algorithmes de DL sont aujourd'hui largement utilisés dans l'industrie et la recherche pour mener à bien des tâches liées au traitement d'images et à la vision par ordinateur. De ce fait, ils sont particulièrement adaptés à l'analyse d'images médicales.

Un premier objectif de cette thèse est de mettre au point un algorithme de segmentation des structures cardiaques : ventricule gauche, ventricule droit et myocarde sur des images IRM acquises en petit-axe afin d'extraire les paramètres quantitatifs à même de faciliter le diagnostic de pathologies cardiaques. Dans un second temps, une méthode permettant de réaliser le suivi des points de contours de ces structures sera mise au point afin de caractériser leur déformation au cours du cycle cardiaque par l'intermédiaire du calcul de courbes de déformation.

1.3 Verrous scientifiques et contributions

Les architectures de segmentation d'images de type U-net souffrent d'un écart sémantique entre les cartes de caractéristiques de l'encodeur et du décodeur. La concaténation des cartes de caractéristiques de l'encodeur directement avec celles du décodeur par l'intermédiaire des "skip connections" est donc sous-optimale. Il est nécessaire d'augmenter le champ réceptif des cartes de caractéristiques de l'encodeur avant concaténation de façon à réduire cet écart sémantique. Dans cette thèse, nous proposons un mécanisme de filtrage des cartes de caractéristiques de l'encodeur basé sur l'attention. Nous utilisons des blocs Swin transformer de façon à ce que l'attention ne consomme pas trop de ressources de calcul pour ces hautes résolutions.

Le calcul de la déformation myocardique se fait habituellement en calculant la déformation entre une image de référence et toutes les autres images de la séquence. Cela nécessite d'estimer le mouvement entre des images distantes dans la séquence, ce qui rend la tâche plus complexe. Par conséquent, nous proposons une architecture d'apprentissage profond reposant sur deux réseaux. Le premier estime le mouvement entre des images adjacentes tandis que le second a pour objectif d'agréger les mouvements intermédiaires prédits par le premier réseau de façon à obtenir un mouvement entre des images distantes dans la séquence.

L'estimation de la déformation myocardique par apprentissage profond se fait traditionnellement à l'aide d'algorithmes d'estimation du flux optique. Ces algorithmes calculent le déplacement de chaque pixel de l'image. Or, seul le déplacement des pixels se trouvant proche des contours des structures cardiaques est important pour estimer la déformation radiale et circonférentielle du myocarde. Par conséquent, nous avons mis au point, à partir des annotations de segmentation de la première image, des cartes de distance qui sont utilisées pour pondérer les fonctions de coûts durant l'entrainement. De ce fait notre entrainement s'effectue dans un cadre semi-supervisé ce qui permet de s'entrainer sur toutes les images de la séquence sans nécessiter d'annotation de vérité terrain pour chaque image. En outre, cette approche permet de prédire à la fois la déformation globale du muscle cardiaque, mais aussi la déformation myocardique régionale.

1.4 Valorisation des contributions

Article de journal

N. Portal et al. Attention-based neural network for cardiac MRI segmentation : application to strain and volume computation. In Innovation and Research in Bio-Medical engineering (IRBM)

Articles de conférences

N. Portal et al. SFB-net for Cardiac Segmentation : Bridging the Semantic Gap with Attention. In 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)

N. Portal et al. SFB-NET pour la segmentation cardiaque. In Société Francaise de Résonance Magnétique en Biologie et Médecine (SFRMBM), 2023

N. Portal et al. Learning to estimate motion between non-adjacent frames in cardiac Cine MRI data : a fusion approach. In 2024 IEEE 27th International Conference on Pattern Recognition (ICPR)

1.5 Structure du manuscrit

Ce manuscrit se divise en 4 parties.

Dans la première partie, certains concepts clefs relatifs au contexte clinique, au fonctionnement des IRM et aux méthodes d'apprentissage profond sont explicités. Les principaux biomarqueurs ainsi que les principales bases de données pour le traitement d'images IRM sont également présentés.

La seconde partie présente d'abord un état de l'art des méthodes de segmentations avec et sans apprentissage profond. Nous introduisons ensuite un nouvel algorithme de segmentation où l'utilisation de mécanismes d'attention permet d'accroitre les performances d'un réseau de segmentation et d'estimer des biomarqueurs de façon plus précise.

La troisième partie est composée de 3 chapitres. Le premier chapitre décrit d'abord les principales méthodes d'estimation du flux optique avec et sans apprentissage profond. Puis, le deuxième chapitre introduit une première architecture reposant sur l'agrégation de flux de mouvement de manière à suivre les points de contour des structures cardiaques sur les séquences IRM 2D. Cette approche repose essentiellement sur l'utilisation de 2 réseaux de neurones de manière à fusionner les flux de mouvements entre frames voisines dans le but d'obtenir des mouvements entre frames distantes. Enfin, le troisième chapitre propose d'estimer le flux optique à l'aide d'une architecture utilisant un réseau à mémoire de façon à ne plus dépendre de 2 réseaux séparés et à bénéficier, durant l'entrainement, d'une mémoire des précédents mouvements prédits. Cette architecture permet en outre de réduire le nombre de composantes dans la fonction de coût. Lors de l'apprentissage, les différentes composantes de la fonction de coût sont pondérées par des cartes de distances qui donnent plus d'importance à l'estimation des mouvements proches des contours des structures cardiaques.

Enfin la $4^{\text{ème}}$ et dernière partie permet de conclure et d'ouvrir des perspectives sur d'éventuelles améliorations.

Chapitre 2

Contexte clinique et scientifique

2.1 Contexte clinique

2.1.1 Fonctionnement du cœur

Système cardiovasculaire

Le cœur permet d'assurer le bon fonctionnement du corps humain en expulsant du sang oxygéné et porteur de nutriments vers les organes. Il recueille également du sang pauvre en dioxygène et riche en dioxyde de carbone. Il joue ainsi le rôle d'une pompe chargée de renouveler le dioxygène et de drainer les déchets métaboliques du corps, assurant ainsi une circulation sanguine vitale pour maintenir les fonctions corporelles essentielles.

Le cœur (Figure 2.1) se décompose en une partie droite et une partie gauche assurant chacune un rôle différent. Ces deux parties sont composées d'une oreillette et d'un ventricule.

- La partie droite du cœur recueille le sang pauvre en oxygène et chargé en dioxyde de carbone en provenance des organes par les veines caves supérieures et inférieures. Le sang entre d'abord dans l'oreillette droite avant d'être expulsé vers le ventricule droit après l'ouverture de la valve tricuspide. Après la fermeture de cette valve, le sang est éjecté par l'intermédiaire de l'artère pulmonaire vers les poumons pour être renouvelé en oxygène.
- La partie gauche reçoit le sang riche en oxygène en provenance des poumons à travers les veines pulmonaires. Le sang entre d'abord dans l'oreillette gauche avant d'être expulsé vers le ventricule gauche après l'ouverture de la valve mitrale. Après la fermeture de cette valve, le sang est éjecté vers les organes en passant par l'aorte.

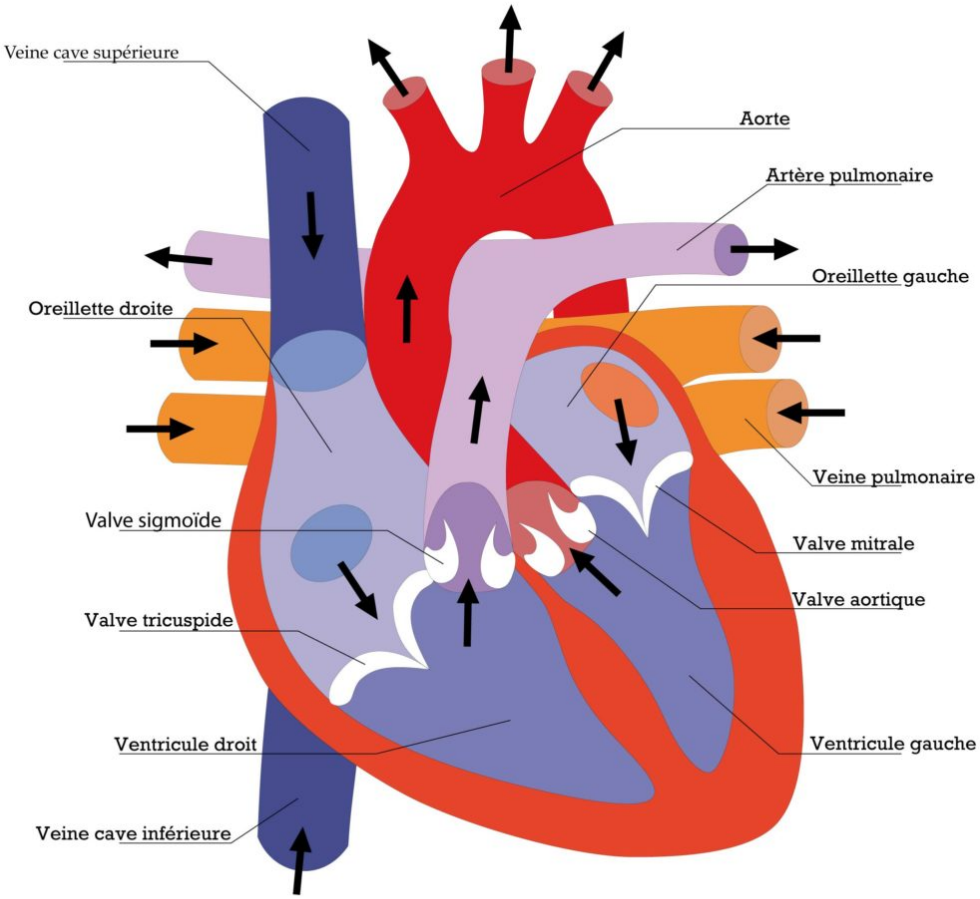


FIGURE 2.1 – Schéma de l'anatomie du cœur [1]

Durant un cycle cardiaque, on distingue la diastole correspondant au relâchement des ventricules et à leur remplissage en sang, et la systole qui, à l'inverse, prend place lorsque les ventricules se contractent pour éjecter le sang vers les poumons ou les organes.

- La diastole coïncide avec l'ouverture des valves atrio-ventriculaires (valves tricuspides et mitrales). Le sang s'écoule alors depuis les veines caves ou pulmonaires vers les ventricules à travers les oreillettes. La relaxation ventriculaire pendant cette phase entraine un gradient de pression entre les oreillettes et les ventricules qui favorise cet écoulement.
- En fin de diastole, la systole atriale a pour rôle de terminer le remplissage des ventricules. Durant cette période, les oreillettes se contractent à la suite du passage d'une onde électrique dans leurs cavités.
- Durant la systole ventriculaire, une stimulation électrique conduit à la contraction des ventricules gauche et droit et à une élévation de la pression dans ces cavités. Le sang est alors éjecté vers l'artère pulmonaire et l'aorte à la suite de l'ouverture des valves sigmoïdes (valves aortiques et pulmonaires). Durant cette période, les oreillettes se relâchent et se remplissent de sang.

1. `https://futurinfirmier.fr/ue-2-2-s1-le-systeme-cardiovasculaire/`

Le myocarde

Le myocarde est le muscle cardiaque responsable de la contraction du cœur. Il est irrigué en sang grâce aux artères coronaires. Comme représenté en Figure 2.2, le myocarde est une couche épaisse située au milieu de la paroi cardiaque, entre l'endocarde (bordure intérieure de la paroi) et l'épicarde (bordure extérieure avant le péricarde).

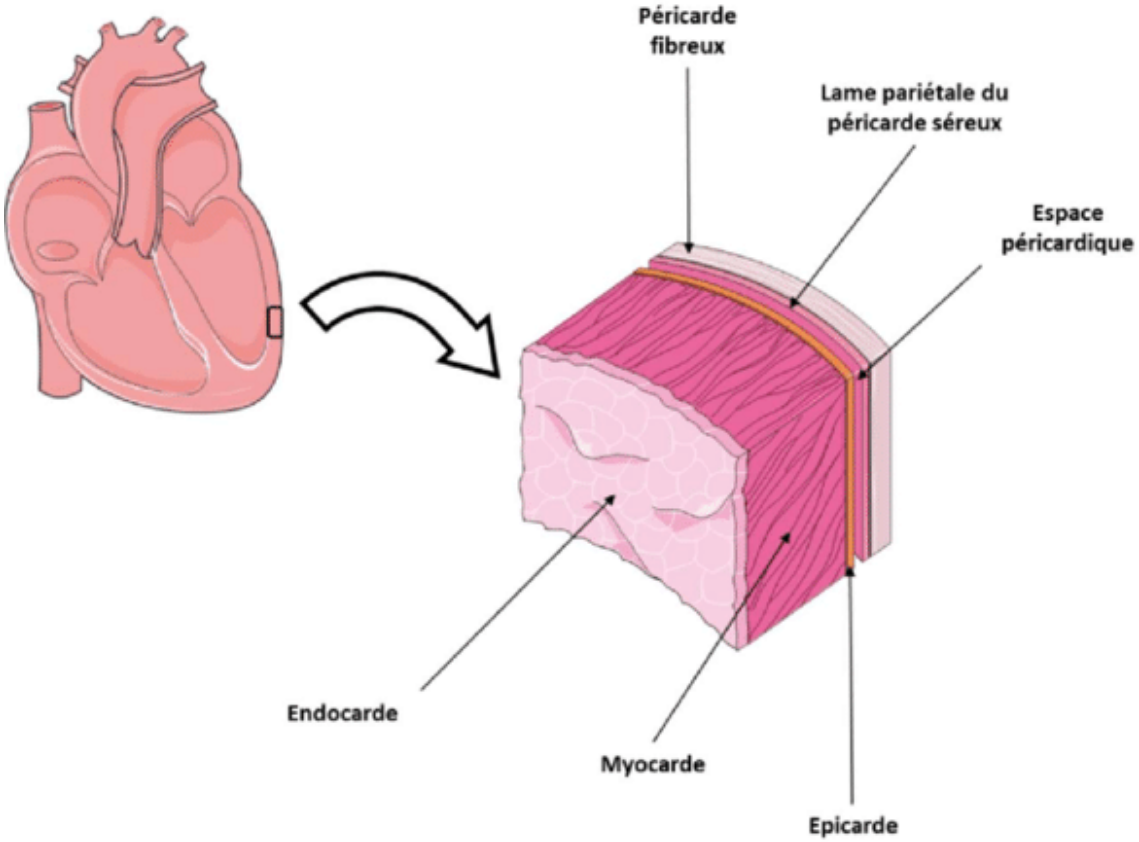


FIGURE 2.2 – Représentation des couches de la paroi cardiaque. Image issue de Fornasier-Santos 2018

Le myocarde est constitué de cellules musculaires ainsi que de cellules pacemaker.

- Les cellules pacemaker permettent de générer le courant électrique responsable de la contraction du cœur. Elles sont contrôlées par le système nerveux autonome, c'est-à-dire qu'elles émettent des impulsions électriques de façon régulière sans contrôle volontaire. Ces cellules s'activent entre 60 et 100 fois par minute, initiant un processus nommé "potentiel d'action cardiaque". Elles sont situées dans une zone appelée nœud sinusal qui se trouve dans la partie supérieure de l'oreillette droite. Les cellules pacemaker sont bien plus petites que les cellules musculaires et représentent seulement 1% de l'ensemble des cellules du cœur.
- Les cellules musculaires (cardiomyocytes) représentent 99% des cellules des ventricules et des oreillettes et sont liées entre elles par des disques intercalaires. Ces disques permettent aux cellules de réagir rapidement et de façon coordonnée lorsqu'une stimulation électrique en provenance des cellules pacemaker les traverse.

Lorsque le cœur se contracte, les cellules pacemaker transmettent des ions vers les cellules musculaires à l'aide de jonctions communicantes qui relient les cellules entre elles. Le courant électrique part du nœud sinusal en direction du nœud atrioventriculaire et continue le long de la paroi cardiaque située entre les deux ventricules. Il atteint ensuite l'apex du cœur avant de remonter vers les fibres de Purkinje (Figure 2.3).

L'activité électrique du cœur se mesure à l'aide d'un électrocardiogramme (ECG). La phase de dépolarisation des oreillettes correspond à l'onde P sur l'ECG. La phase de dépolarisation des ventricules correspond au segment QRS. L'onde T représente la phase de repolarisation des ventricules.

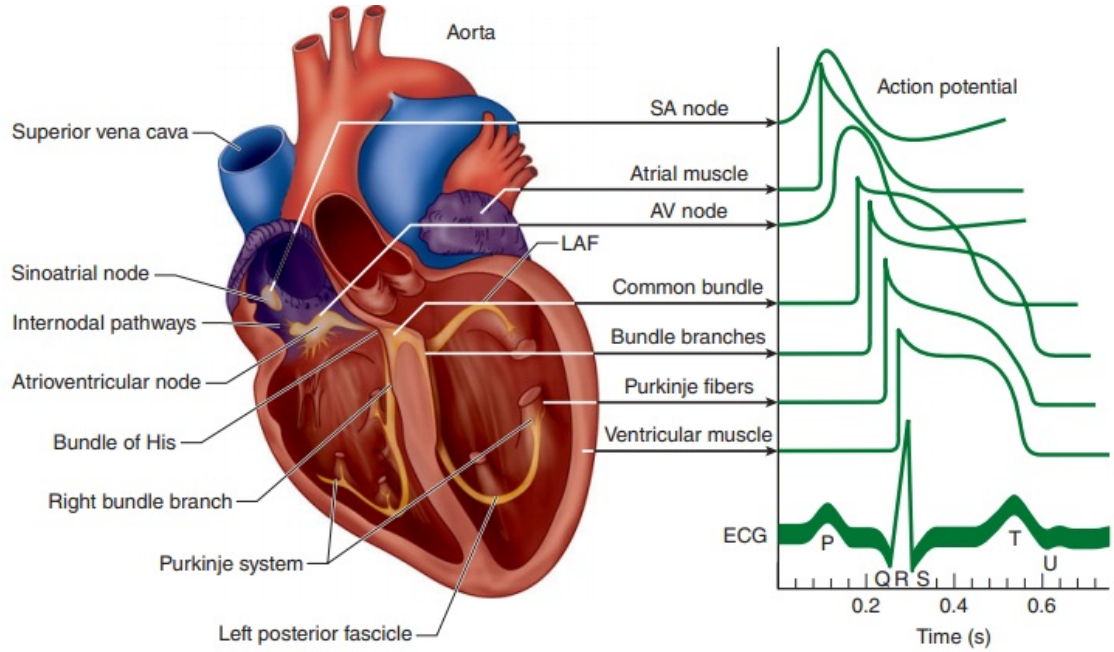


FIGURE 2.3 – Le potentiel d'action cardiaque [2].

2.1.2 Maladies cardiovasculaires en lien avec le myocarde

Cette section introduit les principales affections myocardiques. Ces pathologies cardiaques se manifestent par des douleurs thoraciques, des difficultés à respirer, de la fatigue, des syncopes et peuvent entrainer des insuffisances cardiaques, des arythmies ou des arrêts cardiorespiratoires.

La fibrillation atriale (FA)

La FA se manifeste par un rythme de contraction de l'atrium (oreillette) trop rapide et irrégulier, se traduisant par une réduction du débit cardiaque. Les organes du corps ne sont alors plus alimentés suffisamment en oxygène. La FA a pour conséquence une dilatation de l'oreillette entrainant la fibrose des tissus myocardiques. La sévérité de la fibrose augmente avec la durée de la FA. L'accumulation de tissus conjonctifs du fait de la cicatrisation peut altérer le système de conduction des

2. https://www.brainkart.com/article/Cardiac-Action-Potentials_26936/

impulsions électriques dans le myocarde. La dilatation de l'oreillette et donc la fibrillation atriale est favorisée par l'hypertension ou la présence de maladies des valves cardiaques. Lors de la FA, la conduction électrique est ralentie et la période réfractaire (temps séparant deux impulsions électriques) plus courte. Par conséquent, un ou plusieurs courant électriques circulaires réentrants se forment au niveau de l'oreillette, le plus souvent proche des veines pulmonaires. Ces courants électriques interfèrent avec l'impulsion électrique principale trouvant son origine dans le nœud sinusal. La formation de ces courants réentrants (rotor) s'explique par la séparation du front de l'onde électrique provenant du nœud sinusal en sous-ondes plus petites au contact d'obstacles (on parle de "vortex shedding" en anglais). Dans certaines conditions, ces sous-ondes peuvent se reformer et suivre un mouvement rotatif autour d'un point central (Waks et Josephson 2014). Il est important de noter que le nombre accru d'impulsions électriques dans le cas de la FA ne se traduit pas par un nombre équivalent de contraction cardiaque. En effet, le nœud atrioventriculaire dispose d'une conduction réduite de sorte que toutes les ondes électriques en provenance des oreillettes n'entrainent pas une contraction des ventricules. Cela différentie la FA du terme plus général de tachycardie qui peut faire référence à une arythmie au niveau des oreillettes ou des ventricules.

La FA peut se diagnostiquer à l'aide d'un ECG et se caractérise par l'absence d'onde P ainsi que des complexes QRS irrégulièrement espacés. L'échocardiographie permet également de plus facilement détecter une dilatation de l'oreillette qui se manifeste lors de la FA.
La FA accroit le risque d'infarctus du myocarde ou de thrombose. Par conséquent, des anticoagulants sont souvent prescris aux patients. Si ces anticoagulants sont contre-indiqués ou inefficaces, il est possible de réaliser une opération consistant à fermer l'auricule gauche. Cette zone se situe au sein de l'oreillette et prend la forme d'une cavité. Elle n'assure pas de fonction primordiale mais 90% des thromboses du cœur chez les patients souffrant de FA sont liées à des thrombus se formant dans cette zone (Blackshear et Odell 1996). Par conséquent il est intéressant d'obstruer cette zone de façon à éviter qu'un thrombus ne migre vers un vaisseau sanguin.
Un traitement médicamenteux peut également être prescris de manière à réduire la fréquence cardiaque, notamment en réduisant le courant électrique passant par le nœud atrioventriculaire (Bêta-bloquants ou inhibiteur calcique). Lorsque les traitements médicamenteux sont inefficaces, il est possible de pratiquer une ablation par cathéter de façon à cibler avec précision le tissu cardiaque responsable de l'arythmie (soit par onde radiofréquence, soit par cryoablation) situé le plus souvent au niveau des veines pulmonaires. La chirurgie de Cox ("Cox Maze procedure") (Prasad et al. 2003) se pratique également comme alternative à l'ablation par cathéter. Cette méthode consiste à réaliser des sutures au sein de l'oreillette de façon à guider le signal électrique en provenance du noeud sinusal vers le noeud atrioventriculaire. La version la plus récente de cette méthode (Cox Maze 4 (Damiano et al. 2011)) utilise les ondes radiofréquences plutôt que des sutures. Les méthodes dites "minimaze" reposent sur la méthode de Cox mais limitent le nombre de sutures et/ou les zones concernées de façon à rendre l'opération moins invasive.
La consommation d'alcool, le tabagisme, l'obésité, le diabète, l'absence d'activité physique régulière ainsi que l'apnée du sommeil ou la présence de cardiopathie congénitales favorisent l'apparition de la FA.

En 2014, la FA concernait 2 à 3% de la population européenne et nord-américaine (Zoni-Berisso et al. 2014). 33 millions de personnes étaient affectées dans le monde en 2020 (Chung, Eckhardt et al. 2020).

L'infarctus du myocarde

L'infarctus du myocarde, plus communément connu sous le nom de "crise cardiaque", se traduit par la baisse de l'apport sanguin ou l'arrêt complet de cet apport dans une ou plusieurs des artères coronaires qui irriguent le myocarde (ischémie). Ce défaut de perfusion entraine la mort (nécrose) des cellules cardiaques qui ne sont plus alimentées correctement en oxygène et nutriment. De ce fait, le cœur n'est plus capable de pomper efficacement le sang vers les organes du corps.
Le plus souvent, l'infarctus du myocarde est la conséquence de l'athérosclérose (Reed, Rossi et Cannon 2017 ; Authors/Task Force Members et al. 2008). Les plaques d'athérome, représentées en Figure 2.4, sont des dépôts contenant de la graisse, du cholestérol ou encore du calcium qui se forment sur plusieurs années au sein d'une artère. Avec l'âge, et à mesure que la plaque d'athérome grossit, il est possible que l'endothélium, la paroi interne de l'artère séparant la plaque d'athérome de la partie creuse de l'artère (lumière ou lumen), se rompe. Cette rupture entraine l'arrivée rapide de thrombocytes chargés de colmater la brèche pour assurer la coagulation sanguine. À l'issue de la coagulation sanguine, un thrombus se forme. Ce thrombus va alors faire obstacle à la circulation du sang en réduisant l'espace au sein de la lumière de l'artère (sténose). Lorsque l'artère est complètement obstruée, les tissus en aval ne sont plus alimentés en sang entrainant des lésions irréversibles (*Davidson's principles and practice of medicine. - NLM Catalog - NCBI* 2024).
L'infarctus du myocarde peut être diagnostiqué à l'aide de tests sanguins, d'une coronarographie ou d'un ECG. En effet, selon que l'on observe une élévation du segment ST sur l'ECG ou non, l'infarctus du myocarde sera catégorisé en STEMI ou NSTEMI. Le segment ST est situé entre le segment QRS correspondant à la dépolarisation des ventricules et l'onde T qui correspond à leur repolarisation.
Les facteurs de risque établis pour l'infarctus du myocarde sont l'obésité, le manque d'exercice physique, le tabagisme, l'hypertension, un haut taux de cholestérol dans le sang ou encore la consommation excessive d'alcool et la mauvaise alimentation. Des facteurs génétiques jouent également un rôle important dans son apparition (Perk et al. 2012).

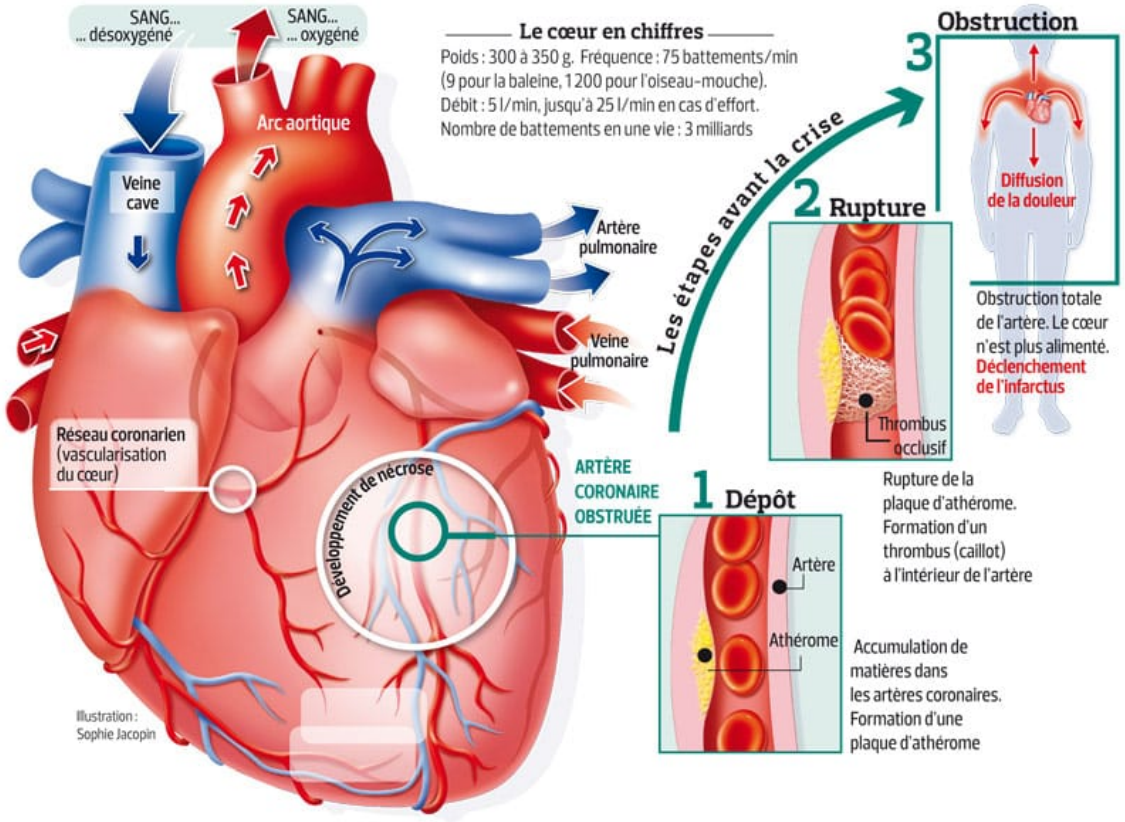


FIGURE 2.4 – L'infarctus du myocarde[3].

La cardiomyopathie dilatée (CMD)

La CMD correspond à une augmentation de la taille du ventricule gauche accompagnée d'un amincissement et d'une perte d'élasticité de la paroi myocardique. Les ventricules ont alors une forme plus arrondie et moins allongée. Du fait de ces altérations, le sang n'est plus pompé efficacement vers les organes (Figure 2.5). Cette cardiomyopathie est également associée à une baisse de la fraction d'éjection (Elliott et al. 2008).
Une personne atteinte de cardiomyopathie dilatée va pouvoir, dans un premier temps, compenser la baisse de la fraction d'éjection du fait de la loi de Starling. Cette loi stipule que l'augmentation du volume télédiastolique s'accompagne d'un plus grand étirement du sarcomère se trouvant dans les cardiomyocytes, induisant une plus grande force de contraction du cœur et donc un volume d'éjection plus important. Par conséquent, les patients atteints de cardiomyopathie dilatée peuvent être asymptomatique, au moins dans un premier temps, du fait de ce mécanisme de compensation. À mesure que la pathologie s'aggrave, ce mécanisme ne sera plus suffisant pour compenser la baisse de la fraction d'éjection (Lilly 2012).
La cardiomyopathie dilatée peut être diagnostiquée à l'aide d'un ECG. En effet, lorsqu'un patient présente un bloc de branche ainsi qu'une déviation axiale droite, il est probable qu'il soit atteint de cardiomyopathie dilatée (Nikolic et Marriott 1985). Ces

3. https://institut.amelis-services.com/sante/autres/infarctus-du-myocarde-definition-symptomes-c

modifications de l'ECG correspondent respectivement à un retard dans la contraction du ventricule gauche (courant électrique lent ou inexistant dans le faisceau de His) et à une déviation du sens de propagation du courant électrique dans le cœur. Par ailleurs, le recours à un cathétérisme cardiaque ou à l'angiographie permet d'exclure l'existence d'une maladie coronarienne. Les tests génétiques jouent également un rôle important dans le diagnostic de la cardiomyopathie dilatée puisqu'il a été montré que 25 à 35% des patients atteints ont des antécédents familiaux (Kumar et al. 2007). L'IRM cardiaque permet également de faciliter le diagnostic de cette pathologie (Dent 2019).

La cardiomyopathie hypertrophique (CMH)

La CMH correspond à un épaississement des tissus musculaires du cœur (Figure 2.5). Cette pathologie touche principalement le ventricule gauche. L'accroissement de la taille du myocarde réduit son élasticité et sa capacité à contracter le cœur. Par ailleurs, la conduction électrique au sein du myocarde se dégrade, en particulier pour le faisceau de His et la fibre de Purkinje, zones jouant un rôle essentiel pour la dépolarisation des ventricules. L'élargissement du myocarde peut conduire le muscle cardiaque à faire obstacle à l'éjection du sang vers les organes ; on parle alors de cardiomyopathie obstructive. Cette obstruction est liée à la fois à l'élargissement du septum (paroi séparant les deux ventricules), mais aussi au déplacement de la valve mitrale vers le septum. Ce déplacement est lié à l'effet Venturi qui stipule que la pression diminue lorsque la vitesse d'écoulement s'accélère dans des espaces étroits. Du fait de cette diminution de la pression au niveau de la voie d'éjection du sang du ventricule gauche, la valve mitrale est attirée vers le septum (effet du gradient de pression) (Basit, Brito et Sharma 2024).
Par conséquent, il est possible de diagnostiquer la CMH à l'aide d'un cathétérisme cardiaque, en mesurant la différence de pression entre le ventricule gauche et la partie ascendante de l'aorte. L'échocardiographie est généralement utilisée pour diagnostiquer cette pathologie avec un taux de succès supérieur à 80% (Parato et al. 2016). Dans le cas où l'échocardiographie ne permet pas de détecter la maladie, on a souvent recours à l'IRM cardiaque. Il est par exemple difficile de déceler la CMH pour les enfants ayant moins de 13 ans avec l'échocardiographie (Maron 2002). En revanche, en utilisant l'IRM, des chercheurs ont montré qu'il était possible d'identifier des anomalies dans les cellules musculaires au niveau du septum chez des patients asymptomatiques. Ces anomalies seraient des marqueurs précoces de l'apparition de la CMH (Germans et al. 2006). Plus généralement, lors de la présence d'une CMH, l'IRM permet dans 60 à 70% des cas de montrer un épaississement de plus de 15mm de la partie basse du septum interventriculaire (Amano et al. 2018).
La CMH résulte principalement de prédispositions génétiques. La pathologie survient à la suite de l'altération de certains gènes responsables du développement des protéines au sein du sarcomère. Ainsi, entre 50 et 60% des personnes avec une forte suspicion de CMH ont une mutation sur au moins l'un des 9 gènes du sarcomère. De plus, la CMH étant un caractère génétique autosomique dominant, si l'un des parents présente cette mutation génétique, l'enfant aura 50% de chances d'hériter de la mutation (Cirino et Ho 1993). Par conséquent, au-delà des techniques d'imagerie, le diagnostic de cette pathologie s'appuie parfois sur des tests génétiques ainsi que sur la consultation des antécédents familiaux (Gersh et al. 2011).

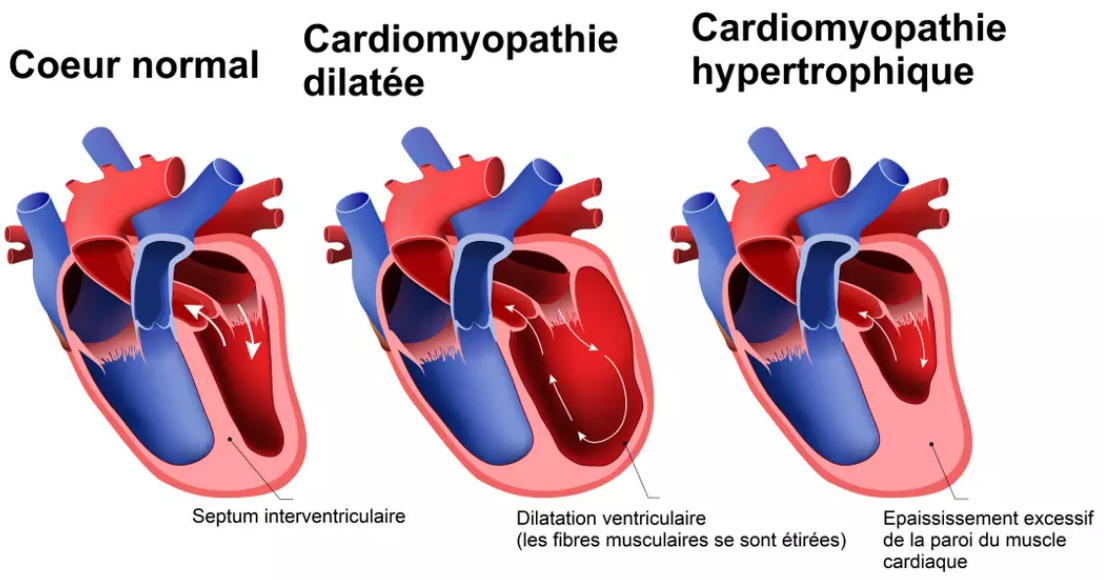
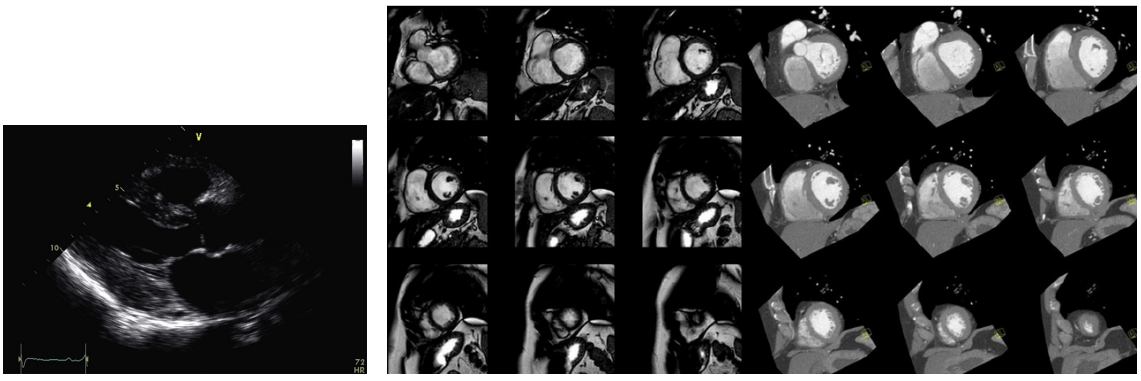


FIGURE 2.5 – Cardiomyopathies dilatée et hypertrophique [4].

2.1.3 Imagerie cardiaque

Modalités d'imagerie cardiaque

Avant de revenir plus en détails sur le fonctionnement de l'IRM, il convient d'évoquer les principales différences avec les deux autres principales modalités d'imagerie cardiaque que sont l'échocardiographie et le scanner (Computed Tomography, CT). Des exemples de ces 3 modalités d'imagerie sont présentés Figure 2.6



(a) (b)

FIGURE 2.6 – (a) Exemple d'échocardiographie (Fitzgerald, Bashford et Scalia 2017). (b) Images petit-axe d'un même patient à fin systole avec IRM (à gauche) et scan CT (à droite) (Martini et al. 2010)

4. https://sante.journaldesfemmes.fr/fiches-maladies/2733453-cardiomyopathie-symptome-esperance-d

- L'échocardiographie fonctionne à l'aide d'une sonde que l'on place sur la poitrine du patient et qui émet des ondes ultrasonores à travers le corps. Lorsque les ondes entrent en contact avec certaines structures, elles sont renvoyées vers la sonde qui enregistre un écho. Il s'agit d'une modalité d'imagerie non-invasive, indolore et non ionisante. L'échocardiographie permet de produire des images dynamiques en temps réel mais reste généralement 2D. De plus, le positionnement du cœur dans la cage thoracique limite les positions de la sonde permettant de l'imager. Elle est particulièrement utile pour mesurer des paramètres de la fonction cardiaque (comme les volumes ou la déformation myocardique). Par ailleurs, il est également possible de mesurer la vitesse du sang dans les cavités cardiaques grâce à l'effet Doppler. En effet, la fréquence des ondes émises par la sonde est modifiée par les objets en mouvement comme les globules rouges du sang. Cette mesure permet ainsi d'évaluer la présence de fuite ou de sténose au niveau des valves mitrale et aortique. Enfin, l'échocardiographie, de par son caractère dynamique, permet aussi de diagnostiquer des infarctus du myocarde, des maladies liées aux valves cardiaques, ou un dysfonctionnement dans le pompage du sang.
- Le scanner CT fonctionne à l'aide d'un tube à rayons X qui effectue des rotations autour du patient et dont les rayons émis sont captés par un détecteur situé à l'opposé. Le patient est positionné entre l'émetteur et le récepteur et les rayons X sont atténués par les différents tissus du corps en fonction de leur densité. Le sinogramme obtenu est ensuite traité informatiquement pour reconstruire un volume tomographique 3D de haute résolution. Bien que l'acquisition d'un volume CT soit rapide (la positionnant comme une modalité de choix en cas d'urgence), son caractère ionisant la restreint à de l'imagerie statique. Les travaux récents de réduction de dose, ont permis l'émergence de séquence dynamique mais au prix d'une moins bonne résolution et d'un moins bon contraste. Le taux de contraste des images CT, qui peut être amélioré en injectant un produit de contraste, est plus élevé pour les structures denses comme les os mais plus faible pour les tissus mous comme la graisse qui absorbent moins les rayons X. La tomodensitométrie est fréquemment utilisée pour détecter, diagnostiquer ou effectuer un suivi des maladies coronaires. On peut en effet détecter des plaques de calcium responsable d'ischémies, ou analyser les artères coronaires en injectant un produit de contraste (angiographie coronaire CT).
- L'Imagerie par Résonance Magnétique (IRM) est une technique d'imagerie non invasive, indolore et non ionisante, capable de générer des images avec une bonne résolution, un bon rapport signal sur bruit et un taux de contraste élevé, notamment pour les tissus mous. Cela en fait une méthode particulièrement adaptée pour l'imagerie cardiaque et en particulier pour générer des séquences d'images temporellement résolues. Nous revenons en détail sur le fonctionnement de l'IRM dans la section suivante.

Fonctionnement de l'imagerie IRM

Les protons des atomes d'hydrogènes possèdent un mouvement intrinsèque appelé "spin". Le spin est souvent assimilé au moment cinétique des particules quantiques et possède donc une direction et une magnitude. Sa direction est une valeur vectorielle correspondant à l'axe de rotation tandis que sa magnitude correspond à sa vitesse de rotation. S'il n'est pas possible de mesurer la direction d'un spin individuel (principe d'incertitude de Heisenberg), il est en revanche possible de mesurer sa direction selon un axe particulier (x, y ou z) ainsi que sa magnitude. À l'état naturel, le spin de ces protons est désynchronisé et la somme de leurs moments magnétiques de spin est nulle. L'Imagerie par Résonance magnétique (IRM) génère un champ magnétique B_0 de façon à aligner les spins des protons soit dans le même sens que B_0 soit dans le sens opposé (polarisation). De ce fait, l'IRM fonctionne bien pour la matière composée de nombreux atomes d'hydrogène, comme c'est le cas du corps humain (60% d'eau). En revanche, l'IRM est inefficace pour l'étude des parties osseuses, pauvre en molécules d'eau. La fréquence de rotation des protons (précession de Larmor) ω_L est proportionnelle à l'intensité du champ magnétique B_0 :

$$\omega_L = -\gamma B_0, \tag{2.1}$$

Après application de B_0 dans le sens de l'axe z, les protons précessent autour de l'axe z (la somme des moments magnétiques de spin de l'ensemble des protons est différente de 0). Pour obtenir des images, une pulsation "radiofréquence" va être appliquée dans une direction perpendiculaire à B_0. À la suite de cette impulsion, les protons vont s'orienter de façon perpendiculaire à B_0 et précesser de façon synchronisée. Une fois l'impulsion émise, les protons vont progressivement se réorienter vers l'axe z, le sens d'application de B_0. L'orientation va gagner en composante longitudinale et perdre en composante transversale. La mesure des durées de récupération des composantes longitudinale ou transversale permettent d'obtenir des images IRM. Le processus est décrit Figure 2.7. Le temps de relaxation des différents protons d'hydrogène dans le corps n'est pas le même pour chaque organe conduisant à des images avec un fort contraste. Durant la phase de diminution de la composante transversale, on observe une perte progressive de cohérence dans les spins des protons, laquelle contribue à un affaiblissement progressif du signal (FID). Le temps entre le moment de l'émission de l'impulsion radiofréquence et le moment de récupération du signal est nommé "Temps d'Echo" (TE) tandis que celui entre l'émission de deux impulsions radiofréquence est nommé "Temps de Répétition" (TR). En fonction de la durée de TE et TR, il est possible de modifier le contraste de l'image en privilégiant la relaxation longitudinale (appelée T1) ou transversale (appelée T2). Plus TE et TR seront courts et plus l'image obtenue sera pondérée en T1. À l'inverse, plus ils seront longs et plus la pondération T2 sera importante. Lorsque le contraste de l'image est principalement déterminé par le temps de relaxation longitudinale des tissus, on dit que l'image est pondérée en T1 ("T1-weighted image"). Dans le cas où c'est le temps de relaxation transversale qui détermine le contraste de l'image, on parle d'image pondérée en T2 ("T2-weighted image").

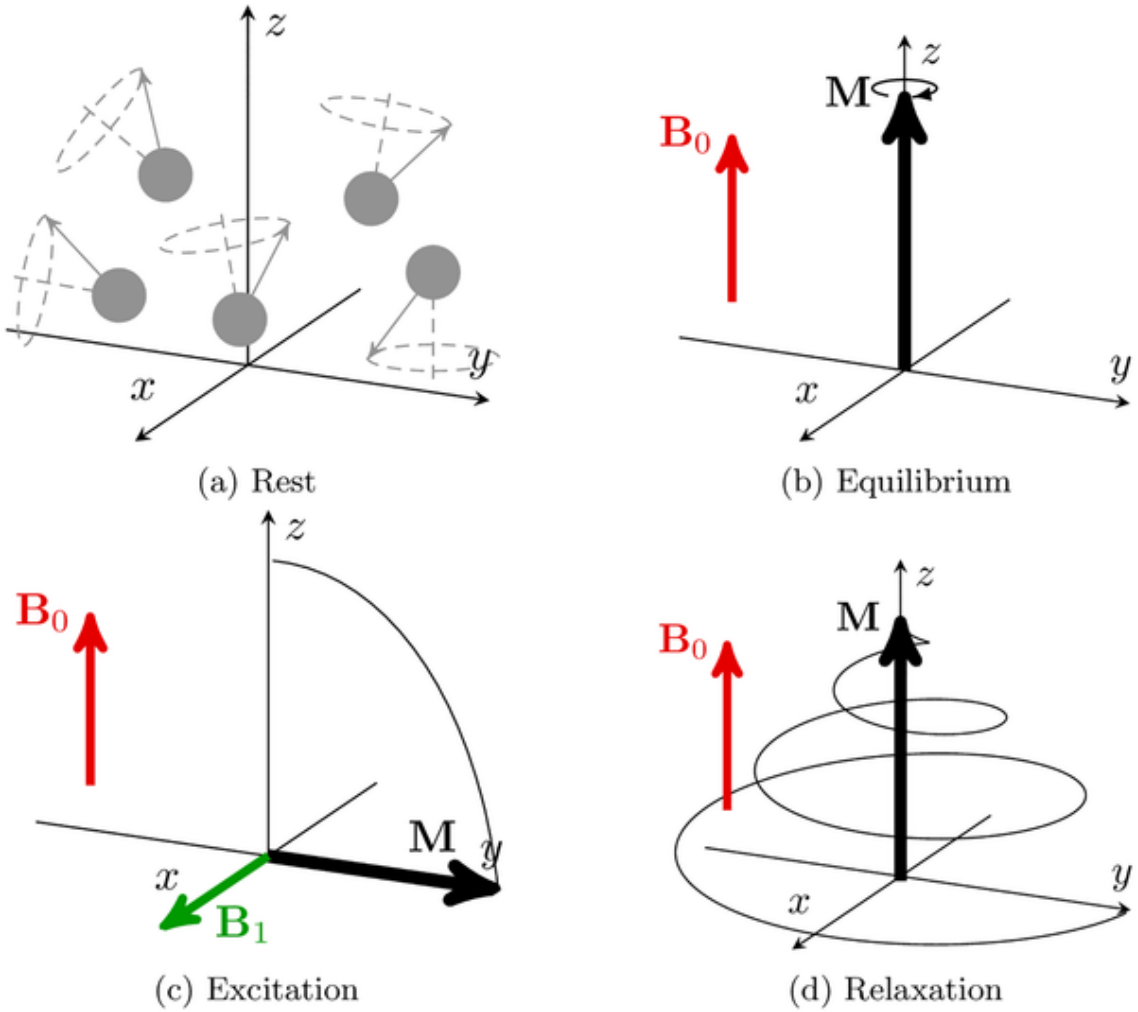


FIGURE 2.7 – (a) Au repos, les protons d'hydrogène ont un moment magnétique dont l'orientation est aléatoire. (b) L'application d'un champ magnétique B_0 permet d'aligner les moments magnétiques des protons d'hydrogène de façon parallèle ou antiparallèle à ce champ. La somme des moments magnétiques des protons est appelée aimantation résultante nette ("net magnetization") et est notée M. Elle est en générale positive car il y a davantage de protons alignés dans le sens de B_0 que dans le sens opposé. (c) L'impulsion d'une onde radiofréquence à la fréquence de Larmor modifie l'orientation du vecteur M (ici à 90^o par rapport à B_0). (d) M se réaligne ensuite avec B_0 en récupérant progressivement sa composante longitudinale (relaxation T1) et en perdant celle transversale (relaxation T2). Les bobines de l'IRM récupèrent le signal émis par les protons lors de leur relaxation. Image issue de Puiseux et al. 2021.

Imagerie IRM et orientation dans l'espace

Pour obtenir des images selon différentes orientations (petit-axe, grand axe), il est nécessaire d'obtenir un signal variant dans les 3 directions de l'espace. Autrement dit, il est nécessaire d'avoir des protons dont la fréquence de précession varie spatialement. Du fait de l'homogénéité du champ B_0 selon l'axe z, cela n'est pas directement possible car la fréquence de Larmor est la même pour tous les protons. En ajoutant un gradient magnétique variant linéairement selon l'axe z au champ B_0,

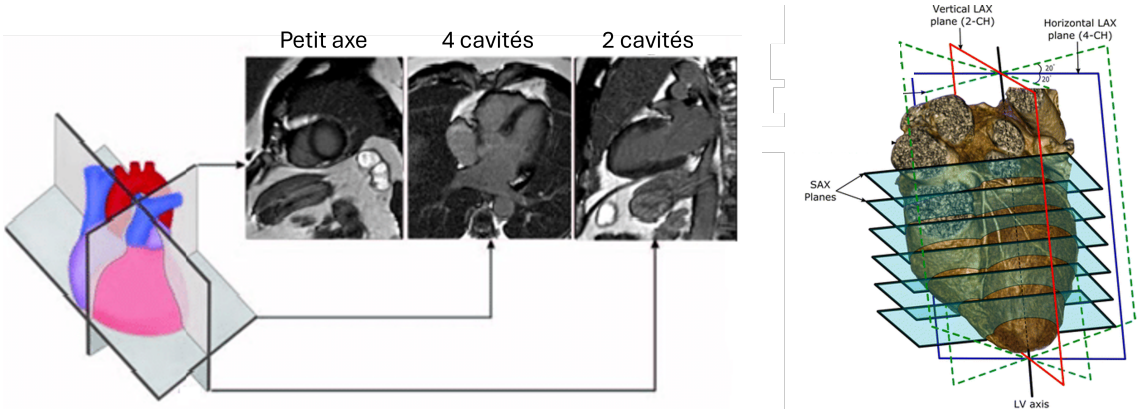
la fréquence de Larmor des protons varie également linéairement selon cet axe. En émettant une onde radiofréquence dont la fréquence correspond à la fréquence de Larmor d'une plage particulière de protons, on est ainsi capable de sélectionner une coupe du volume. Pour obtenir des voxels ayant des valeurs différentes selon l'axe des x et des y, on utilise également un gradient magnétique variant linéairement dans ces deux directions. On utilise un gradient magnétique avec une fréquence variant selon l'un des deux axes pour distinguer l'intensité des voxels selon cet axe, c'est l'étape de "frequency encoding". Pour chaque voxel x selon l'axe d'encodage des fréquences, la fréquence de précession $f(x)$ devient la somme de la fréquence du gradient G_f et du champ magnétique de départ B_0, pondéré par la constante gyromagnétique γ : $f(x) = \gamma(B_0 + xG_f)$. Pour le second axe, on utilise un gradient magnétique faisant varier la phase des signaux, c'est l'étape de "phase encoding". Au final, chaque pixel au sein d'une slice sélectionnée selon l'axe z dispose d'une fréquence et d'une phase différente. Ces informations de phase et de fréquence sont temporairement stockées dans une matrice de même taille que l'image produite en sortie, il s'agit du "k-space" (ou domaine de Fourier). A la fin de l'acquisition, on applique la transformée de Fourier inverse pour reconstruire l'image spatiale et récupérer l'intensité des voxels.

De cette façon, l'IRM permet d'obtenir des images selon des orientations différentes : axiale, sagittale et coronale. Pour l'imagerie cardiaque, on distingue les images petit-axe des images grand-axe.

- Les images grand-axe sont acquises de façon parallèle au grand axe qui va de la base à l'apex du coeur. On distingue les images grand-axe 2 cavités des images 3 cavités et 4 cavités. Les images grand-axe 2 cavités présentent le ventricule gauche et l'oreillette gauche. Les images grand-axe 3 cavités contiennent également la valve aortique ainsi que l'aorte ascendante. Les images grand axe 4 cavités permettent de distinguer les ventricules gauche et droit ainsi que les oreillettes gauche et droite. L'orientation grand-axe est privilégiée pour évaluer la taille, la forme et la déformation des oreillettes, ce qui la rend particulièrement utile pour diagnostiquer, par exemple, la fibrillation atriale.
- Les images petit-axe sont acquises de façon perpendiculaire au grand axe. Elles présentent une vue transversale des ventricules gauche et droit ainsi que de la paroi cardiaque. Par conséquent, cette orientation est privilégiée pour évaluer l'épaisseur du myocarde. En effet, la paroi cardiaque peut être divisée en segments dont la taille et la déformation peuvent être analysées pour diagnostiquer des infarctus du myocarde ou des cardiomyopathies hypertrophiques. L'orientation petit-axe ne permet pas de distinguer les oreillettes des ventricules.

Des exemples de coupes petit-axe et grand-axe ainsi que le plan de coupe permettant de les obtenir sont présentés Figure 2.8a. Cette thèse porte uniquement sur l'étude des images petit-axe de façon à analyser la déformation de la paroi cardiaque. Plusieurs de ces coupes sont alignées selon l'axe z (en profondeur) pour former un volume 3D (Figure 2.8b) allant de la base à l'apex du coeur. De par la nature dynamique de la méthode SSFP, on dispose donc finalement d'un volume 3D pour

chaque phase du cycle cardiaque, conférant à ces données une dimension spatiale et temporelle (3D+t).



(a) (b)

FIGURE 2.8 – (a) Exemples d'images petit-axe et grand-axe ainsi que de leur plan de coupe respectif (Kruk et al. 2017). (b) Plusieurs coupes 2D petit-axe sont "empilées" le long du grand axe pour former un volume 3D (El-Rewaidy et Fahmy 2016).

Imagerie IRM Ciné

L'imagerie IRM Ciné est une série d'acquisitions d'images IRM formant une vidéo permettant d'observer le cœur à différents instants du cycle cardiaque. Ces images sont générées à l'aide de la méthode SSFP (Steady-state free precession imaging). Cette technique repose sur la méthode de l'écho de gradient qui utilise un gradient magnétique pour s'assurer que les spins sont en phase (McRobbie et al. 2017). Les séquences SSFP sont rendues possibles par des séries d'impulsions radiofréquences avec un faible TR ($TR < TE$) de sorte que le début de l'écho se confond avec la fin de la FID (Figure 2.9). Cela permet le maintien d'une magnétisation transversale résiduelle constante entre l'émission des radiofréquences (Carr 1958). En imagerie cardiaque, ces séquences sont généralement réalisées sous apnée du patient (10-15s) et un cycle cardiaque « moyen » est reconstruit (40-50 images par cycle).

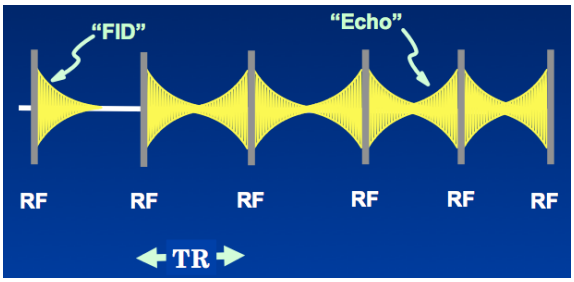


FIGURE 2.9 – Les séquences SSFP se caractérisent par un TR faible et un écho arrivant avant la dégradation totale du signal en jaune. Le signal émis par les protons d'hydrogène lors de la perte de la composante longitudinale et transversale après chaque impulsion radiofréquence est detecté par les bobines de l'IRM [5].

5. https://mriquestions.com/what-is-ssfp.html

Les séquences ciné du cœur sont généralement utilisées pour évaluer la fonction cardiaque grâce à l'information de mouvement qu'elles fournissent. Ainsi, ces séquences permettent d'analyser la façon dont le cœur se contracte et se relâche pour identifier des anomalies liées au mouvement de la paroi cardiaque, à la fraction d'éjection ou au volume des ventricules (Schulz-Menger et al. 2013 ; Pennell et al. 2004 ; Grothues et al. 2002). L'IRM ciné a également été utilisée pour aider au diagnostic de pathologies liées aux valves du cœur (Hundley et al. 1995 ; Cawley, Maki et Otto 2009). Des travaux ont été conduits pour mesurer le flux sanguin et faciliter le diagnostic de pathologies liées à l'hémodynamique cardiovasculaire et utilisent généralement l'IRM à contraste de phase sur des séquences ciné (Lotz et al. 2002 ; Gatehouse et al. 2010 ; Tang, Blatter et Parker 1993). L'IRM ciné fournit également des informations pour préparer au mieux les interventions cardiaques notamment le cathétérisme (Amin, Campbell-Washburn et Ratnayaka 2022 ; Razavi et al. 2003) pour les malformations congénitales. Plus récemment, les techniques utilisant le Deep Learning ont gagné en popularité et ont montré leur efficacité pour classer certaines pathologies cardiaques ou extraire des paramètres quantitatifs concernant la fonction cardiaque (Papandrianos et al. 2022 ; Dong, Luo et al. 2018 ; Bai et al. 2018).

2.1.4 Biomarqueurs d'imagerie

L'étude de l'imagerie cardiaque a pour but d'extraire des mesures quantitatives à même de renseigner sur certaines fonctions physiologiques des patients. Ces mesures, appelées biomarqueurs, permettent de faciliter le diagnostic et le pronostic des pathologies cardiaques, de surveiller leur évolution ou encore d'adapter le traitement suivi par le patient. Nous présentons ci-dessous les biomarqueurs qui sont estimés dans les travaux de cette thèse.

Volumes des ventricules à fin diastole et fin systole

Les volumes des ventricules en télé-diastole (End Diastolic Volume, EDV) et télé-systole (End Systolic Volume, ESV) renseignent sur la façon dont le sang se déverse et est expulsé de ces cavités. Ils constituent donc une bonne mesure de la fonction cardiaque. D'après une méta-analyse (Kawel-Boehm et al. 2020), les volumes du ventricule gauche en télé-diastole et télé-systole pour un sujet adulte se situent autour de 155 ± 30mL et 55 ± 15mL respectivement pour les hommes, contre 123 ± 22mL et 43 ± 11mL pour les femmes. Pour le ventricule droit, les valeurs sont de 166 ± 39mL et 73 ± 22mL pour les hommes, contre 122 ± 27mL et 50 ± 15mL pour les femmes.

Le volume d'éjection systolique

Le volume d'éjection systolique (Stroke Volume, SV) est la différence entre le volume télé-diastolique et télé-systolique (SV = EDV - ESV). Il est un bon indicateur du débit cardiaque. Pour le ventricule gauche, les valeurs sont autour de 103 ± 21mL pour les hommes, contre 83 ± 16mL pour les femmes. Pour le ventricule droit, le volume d'éjection se situe autour de 95 ± 26mL pour les hommes contre 74 ± 18mL pour les femmes (Kawel-Boehm et al. 2020).

La fraction d'éjection

La fraction d'éjection représente le pourcentage du sang présent dans une cavité qui est éjecté à chaque battement. Pour les ventricules, la fraction d'éjection est une mesure de l'efficacité du pompage du sang vers le système circulatoire (ventricule gauche) et vers les poumons (ventricule droit). En particulier, elle est un bon indicateur de l'insuffisance cardiaque (réduction de la quantité de sang pompé vers les organes). Elle se calcule comme le rapport du volume d'éjection sur le volume du ventricule en télé-diastole (EF = SV/EDV). Pour le ventricule gauche, la fraction d'éjection se situe autour de $64 \pm 8\%$ pour les hommes contre $66 \pm 7\%$ pour les femmes. Pour le ventricule droit ces valeurs sont autour de $57 \pm 8\%$ pour les hommes et $60 \pm 7\%$ pour les femmes (Kawel-Boehm et al. 2020).

La masse du myocarde

La masse du myocarde est estimée en multipliant le volume du myocarde (en mL) par la densité de celui-ci (estimée à 1.053g/ml (Vinnakota et Bassingthwaighte 2004)). La masse du myocarde donne des informations importantes pour détecter, entre autres, la cardiomyopathie hypertrophique. La masse du myocarde se situe autour de 121 ± 28g pour les hommes et 83 ± 21g pour les femmes (Kawel-Boehm et al. 2020).

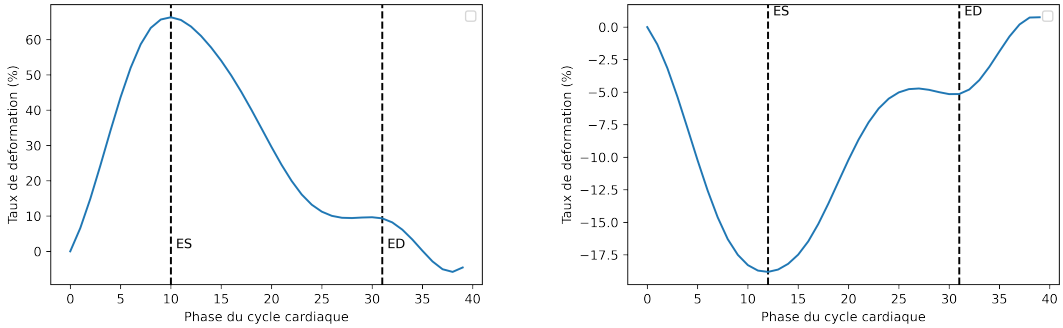
La déformation myocardique

La déformation du myocarde (ou strain) mesure la variation en épaisseur (strain radial), en circonférence ou en longueur du myocarde au cours du cycle cardiaque. Pour les images petit-axe, seules les déformations radiales et circonférentielles sont accessibles. Cette déformation est calculée à chaque instant t du cycle cardiaque par

$$\epsilon_t = \frac{L_t - L_0}{L_0} \tag{2.2}$$

où L est la mesure (circonférence, longueur, épaisseur), et t_0 un instant de référence (généralement la télé-diastole). Une fois le strain calculé sur l'ensemble de la séquence, on obtient des courbes de strain dont des exemples sont présentés Figure 2.10.

La déformation donne des informations utiles pour diagnostiquer les ischémies, les infarctus du myocarde ou les blocs de branche car ces conditions se manifestent par une altération de la force de contraction du myocarde et donc sa déformation notamment en télé-systole. Les pourcentages de déformation globale radiale et circonférentielle du myocarde en télé-systole par rapport à la première phase du cycle cardiaque se situent entre 50 et 70% et entre 15 et 25% respectivement (Marwick 2006 ; Sun, Popović et al. 2004).



(a) (b)

FIGURE 2.10 – (a) Courbe de strain global radial. (b) Courbe de strain global circonférentiel. Les pics systoliques et diastoliques sont indiqués sur les figures.

2.2 L'apprentissage profond

Les algorithmes d'apprentissage profond (Deep Learning, DL) forment une sous-catégorie au sein de l'ensemble plus large des algorithmes d'apprentissage machine (Machine Learning, ML). De ce fait, ils partagent de nombreuses caractéristiques avec ces derniers. En particulier, les algorithmes de DL, tout comme les algorithmes de ML, ont pour but de minimiser une fonction objectif ou fonction de coût préalablement déterminée pour mener à bien une tâche donnée. Pour ce faire ces deux types d'algorithmes reposent sur l'utilisation d'une grande quantité de données. Ces données sont séparées en deux groupes utilisés séparément pour entrainer et tester l'algorithme. Aucune donnée présente dans le groupe de test n'est présente dans le groupe d'entrainement. De cette façon les performances de l'algorithme sur le groupe de test révèlent sa capacité à généraliser ce qui a été appris durant l'entrainement à de nouvelles données. Une différence importante de l'apprentissage profond par rapport aux autres méthodes d'apprentissage machine repose sur l'utilisation d'un réseau de neurones pour traiter les données et minimiser la fonction de coût.

2.2.1 Les réseaux de neurones

Principe et entrainement

Les réseaux de neurones artificiels, qui s'inspirent du fonctionnement des neurones biologiques, permettent d'approximer des fonctions non-linéaires contenant de nombreux paramètres.
Un réseau de neurones est constitué d'une succession de couches de neurones où chaque neurone d'une couche est lié aux neurones des couches précédente et suivante par des arêtes pondérées par des poids. Les arêtes jouent le rôle des synapses dans le cerveau humain et les poids déterminent l'intensité du flux d'information entre deux neurones. Le réseau forme ainsi un graphe plus ou moins profond en fonction du nombre de couches. Chaque couche d'un réseau de neurones peut être vue comme une fonction prenant en entrée la sortie de la couche précédente et générant elle-même une sortie utilisée par la couche suivante.
Le réseau extraie de l'information à partir des données qui lui sont passées en entrée et propage cette information à travers les neurones vers la sortie du réseau pour obtenir la prédiction ("forward propagation" en anglais). Chaque neurone reçoit ainsi un signal de la couche précédente, traite ce signal et transmet ce signal traité à la couche suivante. Pour déterminer la quantité d'information transmise par le neurone o_j, on a recours à une fonction non-linéaire dite " fonction d'activation", différentiable, notée f. Celle-ci est appliquée à l'activation a_j du neurone, définie comme la somme pondérée des valeurs des k neurones de la couche précédente :

$$o_j = f(a_j) = f\left(\sum_{k=1}^{n} w_{kj} o_k + b\right) \tag{2.3}$$

avec o_k la sortie du k ème neurone de la couche précédente, n le nombre de neurones dans la couche précédente, w_{kj} le poids reliant o_k et o_j et b le biais. En notant x_k les données fournies au neurone k de la couche d'entrée du réseau, on peut remplacer o_k par x_k dans l'équation ci-dessus lorsque o_j est situé dans la première couche du réseau après la couche d'entrée.

Dans le cadre de cette thèse, la fonction d'activation Gaussian Error Linear Unit (GELU) (Hendrycks et Gimpel 2023) est utilisée pour tous les travaux. Etant donnée la fonction de répartition de la loi normale notée $\Phi(x) = P(X \leq x)$, GELU est définie de la façon suivante :

$$\text{GELU}(x) = x\Phi(x) = x \cdot \frac{1}{2}\left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right) \tag{2.4}$$

avec erf la fonction d'erreur telle que $\text{erf}(z) = 2\Phi\left(z\sqrt{2}\right) - 1$. Contrairement à la fonction d'activation RELU, (Fukushima 1975), GELU permet d'avoir des gradients négatifs pour $x < 0$, évitant ainsi le problème du "dying-RELU" (Lu, Shin et al. 2020 ; Arnekvist et al. 2020) où des neurones tombant dans la partie négative de RELU produisent systématiquement un signal nul. Une fois que ces neurones n'ont plus d'activation, il est difficile de les réactiver au cours de l'entrainement, rendant ces neurones inutiles. Par ailleurs, comme GELU n'est pas strictement monotone pour $x > 0$, des fonctions plus complexes peuvent être approximées (Hendrycks et Gimpel 2023). Enfin, contrairement à la fonction d'activation leaky-RELU (Maas 2013) qui rééchelonne les valeurs négatives de façon linéaire, GELU est bornée ce qui permet d'éviter que les valeurs d'entrées fortement négatives influencent négativement la valeur de sortie. Une représentation graphique des fonctions GELU, RELU et leaky-RELU est disponible Figure 2.11.

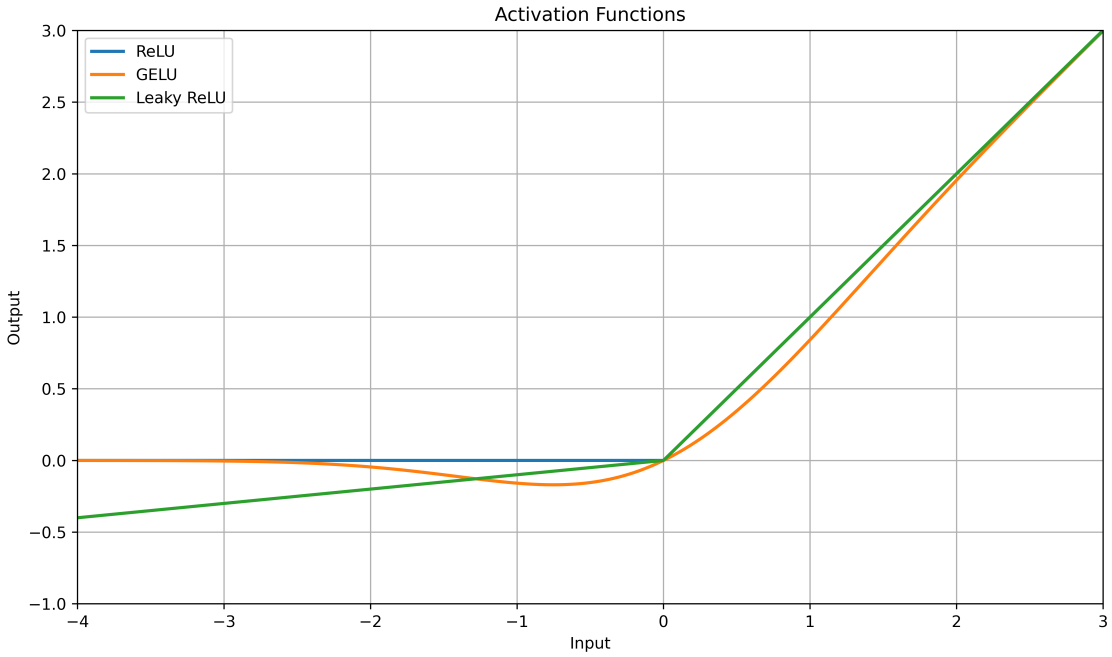


FIGURE 2.11 – RELU, GELU et leaky-RELU ($\alpha = 0.1$).

L'utilisation d'un réseau de neurones s'effectue en deux temps. La première étape permet d'entrainer le réseau en ajustant ses poids afin de minimiser une fonction de coût. La seconde étape correspond à la phase de test durant laquelle les poids du réseau sont gelés et le réseau est utilisé pour faire des prédictions sur de nouvelles données non présentes durant l'entrainement. L'intérêt des réseaux de neurones réside dans leur capacité à généraliser ce qui a été appris à de nouvelles données

durant la phase de test. Généralement, les données à disposition sont séparées en deux groupes, un groupe pour l'entrainement (qui peut être lui-même divisé en base d'entrainement et base de validation) et l'autre pour la phase de test.

Durant la phase d'entrainement, le réseau prend en entrée une partie des données, effectue une prédiction à partir de celles-ci et met à jour les poids (rétropropagation) de manière à minimiser la fonction de coût. Cela constitue une itération. Une epoch est composée des itérations nécessaires au parcours de toutes les données de la base. Le nombre d'epochs ainsi que le nombre d'itérations par epoch durant un entrainement sont des hyperparamètres qui doivent être définis au préalable.

Durant la phase de test, les poids du réseau sont gelés et on effectue uniquement une prédiction pour toutes les données de test. Un algorithme obtenant de bonnes performances sur le jeu de données d'entrainement ne sera pas nécessairement bon sur le jeu de données de test. Il se peut en effet que les fonctions de coût aient été mal choisies ou que le choix des hyperparamètres favorise de bonnes performances en entrainement et non en test. La différence entre les performances obtenues durant l'entrainement et la phase de test résulte souvent d'un sur-apprentissage (overfitting). Lorsque l'algorithme sur-apprend, les caractéristiques les plus intéressantes des données n'ont pas été comprises et l'algorithme s'est concentré sur le bruit contenu dans les données d'entrainement. Le modèle a appris "par coeur" sans réellement comprendre la relation entre les données d'entrée et la prédiction. Par conséquent, l'algorithme n'est pas capable de généraliser ce qui a été appris à d'autres données non présentes lors de l'entrainement. Le sur-apprentissage est souvent la conséquence de modèles trop complexes, possédant de trop nombreux paramètres. De ce fait, il est souvent utile d'utiliser des modèles ayant moins de paramètres pour réduire la différence de performance entre les phases d'entrainement et de test. Il est également possible d'avoir recours au concept de "weight decay", aussi appelé régularisation L2 qui consiste à ajouter une pénalité à la fonction de coût. Cette pénalité correspond à la norme euclidienne des poids du réseau pondérée par un hyperparamètre α :

$$C(y,p) = \tilde{C}(y,p) + \alpha\|w\|_2 = \tilde{C}(y,p) + \alpha\sqrt{\sum_i w_i^2} \tag{2.5}$$

avec $\tilde{C}(y,p)$ la fonction de coût définie pour résoudre la tâche, prenant en entrée la vérité terrain y et la prédiction p et w les poids du réseau. De cette façon, on cherche à minimiser également la valeur des poids du réseau. En effet, des poids élevés conduisent à des frontières de décision complexes qui correspondent bien aux données d'entrainement mais entrainent souvent une faible capacité à généraliser ce qui a été appris. Une autre manière de limiter la différence de performance entre les données d'entrainement et de test consiste à entrainer le modèle sur un jeu de données plus large. L'obtention de nouvelles données pouvant être complexe, il est possible de générer artificiellement celles-ci en appliquant des transformations aléatoires aux données existantes (on parle d'augmentation de données (Ying 2019)).
Généralement, une partie des données d'entrainement n'est pas traitée par le réseau durant chaque epoch. Ces données, dites de validation, permettent de mesurer les performances de l'algorithme de façon répétée après un certain nombre d'epochs durant l'entrainement. Ces données de validation sont également utiles pour éviter le sur-apprentissage. Il est en effet possible d'arrêter l'entrainement au moment où

l'erreur sur les données de validation commence à augmenter ou lorsque cette erreur augmente pendant plusieurs epochs d'affilée ; on parle d'early stopping (Wu et Shapiro 2006 ; Caruana, Lawrence et Giles 2000). Enfin, la capacité de généralisation du réseau peut être améliorée avec du drop-out qui consiste à ignorer certains neurones durant l'entrainement. Ainsi, à chaque itération, une proportion des neurones est masquée de façon aléatoire. De cette façon, le réseau se focalise moins sur des neurones spécifiques, l'obligeant à apprendre des caractéristiques plus robustes. En ignorant certains neurones, le réseau apprend à optimiser un réseau différent à chaque itération de la même manière que les méthodes d'ensemble, ce qui tend à améliorer la capacité de généralisation (Hinton et al. 2012). Un des intérêts du drop-out est qu'il peut être appliqué sur certaines couches du réseau de façon localisée, sans être appliqué à toutes les couches. De cette façon, on peut réduire la dépendance du réseau à certains "chemins" pour en privilégier d'autres (Seong, Oh et al. 2021).

On distingue généralement l'apprentissage supervisé de l'apprentissage non supervisé. Dans le cadre de l'apprentissage supervisé, la fonction de coût pénalise les prédictions qui s'écartent d'une "vérité de terrain" préalablement définie. Par exemple, dans le cadre de la segmentation d'images, la valeur de la fonction de coût est d'autant plus importante que la prédiction est éloignée de la segmentation préalablement établie. À l'inverse, l'apprentissage non supervisé n'utilise pas de vérité de terrain pour guider l'apprentissage du réseau. Lorsque l'algorithme comprend à la fois des fonctions de coût supervisées et non supervisées, on parle d'entrainement semi-supervisé.

Propagation vers l'avant et rétro propagation du gradient

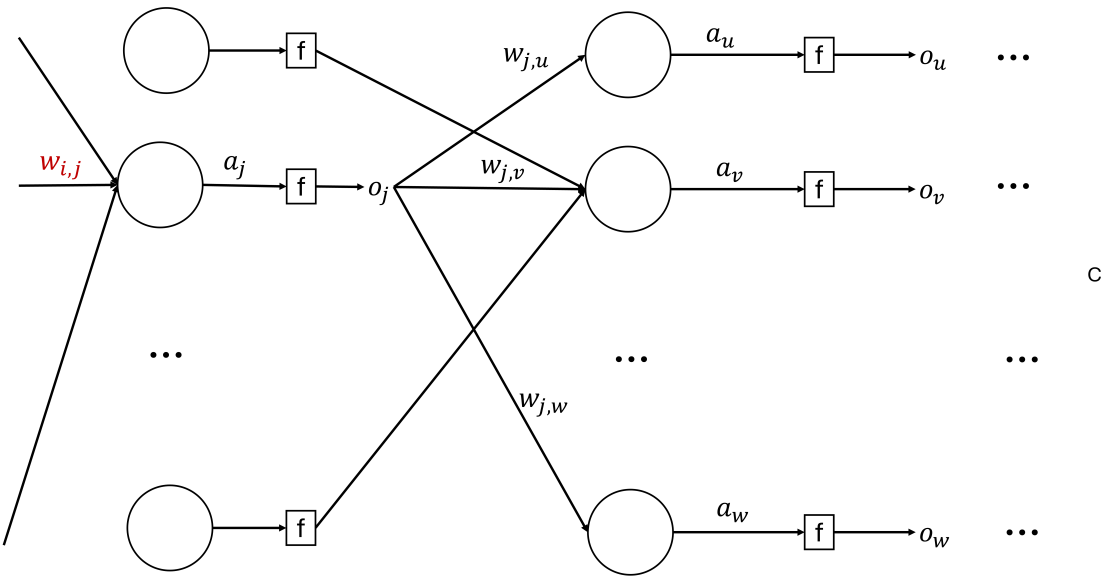


FIGURE 2.12 – Zoom sur un réseau pour la mise à jour du poids w_{ij}

Dans le cadre de l'apprentissage supervisé, et prenant l'exemple simple d'un

perceptron multi-couches (Multi Layer Perceptron MLP), on peut réécrire la fonction de coût $C(y, p)$, notée C par la suite par simplification, de la façon suivante :

$$C = C(y, p) = C(y, f^L(W^L f^{L-1}(W^{L-1} \cdots f^2(W^2 f^1(W^1 x)) \cdots))) \tag{2.6}$$

avec x les données d'entrées, W^l la matrice des poids entre les couches $l-1$ et l et f^l la fonction d'activation pour la couche l.

Pour chaque itération, une fois la prédiction p obtenue et la valeur de la fonction de coût calculée, les poids du réseau sont ajustés en commençant par les neurones les plus proches de la sortie et en remontant vers ceux les plus proches de l'entrée, c'est la rétro propagation du gradient ("backpropagation" en anglais). Elle consiste à calculer le gradient de la fonction de coût par rapport aux poids du réseau pour mettre à jour ces poids. Pour un poids w_{ij} quelconque (voir Figure 2.12), situé avant la dernière couche du réseau, on utilise le théorème de dérivation des fonctions composées pour obtenir le gradient de la fonction de coût :

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial C}{\partial o_j}\frac{\partial o_j}{\partial w_{ij}} = \frac{\partial C}{\partial o_j}\frac{\partial o_j}{\partial \mathrm{a}_j}\frac{\partial \mathrm{a}_j}{\partial w_{ij}} \tag{2.7}$$

a_j représente l'activation du neurone j et o_j sa sortie. On a donc :

$$\frac{\partial \mathrm{a}_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_{k=1}^{n} w_{kj} o_k\right) = \frac{\partial}{\partial w_{ij}}\left(w_{ij} o_i\right) = o_i. \tag{2.8}$$

En nommant $L = u, v, \ldots, w$ l'ensemble des neurones recevant o_j comme entrée comme dans la figure 2.12, on peut à nouveau utiliser le théorème de dérivation des fonctions composées pour calculer $\frac{\partial C}{\partial o_j}$ de façon récursive :

$$\frac{\partial C}{\partial o_j} = \sum_{\ell \in L}\left(\frac{\partial C}{\partial \mathrm{a}_\ell}\frac{\partial \mathrm{a}_\ell}{\partial o_j}\right) = \sum_{\ell \in L}\left(\frac{\partial C}{\partial o_\ell}\frac{\partial o_\ell}{\partial \mathrm{a}_\ell}\frac{\partial \mathrm{a}_\ell}{\partial o_j}\right) = \sum_{\ell \in L}\left(\frac{\partial C}{\partial o_\ell}\frac{\partial o_\ell}{\partial \mathrm{a}_\ell} w_{j\ell}\right) \tag{2.9}$$

On peut donc à présent réécrire l'équation 2.7 de la façon suivante :

$$\frac{\partial C}{\partial w_{ij}} = o_i \delta_j \tag{2.10}$$

avec :

$$\delta_j = \frac{\partial C}{\partial o_j}\frac{\partial o_j}{\partial \mathrm{a}_j} = \left(\sum_{\ell \in L} w_{j\ell}\delta_\ell\right)\frac{\partial o_j}{\partial \mathrm{a}_j} = \left(\sum_{\ell \in L} w_{j\ell}\delta_\ell\right)\frac{df(\mathrm{a}_j)}{d\mathrm{a}_j} \tag{2.11}$$

Nous avons ici calculé la dérivée de l'erreur par rapport au poids w_{ij}. Après chaque itération ce calcul est effectué pour tous les poids du réseau afin d'obtenir le gradient de la fonction de coût par rapport aux poids du réseau noté $\nabla C(\mathbf{W})$. Ce gradient indique à quel point chaque poids du réseau doit être modifié de façon à minimiser C. Dans sa forme la plus simple, pour l'itération t, les poids du réseau sont mis à jour par descente de gradient :

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \gamma \nabla C(\mathbf{W}_t) \tag{2.12}$$

où γ est le pas d'apprentissage. Les poids du réseau sont donc modifiés en allant dans la direction opposée du gradient puisque le gradient indique la direction de plus

forte pente. Si C est convexe et lipschitzienne, l'algorithme de descente de gradient garantit la convergence vers un minimum global. Dans la pratique, les fonctions de coût utilisées dans les algorithmes d'apprentissage profond sont rarement convexes et présentent de nombreux minimums locaux et points selles. Cela est lié au grand nombre de paramètres des fonctions que l'on cherche à approximer. Par conséquent, on utilise la descente de gradient pour trouver un minimum local qui permette à l'algorithme de bien généraliser à de nouvelles données. Il a été montré que ces minimums correspondent à des "régions plates" ou "vallées plates", par opposition aux "ravins pentus" (sharp ravine) qui donnent des modèles avec de mauvaises performances de généralisation (Hochreiter et Schmidhuber 1997 ; Hochreiter et Schmidhuber 1994 ; Keskar, Mudigere et al. 2017).

Théoriquement, l'algorithme de descente du gradient (GD) calcule $\nabla C(\mathbf{W})$ à partir de l'ensemble des données du jeu de données, c'est-à-dire qu'on utilise toutes les données du jeu de données pour chaque itération. En pratique, calculer le gradient en utilisant toutes les données est très couteux en termes de calcul et favorise la convergence vers un minimum local. Par conséquent, on utilise plutôt des variantes de la descente de gradient qui utilisent un ou plusieurs échantillons pour calculer le gradient. Selon que l'on utilise un seul ou plusieurs échantillons on parlera d'algorithme de descente de gradient stochastique (SGD) ou d'algorithme de descente de gradient par "mini-batch". Le batch représente un groupe de données passé en entrée du réseau à chaque itération. L'utilisation d'un "batch" de données plutôt qu'un seul échantillon permet d'obtenir une descente de gradient plus régulière et moins erratique qu'avec un seul échantillon car le gradient est moyenné sur plusieurs échantillons.

Le pas d'apprentissage

Dans l'algorithme de descente de gradient, il est difficile de choisir le pas d'apprentissage γ de façon optimale. S'il est trop faible, l'algorithme n'aura pas le temps de converger. S'il est trop élevé, il risque de diverger. Pour éviter cette divergence, une solution consiste à diminuer progressivement le pas d'apprentissage au fur et à mesure que l'entrainement progresse. Le pas d'apprentissage est alors une fonction décroissante du nombre d'itérations. En général, le pas d'apprentissage suit un rythme de décroissance linéaire, exponentiel, polynomial ou cosinus (Loshchilov et Hutter 2017). Il peut également être utile de partir d'un faible pas d'apprentissage et, avant de commencer sa décroissance, de l'augmenter progressivement durant les premières itérations ou epochs de l'entrainement afin d'atteindre une valeur plafond. On parle d'échauffement ("warm up"). Cela est fait afin d'éviter que les performances du réseau soient trop dépendantes des données traitées au tout début de l'entrainement où le pas d'apprentissage est le plus élevé. En effet, si les premières données sont trop complexes ou non représentatives du reste des données, le gradient peut descendre vers une direction non optimale. Récemment, des travaux (Smith 2017 ; Smith et Topin 2018) ont mis en avant que faire varier le pas d'apprentissage de façon cyclique peut mener à de meilleures performances. Ces méthodes font varier le pas d'apprentissage entre une valeur plancher et plafond de façon régulière au cours de l'entrainement, on parle de cycles. Cela a pour principal intérêt de permettre de sortir des points selles, très nombreux, et pour lesquels le gradient s'annule (points

critiques comme les minimums locaux et globaux). Dans ces régions le processus d'optimisation est lent car le gradient est faible. En augmentant le pas d'apprentissage, on peut éviter ou sortir de ces régions plus facilement. Des variantes de l'algorithme de descente de gradient stochastique sont souvent employées dont l'optimiseur Adam (Kingma et Ba 2017). L'efficacité d'Adam tient dans sa capacité à ajuster la mise à jour des poids au cours de l'entrainement de manière individuelle pour chaque paramètre du réseau. Cela permet aux paramètres d'être modifiés en fonction du "paysage" de la fonction de coût. Pour les zones avec une pente qui change de façon erratique, l'algorithme fera des pas plus petits. Les pas seront plus grands lorsque la pente est plus prévisible et change de façon moins chaotique. Mathématiquement, Adam calcule une moyenne mobile exponentielle du premier et du second moment du gradient m_t et v_t respectivement avec t l'index de l'itération courante :

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1)\nabla C(\mathbf{W}_t) \\ v_{t+1} &= \beta_2 v_t + (1-\beta_2)\left(\nabla C(\mathbf{W}_t)\right)^2 \end{aligned} \tag{2.13}$$

avec β_1 et β_2 des constantes de lissage déterminant à quel point on tient compte des gradients passés pour mettre à jour les poids. m_t et v_t permettent de tenir compte respectivement de l'inclinaison et de la vitesse de changement de la pente de la fonction de coût. Comme m et v sont initialisés à 0, Adam applique une correction à ces deux valeurs pour tenir compte d'un biais vers 0 au début de l'entrainement afin d'obtenir $\hat{m}$ et $\hat{v}$:

$$\begin{aligned} \hat{m} &= \frac{m_{t+1}}{1-\beta_1^t} \\ \hat{v} &= \frac{v_{t+1}}{1-\beta_2^t} \end{aligned} \tag{2.14}$$

A mesure que t augmente, les dénominateurs $1-\beta_1^t$ et $1-\beta_2^t$ tendent vers 1 ($\beta_1 < 1$ et $\beta_2 < 1$), ce qui réduit la correction appliquée. Cette correction a donc surtout de l'importance au début de l'entrainement. Les poids du réseau sont ensuite mis à jour avec les valeurs corrigées $\hat{m}$ et $\hat{v}$:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} \tag{2.15}$$

avec ϵ une valeur permettant d'éviter une division par zéro (souvent fixée à 10^{-8}). Bien que l'optimiseur Adam soit très populaire, il a été montré que, lorsque le pas d'apprentissage est bien choisi, l'optimiseur SGD peut donner de meilleures performances. Cela est lié à la modification non-uniforme du gradient qui, bien que permettant une convergence rapide dans les premières itérations de l'entrainement, peut entrainer de moins bons résultats de généralisation sur le jeu de données de test (Keskar et Socher 2017 ; Wilson et al. 2017 ; Zhang, Ma et al. 2018). La difficulté d'Adam à généraliser à de nouvelles données est aussi liée à une mauvaise gestion de la régularisation L2, aussi appelée "weight decay". De ce fait, Loshchilov et Hutter 2019 ont introduit une variante de Adam nommée AdamW qui modifie l'implémentation du weight decay :

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta\left(\frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} + w\mathbf{W}_t\right) \tag{2.16}$$

La normalisation des données

Il est d'usage de normaliser les données en entrée du réseau. On distingue la normalisation de la standardisation selon que l'on normalise les données entre 0 et 1 ou si l'on s'assure que les données sont centrées et réduites respectivement. Cela permet au gradient d'être plus stable durant l'entrainement car les caractéristiques d'entrée ont la même échelle. Dans le cas contraire le réseau serait trop sensible aux outliers ayant une valeur en entrée trop haute.
Au-delà de la normalisation des données d'entrées, il est important que les caractéristiques soient normalisées avant chaque couche du réseau. En effet, du fait de l'initialisation aléatoire des poids du réseau et de la variabilité dans les données d'entrées, la distribution des caractéristiques avant chaque couche n'est pas stable avec une moyenne et une variance qui varient, on parle "d'internal covariance shift". Pour cette raison, Ioffe et Szegedy 2015 ont introduit la batch-normalisation qui vise à obtenir des caractéristiques ayant une distribution standardisée avant chaque couche du réseau. La batch-normalisation rendrait le réseau moins sensible au choix des hyperparamètres et permettrait une meilleure capacité de généralisation. Contrairement aux données d'entrées qui peuvent être standardisées en utilisant la moyenne et l'écart type de l'ensemble du jeu de données d'entrainement, on utilise ici la moyenne et l'écart type du batch et standardise ensuite chaque dimension d de l'entrée d'une couche de neurones $x = (x^{(1)}, \ldots, x^{(d)})$. On peut alors calculer le résultat $y_i^{(d)}$ de la batch-normalisation pour une dimension k :

$$y_i^{(d)} = \gamma^{(d)} \hat{x}_i^{(d)} + \beta^{(d)} \tag{2.17}$$

avec $\hat{x}_i^{(d)}$ le vecteur standardisé, $\gamma^{(d)}$ et $\beta^{(d)}$ des paramètres appris par le réseau.

Durant l'entrainement, la batch-normalisation calcule et met progressivement à jour une approximation de la moyenne et de la variance de l'ensemble du jeu de données à l'aide d'une moyenne mobile exponentielle. Ces moyenne et variance sont utilisées en inférence pour standardiser les caractéristiques. De cette manière la standardisation ne dépend plus de la taille du batch en inférence. Bien que la batch-normalisation ait été introduite pour lutter contre l'internal covariance shift, il n'est pour l'instant pas établi clairement pourquoi elle donne de si bons résultats. En particulier Santurkar et al. 2018 ont montré que la batch-normalisation n'avait pas d'effet sur l'internal covariance shift mais qu'elle permettrait plutôt d'avoir un "paysage" de la fonction de coût plus régulier. De plus, la meilleure capacité de généralisation du réseau serait liée à une minimisation de la fonction de coût vers des minimums locaux plats (Hochreiter et Schmidhuber 1997 ; Keskar, Mudigere et al. 2017).
Il existe d'autres méthodes permettant de standardiser les caractéristiques au sein du réseau. Nous présentons ici uniquement les principales techniques de normalisation. Étant donnée une carte de caractéristiques de taille (N, C, H, W) où N est la taille du batch, C le nombre de caractéristiques, H et W les hauteur et largeur de la carte de caractéristiques, la batch-normalisation normalise séparément chaque caractéristique. La normalisation est donc effectuée selon N, H et W. La "layer normalisation" (Ba, Kiros et Hinton 2016) normalise toutes les caractéristiques avec les mêmes constantes μ et σ (normalisation selon C, H, et W). L'instance normalisation (Ulyanov, Vedaldi et Lempitsky 2017) standardise séparément chaque

caractéristique et chaque élément du batch (normalisation selon H et W). Enfin, la Group normalisation (Wu et He 2018) divise le nombre de caractéristiques C en g groupes et calcule μ et σ indépendamment pour chaque groupe (normalisation selon H, W et C/g). La Figure 2.13 présente ces différentes techniques de normalisation de façon schématique.
La batch-normalisation conduit généralement à de meilleurs résultats que les autres méthodes lorsque la taille du batch est suffisante. En revanche lorsque la taille du batch est limitée à 1, la batch-normalisation ne fonctionne plus.

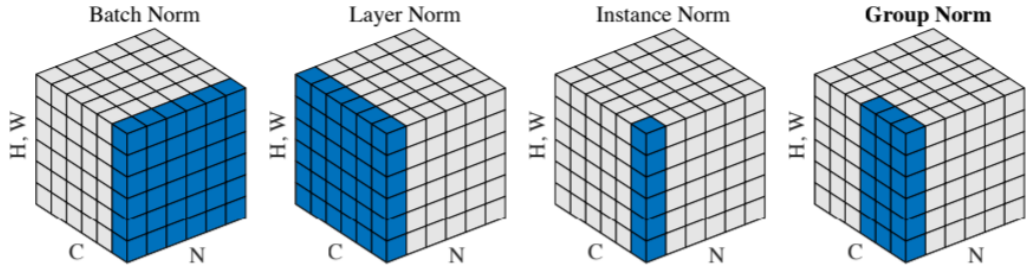


FIGURE 2.13 – Représentation schématique des dimensions à partir desquelles sont calculés μ et σ pour les différentes méthodes de normalisation (Wu et He 2018).

2.2.2 Convolutional Neural Network (CNN)

Les réseaux de neurones convolutionnels (CNN) incluent dans leur architecture une ou plusieurs couches de convolution. Chaque couche prend en entrée une carte de caractéristiques (tenseur), et produit une autre carte de caractéristiques. Pour la première couche de convolution, la carte de caractéristiques d'entrée est constituée de l'image originale à traiter de profondeur 1 (image en niveau de gris) ou 3 (image couleur). Les couches de convolution appliquent un nombre N de filtres pouvant varier d'une couche à l'autre (32, 64, 128, etc...), produisant des nouvelles cartes de caractéristiques de profondeur N. Plus le nombre de filtres est important et plus le réseau sera capable d'approximer des fonctions complexes. Par contre, il aura aussi plus de paramètres à estimer et pourra conduire à du sur-apprentissage ou sous-apprentissage (si le nombre d'itérations n'est pas suffisant).
Les couches de convolution réalisent la convolution de la carte d'entrée par un filtre ayant une taille fixe, généralement bien plus petite que la taille de la carte de caractéristiques. Le filtre glisse le long de la hauteur et de la largeur (dans le cas du 2D) de la carte de caractéristiques d'entrée et effectue la somme des produits terme à terme entre ses valeurs et celles de la carte de caractéristiques (un bais est également ajouté à cette opération). Pour un signal 2D discret tel que les images, la convolution se définit donc de la façon suivante :

$$F_{out}(i,j) = (I * K)(i,j) = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} F_{in}(i+m, j+n)K(m,n) \tag{2.18}$$

où K est un filtre de taille $k \times k$, F_{out} est la carte de caractéristiques de sortie et F_{in} est la carte de caractéristiques d'entrée sur laquelle on effectue la convolution pour le pixel de coordonnée i, j. Les coefficients du filtre constituent les poids du réseau

à apprendre. On nomme champ réceptif (receptive field) la taille de la fenêtre 2D d'une carte de caractéristiques d'entrée ayant permis de produire une seule valeur de la carte de sortie. Ce champ réceptif est nécessairement local du fait du fonctionnement des convolutions et croit en proportion du nombre de couches de convolution dans le CNN. Ainsi, plus le nombre de couches de convolution est important et plus le champ réceptif sera large, donnant accès à un contexte spatial plus large. Par conséquent, les premières couches de convolution d'un CNN ont généralement accès à un contexte restreint et permettent d'extraire de l'information sémantiquement pauvre comme les contours des objets. À l'inverse, les couches de convolution situées près de la sortie du réseau ont accès à de l'information sémantique plus riche, *i.e* une représentation plus abstraite et complexe de l'image initiale. Un exemple schématique de champ réceptif se trouve Figure 2.14. Chaque couche de convolution fait croitre le champ réceptif de façon linéaire (si tous les filtres ont la même dimension). Plutôt que d'empiler les couches de convolution pour augmenter le champ réceptif, il est possible d'utiliser des filtres de plus grande taille. Néanmoins, cette dernière approche nécessite davantage de poids à apprendre pour aboutir au même champ réceptif puisqu'une seule convolution avec une fenêtre de taille 5 par 5 requiert 25 poids (sans le biais), contre 18 pour deux fenêtres successives de taille 3*3. Une solution permettant d'accroitre le champ réceptif sans augmenter le nombre de paramètres à apprendre est d'utiliser des convolutions dilatées (avec des "trous" dans la fenêtre de convolution). Cela permet de faire croitre le champ réceptif de façon exponentielle (Kalchbrenner et al. 2017 ; Luo, Li et al. 2016 ; Kuo 2016 ; Yu et Koltun 2016). Les opérations de pooling qui permettent de sous-échantillonner les cartes de caractéristiques font croitre le champ réceptif de façon multiplicative.

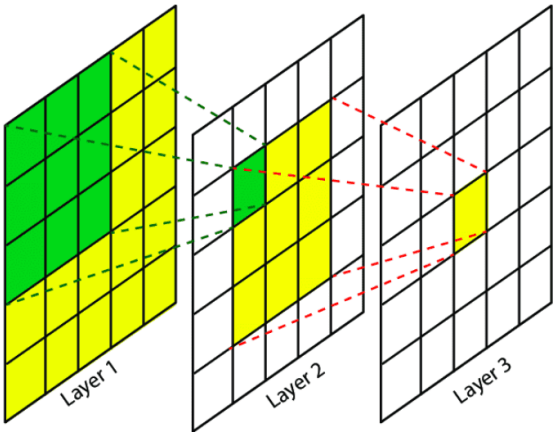


FIGURE 2.14 – Le champ réceptif des couches de convolution.

Dans une couche de convolution, les poids du filtre (paramètres à apprendre) sont partagés pour toutes les positions spatiales. Les CNN traitent donc les images de façon locale. Cela a trois conséquences importantes :

- Les convolutions sont mieux adaptées au traitement des images que les couches entièrement connectées car elles reposent sur un nombre de poids beaucoup plus faible. Pour une image de taille 100 par 100 pixels, une couche entièrement connectée est composée de 10000 poids par neurone de sortie contre seulement 25 pour une convolution avec un filtre de taille 5 par 5. En pratique, le traitement des images avec des couches entièrement connectées est très difficile, ce qui a rendu l'emploi des CNN très populaires. L'utilisation d'un nombre réduit de paramètres permet aussi d'éviter les problèmes "d'évanouissement ou d'explosion du gradient" liés à la rétropropagation (Venkatesan et Li 2017 ; Balas, Kumar et Srivastava 2019).
- Les poids des filtres de convolution sont indépendants de la position (pixel) dans l'image. Cela différencie les convolutions des mécanismes d'attention, évoqués plus bas, dont les poids d'attention dépendent de la zone de l'image (Dai, Liu et al. 2021 ; Zeiler et Fergus 2014 ; Vaswani et al. 2017).
- Puisque le filtre glisse le long de la carte de caractéristiques d'entrée, les convolutions sont invariantes à la position des objets dans l'image et sont donc capables de reconnaitre des objets quelle que soit leur position, on parle de "Translation Equivariance" (Dai, Liu et al. 2021 ; Zhang, Tanida et al. 1988 ; Zhang, Itoh et al. 1990).

2.2.3 Transformers

L'architecture transformer a été introduite par Vaswani et al. 2017 initialement pour le traitement du langage naturel et donc, pour traiter des données séquentielles représentées comme une suite de "tokens". Les transformers reposent sur des mécanismes d'attention permettant de pondérer ces tokens de façon différente en fonction de leur importance. Ces mécanismes d'attention sont présentés plus loin dans le chapitre. Un transformer est composé d'un encodeur et d'un décodeur. L'encodeur est constitué d'une couche de "self-attention" suivie d'un MLP. Le décodeur a une structure similaire mais contient également une couche de "cross-attention" entre la "self-attention" et le MLP. Que ce soit pour l'encodeur ou le décodeur, les MLP constituent l'essentiel des poids du modèle Transformer. Ces couches permettent au réseau d'apprendre la relation entre les tokens en se basant sur la sortie des couches d'attention situées juste avant. Une description de l'architecture Transformer se trouve Figure 2.15.

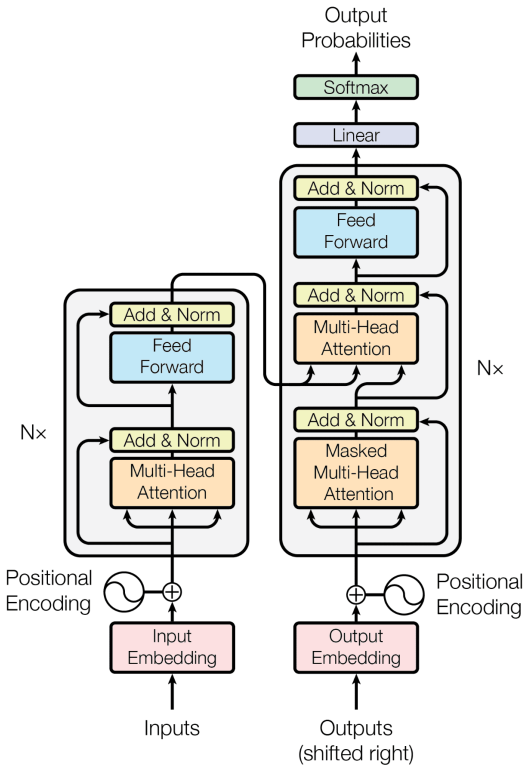


FIGURE 2.15 – L'architecture transformer. À gauche l'encodeur, à droite le décodeur (Vaswani et al. 2017).

Contrairement aux convolutions, les transformers classiques ne prennent pas en compte l'ordre des éléments dans la séquence d'entrée. Par conséquent, Vaswani et al. 2017 ajoutent une information de position à chaque token nommée "positional encoding" et notée $PE \in \mathbb{R}^C$ où C est la profondeur de la carte de caractéristiques. Ces derniers prennent la forme de sinus pour les dimensions paires et de cosinus pour les dimensions impaires :

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/C}}\right) \tag{2.19}$$

$$\mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{(2i+1)/C}}\right) \tag{2.20}$$

avec $i \in [0; \frac{C}{2} - 1]$ et pos la position du token dans la séquence. Du fait de la périodicité des fonctions sinus et cosinus, il est possible d'obtenir les mêmes valeurs pour des positions pos différentes dans la séquence pour une dimension donnée i. Cependant, en faisant varier la fréquence le long de la dimension C, on s'assure que l'information de position est unique pour chaque token (la fréquence dépend de pos et de i). Dans l'architecture de Vaswani et al. 2017, les positional encodings sont passés uniquement à l'entrée du réseau avant l'encodeur et le décodeur. Il est également possible de les passer avant chaque couche d'attention comme cela a été proposé par Carion et al. 2020.

Les mécanismes d'attention *i.e.* self et cross attention constituent le cœur de l'architecture transformer. Les self et cross attention effectuent les mêmes calculs, seule l'entrée de ces couches varie. Ces calculs sont effectués en parallèle pour plusieurs "têtes" (head en anglais). Chaque tête permet au réseau de porter attention

à une zone différente de la séquence d'entrée. Étant donné le nombre de têtes h, la dimension d'une tête est $d = \frac{C}{h}$. Considérons maintenant une séquence $X \in \mathbb{R}^{N \times C}$ de N tokens de dimension C. Pour chaque tête, la couche de self-attention effectue d'abord 3 projections linéaires à l'aide des matrices $W_i^Q \in \mathbb{R}^{C \times d}$, $W_i^K \in \mathbb{R}^{C \times d}$ et $W_i^V \in \mathbb{R}^{C \times d}$ de façon à obtenir la query, la key et la value respectivement notées Q_i, K_i et $V_i \in \mathbb{R}^{N \times d}$ avec $i \in [1; h]$ l'indice de la tête :

$$Q_i = XW_i^Q \quad K_i = XW_i^K \quad V_i = XW_i^V \tag{2.21}$$

Le mécanisme d'attention est effectué à l'aide de la méthode "scaled dot product". La similarité Sim_i entre la query et la key pour une tête i est d'abord obtenue en calculant la distance cosinus entre chaque token de la query et de la key, normalisée par $\sqrt{d}$:

$$Sim_i = \frac{Q_i K_i^T}{\sqrt{d}} \tag{2.22}$$

Puis, la fonction softmax est appliquée selon la dimension de la séquence N de manière à avoir des probabilités dont la somme vaut 1. Le produit scalaire entre la sortie de la fonction softmax et la value permet alors d'obtenir le résultat final de la self-attention pour une seule tête :

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(Sim_i\right) V_i \tag{2.23}$$

avec

$$\text{softmax}(x_k) = \frac{e^{x_k}}{\sum_{j=1}^{N} e^{x_j}} \tag{2.24}$$

Le mécanisme d'attention calcule donc la correspondance entre Q et K et se sert de ces poids pour rééchelonner les valeurs de V. Le calcul de l'équation 2.23 est effectué en parallèle pour chaque tête. Le résultat de chacune de ces têtes est concaténé puis fusionné à l'aide de la matrice de projection W^O :

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h) \cdot W^O \tag{2.25}$$

où $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$.
La cross-attention que l'on trouve dans le décodeur fonctionne de la même façon que la self-attention à la différence près que la key et la value sont calculés à partir d'une séquence différente de celle utilisée pour obtenir la query. Pour la cross-attention, on peut donc réécrire l'équation 2.21 de la manière suivante :

$$Q_i = X_1 W_i^Q \quad K_i = X_2 W_i^K \quad V_i = X_2 W_i^V \tag{2.26}$$

avec $X_1 \in \mathbb{R}^{N_1 \times C}$ la séquence utilisée pour obtenir la query et $X_2 \in \mathbb{R}^{N_2 \times C}$ la séquence à partir de laquelle sont calculées la key et la value. On n'a pas nécessairement $N_1 = N_2$, c'est-à-dire que la séquence utilisée pour la query et la key n'ont pas forcément la même longueur. Dans le cadre de l'architecture présentée par Vaswani et al. 2017, X_2 correspond à la séquence de sortie de l'encodeur tandis que X_1 représente la séquence issue de la couche précédente du décodeur. Après chaque MLP et chaque couche d'attention, la séquence est normalisée à l'aide de la layer normalisation (Ba, Kiros et Hinton 2016) et une connexion résiduelle permet d'ajouter l'entrée de la couche à la sortie. Ces connexions sont similaires à ce qui a été introduit dans Resnet (He, Zhang et al. 2015) et permettent d'éviter d'avoir des

gradients trop petits lors de la backpropagation. Cela permet d'avoir des architectures transformer très profondes, *i.e.* contenant de nombreuses couches.

Par défaut, les transformers ne peuvent pas déduire la position relative ou absolue des tokens dans la séquence. Par conséquent, et comme expliqué plus haut, on ajoute des positional encodings sinusoïdaux ou appris pour leur permettre de comprendre la position absolue des tokens. De nombreuses recherches ont également portées sur l'introduction, au sein des architectures transformer, d'informations de position relative entre les tokens (Relative Positional Encoding (RPE)). Puisque les convolutions sont capables d'extraire cette information à partir de leur fenêtre glissante (cf. section 2.2.2), elles ont été introduites au sein des architectures transformer pour faciliter la convergence (Bello et al. 2020 ; Chu et al. 2023 ; Wu, Xiao et al. 2021 ; Xiao et al. 2021). D'autres recherches préfèrent stocker l'information de position relative au sein d'une matrice de paramètres qui est indexée au moment du calcul de l'attention. L'information de position relative ainsi récupérée est ajoutée aux scores d'attention avant de rééchelonner la value. Dans ces derniers travaux, l'équation 2.21 se réécrit :

$$Sim_i = \frac{Q_i K_i^T + b_i}{\sqrt{d}} \tag{2.27}$$

où b_i est l'information de position relative entre tokens.
Certains travaux intègrent directement l'information de position relative au résultat du calcul de similarité entre la query et la key. L'information de position relative est donc indépendante de Q, K et V. On a donc :

$$b_i = r_i \tag{2.28}$$

avec $r_i \in \mathbb{R}^{N \times N}$ les poids récupérés au sein de la matrice de paramètres pour la tête i. Ici chaque poids est un scalaire. Cela a l'avantage de ne pas trop peser sur les ressources mémoire et d'être rapide à calculer (Dai, Liu et al. 2021 ; Liu, Lin et al. 2021 ; Raffel et al. 2020).
Pour renforcer l'expressivité de l'information de position relative, d'autres recherches font dépendre les RPE d'une ou plusieurs composantes de l'attention. On effectue alors le produit scalaire entre les RPE d'un côté et la query, key et/ou value de l'autre avant d'ajouter le résultat au score d'attention (Shaw, Uszkoreit et Vaswani 2018 ; Ramachandran et al. 2019 ; Huang, Liang et al. 2020 ; Wu, Peng et al. 2021 ; Dai, Yang et al. 2019 ; Wang, Zhu et al. 2020). Dans le cas où les RPE dépendent de la query, on a :

$$b_i = Q_i r_i^T \tag{2.29}$$

avec $r_i \in \mathbb{R}^{N \times N \times d}$. Ici chaque poids est un vecteur de dimension d. Il faut noter que les RPE sont souvent partagés entre les têtes. On peut trouver Figure 2.16 la différence entre positions relatives dépendantes ou indépendantes des tokens d'entrées.

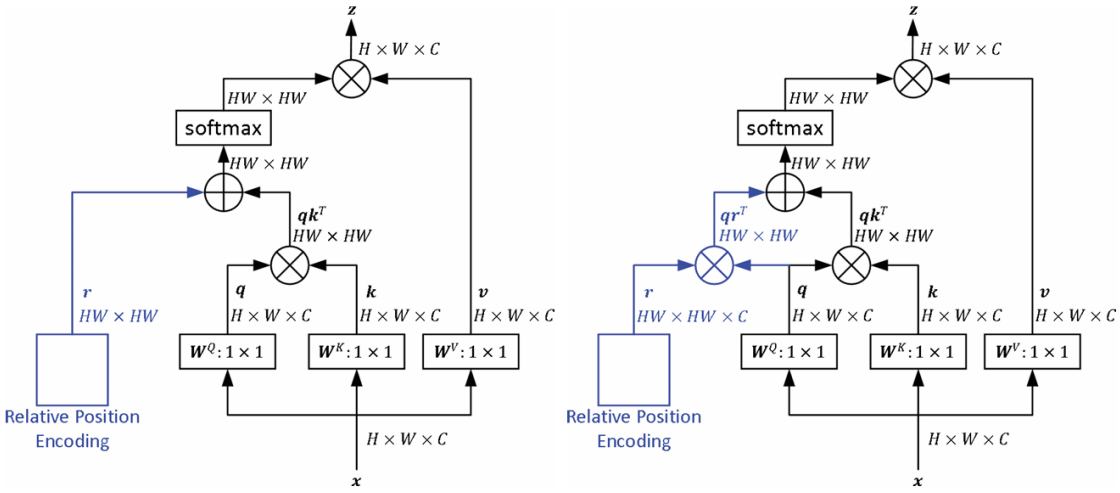


FIGURE 2.16 – Différence entre l'information de position relative indépendante (gauche) ou dépendante (droite) des tokens (ici dépendante de Q) (image isue de Wu, Peng et al. 2021).

Initialement pensés pour le NLP, les transformers ont connus un essor important dans le domaine de la vision depuis les travaux de Dosovitskiy et al. 2021 qui ont montré qu'en divisant une image en patchs de taille égale, un réseau n'utilisant pas de couche de convolution obtient de meilleures performances que les CNN pour la classification d'images. Il est notamment montré qu'un des avantages des transformers est que, contrairement aux CNN, les premières couches de réseau ont directement accès à toutes les zones de l'image, *i.e.* leur champ réceptif correspond à l'image entière. Les architectures transformer disposent donc d'un contexte plus large pour analyser les images. Néanmoins, la complexité de la self-attention étant $\mathcal{O}(N^2)$, avec N la longueur de la séquence d'entrée, les architectures transformer sont assez peu adaptées au traitement d'images. En effet, pour réduire la consommation mémoire il est souvent préférable d'utiliser des couches transformer uniquement pour les résolutions les plus faibles, limitant de fait l'intérêt de leur utilisation. Par conséquent, de nombreux travaux ont porté sur la réduction de cette complexité algorithmique, par exemple en effectuant l'attention dans une zone réduite de l'image (Liu, Lin et al. 2021 ; Wang, Zhu et al. 2020).
Plusieurs méthodes tirent parti du principe d'associativité des matrices pour réécrire l'équation d'attention de façon à atteindre une complexité linéaire avec le nombre de tokens (Katharopoulos et al. 2020 ; Munkhdalai, Faruqui et Gopal 2024 ; Zhuoran et al. 2021 ; Han, Pan et al. 2023). Une fonction de normalisation ou d'activation ρ est appliquée à la query et la key et le produit scalaire est effectué entre cette key normalisée et la value plutôt qu'entre la query et la key. L'idée est d'approximer le rôle de la fonction softmax généralement utilisée dans la self-attention à l'aide de la fonction de normalisation. Le processus de self-attention se définit de la façon suivante :

$$D(Q, K, V) = \rho(QK^T)V \tag{2.30}$$

où ρ est la fonction softmax. Les articles précités réécrivent la self-attention de la façon suivante :

$$E(Q, K, V) = \rho_q(Q)(\rho_k(K)^T V) \tag{2.31}$$

Pour bien comprendre pourquoi l'utilisation de ρ_q et ρ_k permet d'approximer la

fonction softmax on peut se placer dans le cas où ρ est une fonction de mise à l'échelle :

$$\rho(x) = \frac{x}{\sqrt{N}}$$
$$\rho_q(x) = \rho_k(x) = \frac{x}{\sqrt{N}}$$

On a alors :

$$D(Q, K, V) = \frac{QK^T}{N}V \tag{2.32}$$

et

$$E(Q, K, V) = \frac{Q}{\sqrt{N}}\left(\frac{K^T}{\sqrt{N}}V\right) \tag{2.33}$$

En utilisant la propriété d'associativité du produit matriciel ainsi que de commutativité du produit par un scalaire, on obtient :

$$\begin{aligned} E(Q, K, V) &= \frac{Q}{\sqrt{N}}\left(\frac{K^T}{\sqrt{N}}V\right) \\ &= \frac{1}{N}Q(K^TV) \\ &= \frac{1}{N}(QK^T)V \\ &= \frac{QK^T}{N}V = D(Q, K, V) \end{aligned}$$

On voit donc qu'il y a égalité entre $D(Q, K, V)$ et $E(Q, K, V)$ lorsque ρ est une fonction de mise à l'échelle. Néanmoins, dans les faits, ρ n'est pas une fonction de mise à l'échelle mais la fonction softmax. Dans ce cas, les études précédentes essayent d'approximer la fonction softmax à l'aide de ρ_q et ρ_k. En particulier, les valeurs obtenues par ρ_q et ρ_k doivent être un nombre positif. Le plus souvent ρ_q et ρ_k sont définis comme la fonction ELU + 1 (Clevert, Unterthiner et Hochreiter 2016), ce qui permet d'éviter d'avoir un gradient nul pour les valeurs négatives tout en gardant une sortie positive. Etant donné Q, K et $V \in \mathbb{R}^{N\times d}$, en effectuant le produit scalaire entre K et V plutôt qu'entre Q et K, la complexité algorithmique de la self-attention passe de $\mathcal{O}(N^2d)$ à $\mathcal{O}(Nd^2)$. Or, d est souvent inférieur à N, surtout pour les tâches liées à la vision comme le traitement d'images ou de vidéos. On a donc bien une attention de complexité linéaire avec le nombre de tokens N. Un schéma récapitulatif de l'attention linéaire est présenté Figure 2.17.

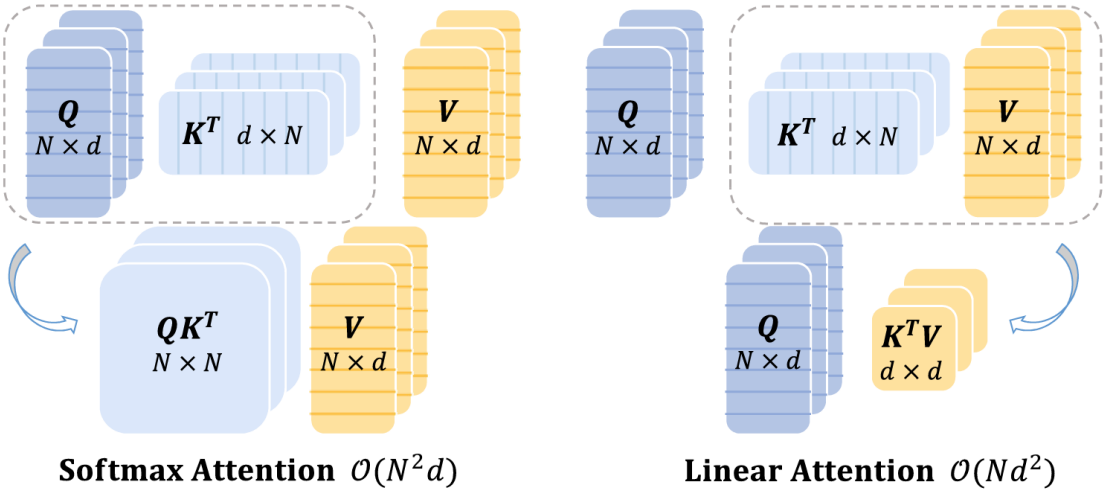


FIGURE 2.17 – Différence entre Softmax attention et attention linéaire. Schéma issu de Han, Pan et al. 2023.

Après une décomposition en valeurs singulières, Wang, Li et al. 2020 montrent que la matrice résultant du calcul de la similarité entre Q et K a un rang faible. De ce fait, ils utilisent une projection supplémentaire sur K et V de façon à réduire le nombre de tokens, rendant le processus d'attention linéaire avec la taille de la séquence d'entrée. Wang, Xie, Li et al. 2021 procèdent également à une réduction du nombre de tokens présents dans la key et la value. Pour ce faire ils réarrangent la séquence sous la forme d'une carte de caractéristiques et la sous-échantillonnent à l'aide d'une convolution 2D avec stride. Zhu, Su et al. 2021 ont mis au point un processus d'attention déformable permettant de porter attention uniquement à certains tokens de la séquence d'entrée. Le réseau apprend alors par lui-même à quelle zone de l'image porter attention, accélérant la convergence et permettant un processus d'attention avec une complexité linéaire.

Les architectures de type transformer sont souvent pré-entrainées sur de larges jeux de données pour produire des résultats satisfaisants. En effet leurs performances dépendent plus de l'étape de pré-entrainement que les CNN (Vaswani et al. 2017 ; Dosovitskiy et al. 2021 ; Khan et al. 2022 ; Luo, Wang et al. 2021). De ce fait de nombreux travaux ont porté sur la mise au point de méthodes efficaces de pré-entrainement. C'est ainsi que plusieurs auteurs ont introduit des méthodes de pré-entrainement auto-supervisées, c'est à dire ne nécessitant pas de vérité terrain. Ces méthodes incluent souvent de la prédiction de zones volontairement masquées ou bruitées dans la séquence initiale (inpainting) (Khan et al. 2022 ; Li, Xu et al. 2022 ; Atito, Awais et Kittler 2022 ; Li, Chen et al. 2021). Dans le cadre général de la vision par ordinateur, ces architectures sont souvent pré-entrainées sur des jeux de données contenant plusieurs dizaines ou centaines de millions d'images comme JFT-300M (Sun, Shrivastava et al. 2017), Imagenet-1k (Russakovsky et al. 2015), Imagenet-21k (Ridnik et al. 2021), webvision (Li, Wang, Li et al. 2017) ou encore OpenVision (Kuznetsova et al. 2020). Dans le cadre de l'imagerie médicale, il n'existe pas de jeux de données publics contenant autant de données. Cela découle de la difficulté d'annotation ainsi que de restrictions concernant le partage de données médicales (Chen, Ma et Zheng 2019). De ce fait l'utilisation des transformers est limitée. Il est

tout de même possible d'obtenir des performances surpassant un réseau entièrement basé sur les convolutions en utilisant un nombre restreint de couches transformer entrainées "from scratch" comme nous le verrons plus bas. Il est à noter qu'une des raisons majeures limitant l'utilisation de modèles pré-entrainés sur les jeux de données précédemment cités semble résider dans l'absence d'images en niveau de gris que l'on trouve régulièrement dans le domaine médicale (Alzubaidi et al. 2021 ; Cherti et Jitsev 2022 ; Xie et Richmond 2018).

2.3 Segmentation d'images

La segmentation d'images consiste à regrouper les pixels d'une image et à leur attribuer une catégorie spécifique en fonction de certaines caractéristiques communes. Ces caractéristiques peuvent par exemple être la couleur des pixels, des textures ou encore la forme des objets. On cherche donc à diviser l'image en segments ou régions de manière à la simplifier pour des traitements ultérieurs. On distingue généralement la segmentation sémantique de la segmentation d'instances.

La segmentation sémantique consiste à assigner à tous les pixels de l'image une étiquette correspondant à leur catégorie. Ainsi, si plusieurs chats sont présents dans une image, ils auront tous la même étiquette. La segmentation d'instance cherche à différencier dans une image les différentes instances d'une même catégorie, ce qui reviendra, dans le cas précédant, à donner une étiquette différente à chacun des chats.

On parle de "things" pour désigner les choses dénombrables qui doivent être segmentées dans la segmentation d'instance. On utilise généralement le mot "stuff" pour parler de choses indénombrables que l'on cherche à segmenter par la segmentation sémantique. La combinaison de la segmentation sémantique et la segmentation d'instance est la segmentation panoptique. Il s'agit d'attribuer à tous les pixels de l'image une catégorie, tout en différenciant les groupes de pixels appartenant à la même catégorie mais pas à la même instance. On cherche alors à segmenter à la fois les "things" et les "stuff".

On distingue généralement les algorithmes de segmentation classiques, qui reposent sur des certaines règles ou principes mathématiques, des approches par apprentissage profond qui reposent quant à elles sur l'extraction de caractéristiques à partir d'une grande quantité de données à l'aide de réseaux de neurones. Les méthodes reposant sur l'apprentissage profond permettent généralement d'obtenir des segmentations de meilleure qualité que les méthodes "classiques" (Plaksyvyi, Skublewska-Paszkowska et Powroźnik 2023 ; Caicedo et al. 2019 ; Jin et al. 2019 ; Sehar et Naseem 2022). Cela s'explique principalement par une meilleure capacité de généralisation. Les méthodes classiques ont en effet plus de difficulté à s'adapter à des données légèrement différentes et nécessitent souvent pour cela de modifier manuellement certains hyperparamètres.
Néanmoins, les performances des algorithmes d'apprentissage profond dépendent grandement de la quantité de donnée annotées disponible. Il peut être préférable de privilégier des méthodes classiques lorsque le jeu de données est de petite taille. En outre le temps d'inférence a tendance à être plus faible pour les algorithmes classiques que pour les approches par apprentissage profond (Bianconi et al. 2021),

en particulier sur CPU (Caicedo et al. 2019).
Parmi les algorithmes de segmentation "classiques", on peut distinguer les méthodes par regroupement, les méthodes par croissance de régions, les approches reposant sur les champs aléatoires de Markov, les techniques de "graph-cut", et enfin les modèles de contours actifs et d'ensemble de niveaux. Nous revenons plus en détails sur chacune de ces méthodes dans la partie 3.1. Les approches par apprentissage profond reposent sur des réseaux de neurones convolutionnels afin d'extraire automatiquement des caractéristiques utiles à la segmentation d'objets d'intérêt. Nous détaillons ces dernières méthodes dans la section 3.2.

2.4 Le flux optique

Le flux optique représente le mouvement des objets dans une scène, entre deux images. Dans le cadre du traitement d'image, il désigne le plus souvent le mouvement de chaque pixel ou d'un groupe de pixels entre deux images. Le calcul du flux optique suppose qu'il n'y a pas de changement de luminosité entre les deux images et donc, que la valeur des pixels reste la même :

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.34}$$

avec $I(x, y, t)$ l'intensité d'un pixel de coordonnées (x, y) au temps t et $(\Delta x, \Delta y, \Delta t)$ la quantité de mouvement dans les 3 dimensions (mouvement entre la première et la seconde image). Le but du calcul du flux optique est alors de retrouver, pour chaque pixel (x, y), les valeurs de Δx et Δy. Tout comme pour les algorithmes de segmentation, on peut établir une distinction entre les approches "classiques" d'estimation du flux optique et les méthodes reposant sur l'apprentissage profond. Les approches classiques cherchent à résoudre des systèmes d'équations en supposant que certaines hypothèses sont vérifiées (régularité du mouvement, constance de la luminosité, faibles déplacements, etc...). Ces modèles sont abordés plus en détails dans la section 5.1. On peut distinguer deux catégories de méthodes pour l'estimation du flux optique par apprentissage profond.

Il existe plusieurs jeux de données publics disposant de vérité terrains de flux optique pour tous les pixels de l'image (c.f section 5.2). Pour ces données, la fonction de coût des architectures intègre le plus souvent une simple mesure de similarité entre le flux prédit et la vérité terrain. Ces modèles reposent souvent sur un volume de coût afin de calculer la corrélation entre différentes zones des deux images. Ces algorithmes sont détaillés Section 5.2.

Dans le cadre des images médicales, le flux optique de vérité terrain est rarement présent et le calcul du flux se traite le plus souvent de la même façon que l'estimation d'un champ de déformation pour le recalage d'images. Ces approches reposent donc essentiellement sur un apprentissage non supervisé, même si elles utilisent parfois des annotations de segmentation afin d'obtenir des mouvements conservant la forme de certaines structures d'intérêt. On a le plus souvent recours à une fonction de coût de similarité des textures et de régularité afin d'estimer un champ de déplacement entre deux images (souvent à partir du spatial transformer (Jaderberg et al. 2015)). L'absence de vérité terrain de flux optique a incité de nombreuses recherches à essayer d'améliorer la régularité du flux estimé en réduisant le nombre de pixels avec

un Jacobien négatif. Nous décrivons ces algorithmes plus en détails en section 5.4.

2.5 Présentation des bases de données

Nous présentons ici succinctement les principaux jeux de données existant pour la segmentation cardiaque petit axe.

2.5.1 Sunnybrook Cardiac Data (SCD)

Le jeu de données Sunnybrook Cardiac Data (SCD) (Radau et al. 2009) a été utilisé pour un challenge à la conférence MICCAI 2009 consistant à segmenter le ventricule gauche sur des images IRM petit axe. Le jeu de données comprend uniquement des annotations pour le ventricule gauche et le myocarde, *i.e.* contours de l'endocarde et de l'épicarde. Les données de 45 sujets sont présentes dans le jeu de données. 9 sujets sont catégorisés comme sain. Les 36 autres patients sont divisés en 3 groupes de taille égale : hypertrophie du ventricule gauche, insuffisance cardiaque avec et sans infarctus. Pour chacun des groupes, le volume télé-systolique et télé-diastolique du ventricule est connu, ainsi que la fraction d'éjection et la masse du ventricule.

2.5.2 Automated Cardiac Diagnosis Challenge (ACDC)

Le jeu de données Automated Cardiac Diagnosis Challenge (ACDC) (Bernard et al. 2018) a été créé pour un challenge lors de la conférence MICCAI 2017. Le but était de comparer la performance des algorithmes d'apprentissage profond pour la segmentation cardiaque ainsi que pour la classification de différentes pathologies cardiaques. Le jeu de données est composé de 100 sujets pour l'entrainement et 50 sujets supplémentaires pour la phase de test. Les sujets sont divisés en 5 groupes de taille égale : volontaires sains, patients avec des infarctus du myocarde, patients avec des cardiomyopathies dilatées, patients avec des cardiomyopathies hypertrophiques et patients avec des ventricules droits anormaux. Les données sont acquises en orientation petit axe en utilisant des IRM Siemens 1.5 et 3 T. Pour chaque patient et pour chaque instant du cycle cardiaque, plusieurs coupes 2D sont alignées pour former un volume 3D allant de la base à l'apex du cœur. Seuls les volumes en télé-diastole et télé-systole disposent d'une annotation de segmentation pour le ventricule droit, le ventricule gauche et le myocarde. L'annotation a été faite par un expert clinique de façon manuelle. Il y a entre 28 et 40 volumes par cycle cardiaque. La résolution spatiale des pixels est isotropique et située entre 1.37 et 1.68 mm2/pixel. L'épaisseur de coupe est de 5 ou 8 mm.

2.5.3 Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge (M&Ms)

Le jeu de données M&Ms (Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge) (Campello et al. 2021) a été mis au point pour un challenge de la conférence MICCAI 2020 dont la but était la segmentation d'images

petit-axe. Il contient 375 sujets, parmi lesquels des sujets sains, des cas de cardiomyopathie hypertrophique ou de cardiomyopathie dilatée. Les données proviennent de 3 pays différents (Espagne, Allemagne, Canada) et ont été générées en utilisant des IRM conçus par 4 fabricants différents (Siemens, GE, Philips, Canon). Les données contiennent des images générées par IRM 1.5 ou 3T. L'épaisseur de coupe est comprise entre 9.2 et 10 mm et la taille des pixels se situe entre 0.85 et 1.42 mm. Les volumes contiennent entre 10 et 13 coupes et un cycle cardiaque comprend entre 25 et 30 volumes. Le jeu de données d'entrainement contient 150 sujets dont les images ont été obtenues sur des IRM fabriquées par 2 des 4 fabricants. Les volumes télé-systolique et télé-diastolique de ces 150 sujets ont été annotés par des experts cliniques en suivant les mêmes règles que pour ACDC. Ces annotations couvrent le ventricule gauche, le ventricule droit et le myocarde. Les données d'entrainement contiennent également 25 sujets supplémentaires issues du 3ème fabriquant et dont les données ne sont pas annotées. Le jeu de données de test contient 50 sujets pour chacun des 3 fabricants du jeu de données d'entrainement, ainsi que 50 sujets supplémentaires issus du 4ème fabricant non présent dans les données d'entrainements. Ce dernier groupe de 50 sujets permet de tester la capacité de généralisation des modèles. Le jeu de données M&Ms permet aux modèles d'apprendre des caractéristiques partagées par les données de plusieurs fabricants, et donc de limiter la baisse de performance généralement observée en testant les modèles sur des données issues d'un fabriquant différent des données d'entrainement. M&Ms donne également la possibilité d'entrainer des modèles à l'adaptation de domaine ("domain adaptation"), i.e. que le réseau apprenne un espace commun entre des données sources correspondant à un fabriquant et des données cibles correspondant à un autre fabriquant.

2.5.4 UK Biobank

La UK Biobank (Bycroft et al. 2018) regroupe les données médicales de 500000 sujets âgés de 40 à 69 ans au moment du recrutement (2006-2010). Ce jeu de données a été créé pour les besoins d'une étude prospective visant à suivre l'évolution des sujets sur plusieurs années de manière à identifier les causes et à mieux diagnostiquer et traiter certaines maladies comme le cancer, les maladies cardiaques ou encore l'arrêt cardiaque. Parmi les 500000 sujets, 48000 participants ont terminé l'étude d'imagerie (Raisi-Estabragh et al. 2021) qui comprend des IRM du cerveau, cœur et abdomen, échocardiographie de l'aorte, ainsi que des scanners rayons X du corps entier. Les données cardiaques contiennent des séquences bSSFP en orientation sagittale, coronale, ainsi que des IRM ciné en long axe et petit axe. Les données contiennent également des images taguées, des IRM de l'aorte ainsi que des images T1 et des images du flux aortique. Pour les données petit axe, les volumes sont composés de coupes allant de la base à l'apex du cœur et couvrant le ventricule droit et gauche. L'épaisseur de coupe est de 8 mm avec un écart de 2 mm entre coupes. La taille d'un pixel est isotropique de 1.8 mm. Les séquences ont une résolution temporelle de 32 ms et ont été interpolées pour contenir 50 phases (20 ms). Les données sont acquises avec une IRM Siemens 1.5 T. Le jeu de données a été analysé de façon à obtenir le volume des ventricules, la fraction d'éjection, le volume d'éjection, la masse du myocarde ainsi que les valeurs de strain (Petersen, Matthews et al. 2016). Actuellement, le ventricule droit, gauche et le myocarde ont été annotés manuellement pour 5000 sujets (Petersen, Aung et al. 2016).

2.5.5 Quorum

Quorum est un jeu de données interne au Laboratoire d'Imagerie Biomédicale, non public et issue d'une étude clinique [6]. Il contient des données multi-centre et multi-fournisseur et inclut des patients ayant subi un infarctus aigu du myocarde (IM) avec différents degrés de gravité. Les images ont été acquises dans 24 centres différents en utilisant des scanners IRM de 3 fabricants (Siemens, General Electric et Philips). Des images petit-axe couvrant le cœur de sa base à son apex ont été acquises pendant la phase aiguë de l'IM, à l'aide d'un scanner IRM de 1,5 ou 3 Tesla, une taille de pixel allant de 0,68 à 2,34 mm^2 (taille moyenne des pixels de 1,43 mm^2) et une épaisseur de coupe entre 6 et 8 mm (épaisseur moyenne des coupes de 7,4 mm). Dans les travaux de cette thèse, nous distinguons deux versions différentes de ce jeu de données : pour la première version, les annotations de segmentation de référence de 271 patients ont été obtenues pour toutes les phases du cycle cardiaque en utilisant le logiciel CardioTrack (Lamy et al. 2018). Ce logiciel utilise un algorithme de suivi des caractéristiques pour suivre les principaux points de contour initialisés manuellement sur la première image de la séquence. Les annotations de segmentation sont générées en fonction de la position de ces points de contour pour chaque image. Pour chaque coupe 2D, le ventricule droit, le ventricule gauche et le myocarde sont segmentés par le logiciel. La plupart des annotations ont été fournies uniquement pour 3 coupes dans le volume, représentant un total de 34452 coupes 2D. Pour ces coupes, le logiciel a également été utilisé pour obtenir les déformations radiales et circonférentielles de référence. Chaque séquence couvre un cycle cardiaque et contient entre 20 et 80 phases, représentant au total 912 séquences vidéo de coupes 2D.
Pour la seconde version, les volumes en télé-diastole (TD) et en télé-systole (TS) de 195 de ces 271 patients ont été annotés manuellement en utilisant un logiciel commercial (QMass, Medis, Leiden, Pays-Bas, version 4.0.24.4). Les annotations sont générées automatiquement par le logiciel et modifiées, si nécessaire, par un expert clinique. Pour toutes les coupes des volumes, les structures cardiaques annotées sont les mêmes que celles segmentées par CardioTrack, représentant 4072 coupes 2D.

2.5.6 Synthèse des différentes bases de données

	SCD	ACDC	M&M's	UK BioBank	Quorum
# patients	45	150	375	5000	271
Structures annotées	LV/MYO	LV/RV/MYO	LV/RV/MYO	LV/RV/MYO	LV/RV/MYO
Champ magnétique (T)	···	1.5/3	1.5/3	1.5	1.5/3
Taille du pixel (mm)	···	[1.37 ; 1.68]	[0.85 ; 1.42]	1.8	[0.68 ; 2.34]
Epaisseur de coupe (mm)	···	5/8	[9.2 ; 10]	8	[6 ; 8]
# fabricants	1	1	4	1	3
# pathologies	3	4	2	0	1

TABLE 2.1 – Principales bases de données pour la segmentation cardiaque sur image IRM.

Dans le cadre de cette thèse nous avons utilisé les jeux de données ACDC et Quorum. En effet ACDC est disponible publiquement et contient de nombreuses pathologies ainsi qu'un nombre de patients satisfaisant. A l'inverse, les données de la UK Biobank ne sont pas disponibles publiquement. Le jeu de données SCD contient

6. https ://www.action-groupe.org/fr/etude/quorum

peu de patients et l'annotation pour le ventricule droit n'est pas disponible. M&M's est similaire à Quorum. Ce dernier étant le jeu de données du LIB, il a été privilégié.

2.6 Discussion et positionnement

Dans cette thèse, nous nous intéressons à la mesure des biomarqueurs à partir des images IRM petit-axe. La plupart de ceux-ci (2.1.4) peut être mesurée à partir d'une segmentation. Par conséquent, dans un premier temps, les travaux de cette thèse se concentrent sur l'élaboration d'un algorithme de segmentation 2D. Afin d'améliorer les algorithmes de segmentation déjà existants, notre attention s'est portée sur l'utilisation des mécanismes d'attention qui permettent de réduire l'écart sémantique entre l'encodeur et le décodeur dans les architectures de type U-Net (Ronneberger, Fischer et Brox 2015). Nous montrons également qu'une utilisation particulière de ces processus d'attention permet de réduire le bruit des données haute résolution en provenance de l'encodeur.

Cet algorithme de segmentation permet d'extraire les points de contours de l'endocarde et de l'épicarde. Ces points de contour sont utilisés pour calculer les déformations radiale et circonférentielle du myocarde. En effet, pour la déformation radiale, on calcule la distance entre chaque point de contour de l'endocarde et le point le plus proche de l'épicarde (segment vert Figure 2.18). Pour la déformation circonférentielle, on calcule la distance entre deux points voisins à l'intérieur du myocarde (segment rouge Figure 2.18). Ces calculs s'effectuent pour chaque phase (image) du cycle cardiaque, afin de mesurer l'évolution des distances dans le temps. En faisant la moyenne de toutes les distances radiales et circonférentielles, on obtient une mesure de la déformation globale du myocarde.

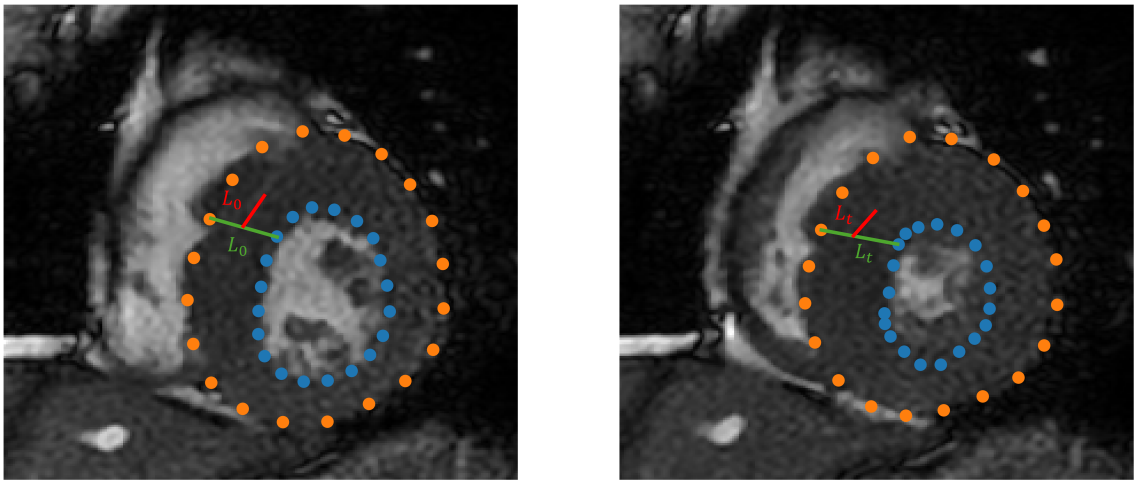


FIGURE 2.18 – Points de contour utilisés pour calculer le strain radial et circonférentiel du myocarde en télé-diastole (gauche) et télé-systole (droite). En rouge le segment utilisé pour le calcul du strain circonférentiel, en vert un segment utilisé pour le calcul du strain radial. L_0 et L_t sont les variables de l'équation 2.2.

Si la déformation globale du myocarde permet d'identifier une réduction de la force de contraction du muscle cardiaque en consultant les courbes de déformation (Figure 2.10b et 2.10a), elle ne permet pas d'identifier avec précision la région au sein du myocarde responsable de cet affaiblissement. Par conséquent, il est courant de diviser le myocarde en 17 segments dont 6 segments pour les coupes basales et de mi-hauteur (numérotés de 1 à 12), et 4 segments ainsi que l'apex pour les coupes apicales (numérotés de 13 à 17) (Cerqueira et al. 2002). Ces segments correspondent à différentes régions du myocarde irriguées par les 3 principales artères coronaires : l'artère coronaire droite (RCA), l'artère circonflexe (LCX) et l'artère interventriculaire antérieure (LAD) (Figure 2.19). De ce fait, il est intéressant de calculer la déformation myocardique pour chacun de ces segments de façon à localiser avec précision la région du myocarde responsable de la plus faible capacité de contraction. En particulier, les médecins peuvent ainsi identifier quelle artère coronaire dysfonctionne et donc plus facilement détecter les ischémies ou thromboses.

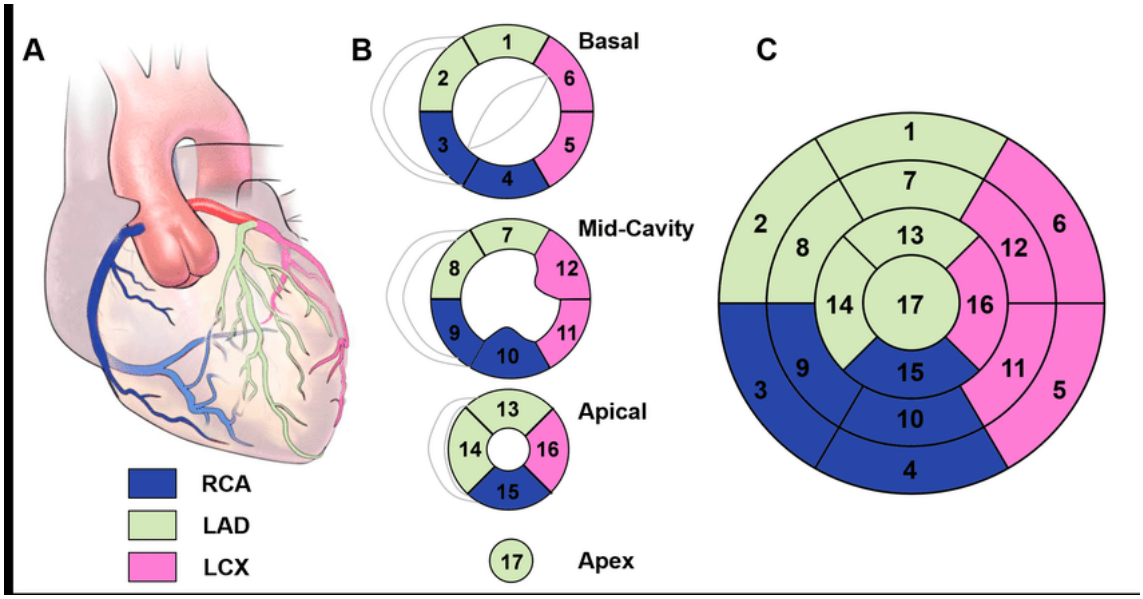


FIGURE 2.19 – Le myocarde est divisé en segments en fonction des zones irriguées par les artères coronaires. Image issue de Velasco Jimeno 2019.

Pour calculer le strain régional, la segmentation avec un seul label pour le myocarde n'est plus suffisante car il est nécessaire d'obtenir la déformation pour un nombre restreint de points de l'endocarde et de l'épicarde présents dans chaque segment du myocarde. Il est donc nécessaire d'avoir recours à des algorithmes capables d'estimer la position d'un point de contour tout au long du cycle cardiaque. Pour ce faire, de nombreuses études ont utilisé des algorithmes de suivi de caractéristiques (Lamy et al. 2018 ; Schuster et al. 2013 ; Augustine et al. 2013 ; Buss et al. 2015). D'autres méthodes utilisent l'imagerie par marquage ("tagging imaging" en anglais) pour mesurer de façon plus précise la déformation du myocarde (Osman et al. 1999 ; Aletras et al. 1999 ; Moore et al. 2000). Le "tagging" applique un motif grillagé sur l'image. Cela est rendu possible par l'utilisation de deux impulsions RF avec un angle de bascule de 45^o pour obtenir des bandes horizontales et verticales. Cela permet de faciliter le suivi du mouvement du muscle cardiaque sur la séquence ciné. Néanmoins, le tagging nécessite de suivre un protocole spécifique pour l'acquisition

des images (utilisation d'un gradient magnétique appelé "Spatial Modulation of Magnetization") et l'acquisition d'images taguées est peu courant en routine clinique. De plus, les motifs produits par le tagging ont tendance à s'effacer au cours du temps du fait du mouvement du myocarde. Plus récemment, des travaux ont été conduits pour estimer le mouvement du myocarde sur des séquences ciné en utilisant des algorithmes d'apprentissage profond (Yu, Sun et al. 2020 ; Qin, Bai et al. 2018 ; Zhang, You et al. 2022). Cependant, ces méthodes se limitent à estimer le mouvement sans en déduire le strain. D'autres études ont eu recours à l'estimation du flux optique avec des algorithmes d'apprentissage profond pour calculer le strain sur séquences IRM ciné (Morales, Boomen et al. 2021 ; V. Graves et al. 2023 ; Alvarez-Florez et al. 2023 ; Masutani et al. 2023 ; Wang, Sun et al. 2023). Cependant, ces méthodes utilisent des volumes 3D ce qui nécessite d'avoir recours uniquement à des paires de volumes à l'entrainement et donc ne permet pas d'assurer une cohérence temporelle dans la séquence. Les méthodes utilisant toutes les images du cycle cardiaque reposent sur des images taguées (Ferdian et al. 2020 ; Ye, Kanski, Yang, Chang et al. 2021). Enfin, certains travaux utilisent un algorithme d'apprentissage profond pour segmenter une région d'intérêt puis, ensuite, effectuent le suivi à l'aide de méthodes manuelles (Hammouda et al. 2020), ou se contentent d'obtenir le strain global sans effectuer de suivi (Kwan et al. 2024). Par ailleurs ces algorithmes reposent sur des architectures employant des convolutions. Ils n'intègrent donc pas de bloc transformer qui ont pourtant montré leur efficacité pour l'estimation du flux optique et la segmentation.

A l'inverse, les travaux de cette thèse présentent des algorithmes reposant entièrement sur l'apprentissage profond pour effectuer le suivi des points de contours. Ceux-ci utilisent des couches transformers ainsi que plusieurs images du cycle cardiaques durant l'entrainement ce qui permet d'inclure l'information temporelle présente dans les séquences pour effectuer des prédictions.

Deuxième partie

Segmentation d'IRM cardiaque

Chapitre 3

Etat de l'art

Ce chapitre introduit les principales méthodes de segmentation de la littérature en les regroupant en deux grandes catégories : les méthodes traditionnelles et celles utilisant l'apprentissage. La dernière section présente les mesures de performance utilisées pour comparer les segmentations obtenues par les différentes méthodes.

3.1 Segmentation d'images avec les méthodes traditionnelles

Parmi les méthodes les plus populaires, on distingue les méthodes par regroupement, par croissance de régions, celles utilisant les champs de Markov ou le graph-cut et enfin les méthodes fondées sur les contours actifs ou les ensembles de niveaux.

3.1.1 Méthodes par regroupement

Les méthodes de regroupement comme le K-means (Lloyd 1982 ; MacQueen et al. 1967) consistent à regrouper les pixels ayant des caractéristiques similaires au sein d'une image. Après avoir défini K valeurs de centroïdes, l'algorithme K-means assigne à chaque pixel le centroïde le plus proche. La valeur du centroïde du groupe est ensuite mise à jour en calculant la moyenne des valeurs du groupe. Ces étapes sont répétées de manière à affiner la valeur des centroïdes. Au final, l'algorithme permet de diviser les pixels de l'image en K groupes. Cette méthode fonctionne aussi bien pour la segmentation binaire que multi-classe. L'inconvénient de cette méthode est de devoir définir au préalable le nombre de groupes K. L'algorithme mean shift (Cheng 1995) est une variante du K-means qui détermine automatiquement la valeur de K. L'algorithme mean-shift cherche les modes de la fonction de densité de l'image. La valeur des pixels de l'image est mise à jour à partir de la moyenne des pixels se trouvant dans un voisinage. L'algorithme de mélange de gaussiennes est également un algorithme souvent utilisé pour la segmentation d'image par regroupement (Nguyen et Wu 2013 ; Gupta et Sortrakul 1998 ; Caillol, Pieczynski et Hillion 1997). Il suppose que la distribution des pixels de l'image est multimodale et peut se modéliser à l'aide de plusieurs gaussiennes de moyennes et écart-types différents. On estime alors les paramètres de ces gaussiennes qui maximisent la vraisemblance à l'aide de l'algorithme itératif espérance-maximisation.

3.1.2 Méthodes de croissance de régions

Les algorithmes de croissance de régions constituent un autre groupe de méthodes permettant de segmenter une image (Pohle et Toennies 2001 ; Tremeau et Borel 1997 ; Tang 2010). Ces algorithmes reposent sur l'emploi de graines ("seed") à partir desquelles les pixels partageant une caractéristique commune croissent. Plus spécifiquement, les pixels se trouvant dans le voisinage de la graine (4 ou 8 connexités) sont inclus dans la région de la graine en fonction d'un certain critère. Il peut s'agir de comparer l'intensité ou la couleur du pixel considéré à celle de la graine ou de la moyenne des pixels de la région. Le procédé est répété pour les pixels ajoutés à la région. La définition des graines à l'initialisation de l'algorithme joue un rôle déterminant sur la nature de la carte de segmentation obtenue. Ce processus peut être automatisé à l'aide d'algorithmes de clustering ou en utilisant l'histogramme de l'image. Un exemple d'algorithme par croissance de région est l'algorithme de partage des eaux ("watershed") (Levner et Zhang 2007 ; Huang et Chen 2004 ; Shafarenko, Petrou et Kittler 1997). Cet algorithme est souvent utilisé pour séparer des objets ou pour passer d'une segmentation binaire à une segmentation multiclasse. L'algorithme considère l'image comme un relief topographique où les zones à forte intensité correspondent à des pics tandis que les régions de faible intensité sont assimilées à des vallées. Généralement, en partant d'une segmentation binaire, on calcule la distance au fond. Les graines sont alors définies comme les minimums locaux dans l'inverse de l'image des distances. Le processus de croissance par région est alors effectué à partir de tous les minima locaux. On dit qu'on inonde les bassins ou vallées. Lorsque les bassins sont complètement remplis, la délimitation entre les objets (ligne de partage des eaux) correspond aux zones de l'image ou les bassins se rencontrent.

3.1.3 Champs aléatoires de Markov

Une autre catégorie de méthodes pour la segmentation d'image s'appuie sur les champs aléatoires de Markov (Markov Random Field, MRF) (Held et al. 1997 ; Liu, Li et al. 2018 ; Panjwani et Healey 1995). L'image à segmenter est alors assimilée à un graphe non orienté où les pixels sont des variables aléatoires. La valeur des pixels dépend uniquement des pixels se trouvant dans un voisinage. Etant donné un ensemble de variables aléatoires $Y = Y_1, Y_2, ..., Y_n$ où n est le nombre de pixels dans l'image, les MRF cherchent à calculer $P(Y)$, *i.e.* la distribution jointe de l'ensemble des variables aléatoires. D'après le théorème de Hammersley-Clifford, si $P(Y) > 0$ alors le MRF peut être représenté comme un champ aléatoire de Gibbs et la loi de probabilité jointe devient une mesure de Gibbs :

$$P(Y) = \frac{1}{Z} \exp(-E(Y)) \tag{3.1}$$

où Z est la fonction de partition telle que $Z = \sum_Y \exp(-E(Y))$ qui permet de s'assurer que $\sum_Y P(Y) = 1$ (distribution de probabilité) et $E(Y)$ est une fonction d'énergie qui détermine la valeur que prend Y_i $\forall i \in [1; n]$. Lorsque l'énergie est faible, $P(Y)$ est plus élevé. On encourage donc les pixels à prendre des valeurs qui minimisent $E(Y)$ de façon à maximiser $P(Y)$. Dans le cas du modèle de Pott utilisé

pour la segmentation multi-classe on a :

$$E(Y) = -\sum_{i,j \in \mathcal{N}} \delta(Y_i, Y_j) \tag{3.2}$$

où i et j sont deux pixels se trouvant dans le même voisinage $\mathcal{N}$. $\delta(Y_i, Y_j)$ est le symbole de Kronecker qui vaut 1 si $Y_i = Y_j$ et 0 sinon. Ce modèle encourage les pixels proches à prendre les mêmes valeurs. En général, on utilise une version modifiée du modèle de Pott de façon à ce que la valeur d'un pixel ne dépende pas uniquement de la valeur du pixel voisin (on aurait alors une image avec une seule classe), mais aussi de l'intensité du pixel indépendamment de son voisinage :

$$E(Y) = -\sum_{i,j \in \mathcal{N}} \delta(Y_i, Y_j) w_{ij} - \lambda \sum_{i \in \mathcal{V}} h_i \tag{3.3}$$

où λ est un poids de régularisation qui détermine le compromis entre régularité et fidélité aux données. w_{ij} est le poids représentant l'intensité du lien entre le pixel i et j. $\mathcal{V}$ est l'ensemble des pixels dans l'image. h_i est un biais qui modélise la probabilité pour le pixel i d'appartenir à la classe $k \in [1; K]$ avec K le nombre de classes. h_i dépend le plus souvent des caractéristiques locales du pixel (intensité, couleur, texture,...). Si $K = 2$ alors l'équation précédente décrit le modèle Ising, un cas particulier du modèle de Pott utilisé pour la segmentation binaire.
Il n'est en fait pas possible de calculer $P(Y)$. En effet, il est nécessaire pour cela de calculer Z qui est une somme de toutes les configurations possibles pour $X_i \ \forall i \in [1; n]$. Comme n est élevé pour une image, la complexité algorithmique du calcul de Z est trop grande. Par conséquent, on a recours à des méthodes d'approximations de $P(Y)$ comme l'échantillonnage de Gibbs ou de Metropolis-Hasting. Une variante des MRF souvent utilisée pour la segmentation (par exemple dans le modèle DeepLab (Chen, Papandreou et al. 2016)) est le champ aléatoire conditionnel (CRF en anglais) (He, Zemel et Carreira-Perpinan 2004). Les CRF sont des MRF qui tiennent compte d'observations X et, conformément à l'inférence bayésienne, cherchent à modéliser la probabilité *a posteriori* $P(Y|X)$. Il ne s'agit donc plus ici d'estimer la distribution jointe $P(Y)$. Pour cette raison on dit que les modèles CRF sont discriminatifs par opposition aux modèles génératifs. Dans le cadre de la segmentation d'images, Y peut désigner les étiquettes à déterminer tandis que X peut correspondre aux pixels de l'image. Pour les modèles CRF la probabilité conditionnelle $P(Y|X)$ s'exprime de la façon suivante :

$$P(Y \mid X) = \frac{1}{Z(X)} \exp(-E(Y, X)) \tag{3.4}$$

Avec :

$$Z(X) = \sum_{Y} \exp(-E(Y, X)) \tag{3.5}$$

On voit donc que les modèles CRF ont la même impossibilité que les modèles MRF de calculer l'inférence exacte du fait de la variable de normalisation $Z(X)$. Les modèles CRF essayent néanmoins d'approximer $P(Y \mid X)$ en utilisant des modèles paramétriques et en apprenant un ensemble de paramètres θ. En particulier les

modèles CRF estiment le maximum de la vraisemblance conditionnelle :

$$L(\theta) = \prod_{i=1}^{N} P(Y^{(i)} \mid X^{(i)}; \theta)$$
$$= \sum_{i=1}^{n} log(P(Y^{(i)} \mid X^{(i)}; \theta))$$

L'apprentissage de θ peut se faire à l'aide de la descente de gradient.

3.1.4 Algorithmes Graph-cut

Les algorithmes de graph-cut sont également populaires pour réaliser la segmentation d'une image (Vicente, Kolmogorov et Rother 2008 ; Grosgeorge et al. 2013 ; Liu, Song et al. 2019). De façon similaire aux MRF, chaque pixel de l'image représente un nœud d'un graphe. Il s'agit ici plus spécifiquement d'un graphe de flux pour lequel il existe, en supplément des pixels de l'image, un nœud source noté s et un nœud puit noté t. Les algorithmes de graph-cut représentent le problème de la segmentation d'images comme l'écoulement d'un fluide depuis la source vers le puit, en transitant par les nœuds du graphe. Les nœuds sont connectés entre eux par des arêtes disposant d'une capacité. Cette capacité désigne la quantité maximale de flux qui peut passer par cette arête. La quantité de flux sortant d'un nœud est nécessairement égale à la quantité entrante. Le problème de segmentation d'images par graph-cut peut se résumer à un problème de maximisation de la valeur du flux du graphe, ce qui correspond à la coupe de capacité minimale (théorème max-flow/min-cut). La "coupe" d'un graphe sépare les arêtes du graphe en deux ensembles disjoints (S, T) tel que $s \in S$ et $t \in T$. Si l'on désigne par X_c l'ensemble des arêtes reliant un nœud de S et un nœud de T, alors on peut définir la capacité d'une coupe $C(S, T)$ de la façon suivante :

$$C(S, T) = \sum_{(u,v) \in X_c} c_{uv} \tag{3.6}$$

où (u, v) est une arête reliant le nœud u au nœud v et c_{uv} est la capacité de cette arête. Une illustration est disponible Figure 3.1a. La capacité d'une coupe désigne donc la somme des capacités des arêtes reliant des nœuds appartenant à deux ensembles différents. Dans le cadre de la segmentation d'images, la source et le puit sont reliés à tous les pixels de l'image. L'ensemble S correspond aux pixels de l'objet tandis que T correspond aux pixels du fond. La capacité des arêtes reliant la source et le puit à un pixel de l'image peut être déterminée par la probabilité que ce pixel appartienne à l'objet ou au fond. La capacité des arêtes reliant les pixels entre eux peut être déterminée par des mesures de similarité fonction de l'intensité, de la couleur ou d'autres caractéristiques. Par exemple, l'algorithme grabCut (Rother, Kolmogorov et Blake 2004) utilise l'initialisation manuelle de l'utilisateur pour attribuer une forte capacité aux arêtes reliant la source aux pixels marqués ("seed" en anglais). La capacité des arêtes pour les pixels non marqués est déterminée à l'aide d'un mélange de gaussiennes. Pour trouver la valeur du flux maximum (la coupe minimum), on peut utiliser l'algorithme "push-relabel" (Goldberg et Tarjan 1988) qui déplace le flux itérativement depuis la source jusqu'au puit en maintenant un "preflow", *i.e.* la quantité de flux entrant dans chaque nœud n'est pas nécessairement égale à la quantité de flux sortant. La méthode de Ford-Fulkerson (Ford et Fulkerson 1957) est

également souvent utilisée pour résoudre le problème de flux maximum. Elle consiste à pousser le flux depuis la source vers le puit tant qu'il existe un chemin les reliant et ayant une capacité positive pour chaque arête.

3.1.5 Modèles de contours actifs et de courbe de niveau

Le modèle de contour actif ou "snake" a été introduit par Kass, Witkin et Terzopoulos 1988. Depuis, ce modèle a été largement réutilisé et adapté pour segmenter des images (Chen, Biffi et al. 2019 ; Yushkevich et al. 2006 ; Gastaud, Barlaud et Aubert 2004). Les modèles de contour actif cherchent à ajuster une courbe paramétrique ou spline aux contours d'un objet d'intérêt dans l'image. La courbe est définie comme un ensemble de n points v_i $i \in [0; n-1]$. Les n points de la courbe sont itérativement déplacés pour s'approcher au mieux des contours de l'objet. Pour ce faire, les modèles par contour actifs reposent sur la minimisation d'une fonction d'énergie comprenant une composante interne E_{internal} et une composante externe. La composante externe dépend des caractéristiques de l'image E_{image} ainsi que de contraintes fixées par l'utilisateur E_{con} :

$$E^*_{\text{snake}} = \int_0^1 E_{\text{snake}}(\mathbf{v}(s))\, ds = \int_0^1 \left(E_{\text{internal}}(\mathbf{v}(s)) + E_{\text{image}}(\mathbf{v}(s)) + E_{\text{con}}(\mathbf{v}(s))\right) ds \tag{3.7}$$

Les contraintes fixées par l'utilisateur peuvent correspondre à la position initiale de la courbe, mais aussi à d'autres contraintes (E_{con}) incluses dans la fonction d'energie et permettant de guider la déplacement de la courbe. La composante interne permet de régulariser le déplacement des points de la courbe en tenant compte à la fois de sa continuité (élasticité du contour) et de sa courbure :

$$E_{\text{internal}} = E_{\text{cont}} + E_{\text{curv}} \tag{3.8}$$

E_{cont} pénalise le déplacement des points de la courbe de façon à ce que celle-ci reste proche de sa forme de départ. Cela permet d'éviter que la courbe s'ajuste au bruit contenu dans l'image. E_{curv} permet de s'assurer que la courbe reste régulière avec peu d'oscillations. E_{internal} peut donc se réécrire à partir de la dérivée première (continuité) et seconde (courbure) de la courbe par rapport au point s :

$$E_{\text{internal}} = \frac{1}{2}\left(\alpha(s)\left\|\frac{d\mathbf{v}}{ds}(s)\right\|^2 + \beta(s)\left\|\frac{d^2\mathbf{v}}{ds^2}(s)\right\|^2\right) \tag{3.9}$$

où $\alpha(s)$ et $\beta(s)$ contrôlent l'importance accordée à l'élasticité et à la courbure respectivement. La composante E_{image} permet d'attirer la courbe vers certaines caractéristiques d'intérêt dans l'image comme les contours, les coins ou encore des zones avec une certaine intensité. Ainsi, la composante E_{image} est généralement définie comme suit :

$$E_{\text{image}} = w_{\text{int}}E_{\text{int}} + w_{\text{grad}}E_{\text{grad}} + w_{\text{coin}}E_{\text{coin}} \tag{3.10}$$

avec w_{int}, w_{grad} et w_{coin} des constantes permettant de pondérer chaque composante. E_{int} représente l'intensité des pixels de l'image. E_{grad} exprime l'intensité des contours et repose sur le calcul du gradient de l'image pour chaque pixel. E_{coin} permet de

s'assurer que la courbe s'ajuste bien aux coins d'un objet. Elle s'appuie sur le calcul des dérivées première et seconde pour détecter les bords de l'objet avec un fort changement dans l'orientation et la courbure. E^*_{snake} peut être minimisée de manière itérative à l'aide de l'algorithme de descente du gradient où les points de la courbe sont déplacés dans la direction opposée du gradient. De nombreuses améliorations du modèle "snake" existent. Une variante importante s'appuie sur les ensembles de niveaux (level set) pour représenter les contours de l'objet d'intérêt comme les points pour lesquels une fonction $\phi(x, y)$ s'annule. Le contour est alors défini comme l'ensemble des points (x, y) tels que $\phi(x, y) = 0$ (Figure 3.1b). L'évolution de l'ensemble de niveaux est alors obtenue par l'équation aux dérivées partielles suivante :

$$\frac{\partial \phi}{\partial t} = F|\nabla \phi| \tag{3.11}$$

où $|\nabla \phi|$ est normale à l'isosurface et t un paramètre de temps artificiel. Cela signifie que le taux de variation de l'ensemble de niveaux par rapport au temps ($\frac{\partial \phi}{\partial t}$) est égal à la magnitude du gradient pondérée par F, où F est une fonction décrivant le taux de variation de l'ensemble de niveaux. F peut être déterminé à partir des caractéristiques locales de l'image comme le gradient, les statistiques de l'objet ou la courbure. Contrairement aux modèles de contours actifs, les méthodes des surfaces de niveau gèrent naturellement les changements de topologie comme la fusion ou la séparation d'objets. En pratique, les méthodes de segmentation d'images par ensemble de niveaux réutilisent souvent le principe de minimisation de la fonction d'énergie des modèles de contour actif (Chung et Vese 2005 ; Chen et Radke 2009 ; Li, Kao et al. 2008) de sorte que l'équation d'évolution de la courbe de niveau se réécrit :

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} \tag{3.12}$$

avec E la fonction d'énergie et ϕ la fonction de niveau ("level set function"). Par exemple, Chan et Vese 1999 ont introduit un algorithme de segmentation populaire qui utilise à la fois la fonction de niveau et le principe de minimisation de fonction d'énergie.

3.2 Segmentation d'images par apprentissage

La très grande majorité des méthodes de segmentation par apprentissage profond utilise les CNN et spécifiquement des architectures de type encodeur/décodeur. Certaines architectures spécifiques ont également été développées dans le cadre des images médicales comme nous le présenterons Section 3.2.2

3.2.1 CNN pour la segmentation

Les réseaux de type "Fully Convolutional Network" (FCN) (Long, Shelhamer et Darrell 2015) sont les premières architectures d'apprentissage profond permettant d'effectuer de la segmentation d'images. Ils sont constitués d'un encodeur de type CNN contenant une série de convolutions et de couches de pooling pour progressivement réduire la résolution de l'image d'entrée. L'encodeur permet d'extraire l'information pertinente en effectuant une compression de l'information contenue dans l'image. La carte de caractéristiques obtenue en sortie du CNN est sur-échantillonnée

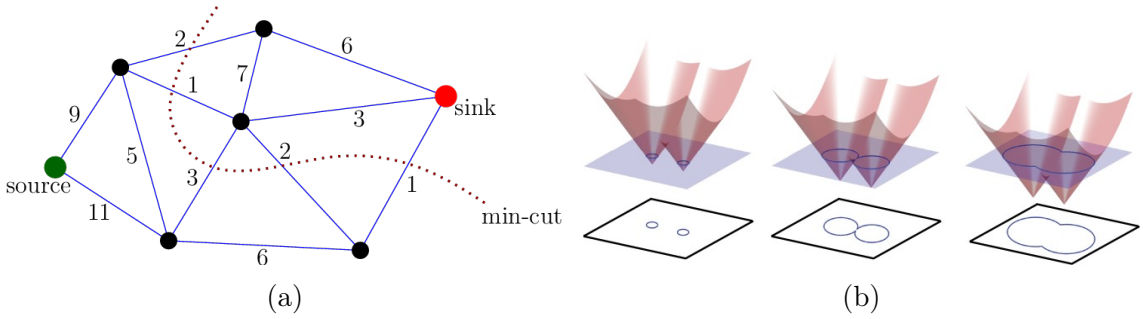


FIGURE 3.1 – (a) Exemple de coupe minimum dans un graphe (Salman 2010). La capacité d'une coupe est la somme des poids des arêtes présentes dans la coupe. (b) Méthodes de courbe de niveau, image issue de (Gibou, Fedkiw et Osher 2018) : les courbes bleues représentent les contours de l'objet à segmenter. Le rectangle bleu est l'isosurface à la surface en rouge qui représente la fonction ϕ décrivant l'objet. On observe que la méthode des ensembles de niveaux gère naturellement la fusion des deux formes (cercles bleus).

à l'aide d'une interpolation bilinéaire pour obtenir le masque de segmentation. La faiblesse de cette architecture réside dans l'utilisation d'un simple sur-échantillonnage bilinéaire pour décoder l'image. Par conséquent, Noh, Hong et Han 2015 introduisent les couches de convolution transposées qui permettent de construire des décodeurs de façon symétrique à l'encodeur et contenant également des paramètres appris par le réseau. Ronneberger, Fischer et Brox 2015 reprennent l'architecture FCN en lui ajoutant un décodeur convolutionnel et nomment leur réseau "U-net" du fait de sa forme en U. Surtout, ils concatènent les cartes de caractéristiques générées à chaque étage de l'encodeur avec les cartes de caractéristiques de même résolution du décodeur. Ce simple changement architectural nommé "skip connection" permet une nette amélioration des performances de segmentation. En effet, les cartes de caractéristiques du décodeur contiennent beaucoup d'informations sémantiques (informations abstraites relatives à la forme des objets ou à leur représentation) mais ont perdu l'information de haute résolution que l'on trouve dans les cartes de caractéristiques les plus proches de l'entrée du réseau. En concaténant ces dernières avec les cartes de caractéristiques du décodeur, le réseau est capable de récupérer les détails spatiaux des objets présents dans l'image (par exemple l'information relative aux contours des objets), améliorant la précision de la segmentation. Néanmoins, la différence d'information sémantique entre les cartes de caractéristiques de l'encodeur et celles du décodeur peut poser problème au réseau lorsque ces dernières sont directement concaténées (Pang et al. 2019 ; Wang, Wang, Zhong et al. 2021). Par la suite, de nombreuses variantes du U-net ont été conçues. U-net++ (Zhou, Siddiquee et al. 2018) introduit des convolutions supplémentaires entre l'encodeur et le décodeur pour réduire l'écart sémantique entres les données. Dans la même optique, MultiResUnet (Ibtehaz et Rahman 2020) ajoute des blocs résiduels sur le chemin des skip connections, ce qui permet de réduire l'écart sémantique plus efficacement en gardant un gradient fort. Linknet (Chaurasia et Culurciello 2017) utilise des connections résiduelles dans l'encodeur et le décodeur pour améliorer le flux du gradient. Attention-U-net (Oktay, Schlemper et al. 2018) ajoute des modules d'attention entre l'encodeur et le décodeur pour permettre au réseau de se concentrer sur l'information pertinente. R2U-net (Alom et al. 2018) rajoute des connections

résiduelles qui permettent d'entrainer des réseaux plus profonds et remplace les convolutions par des convolutions récurrentes afin d'apprendre des caractéristiques plus utiles. 3D U-net (Çiçek et al. 2016) remplace toutes les opérations 2D du U-net par leur équivalent en 3D afin d'obtenir des segmentations volumiques sur des données peu annotées. DeepMedic (Kamnitsas et al. 2016) utilise un 3D U-net avec deux encodeurs prenant chacun en entrée des patchs de résolution différente. Les encodeurs ne sous-échantillonnent pas les patchs et les caractéristiques extraites de ces patchs sont fusionnées en sortie des encodeurs. Cela permet d'obtenir un champ réceptif important malgré l'absence de sous-échantillonnage.

Les travaux ultérieurs se sont concentrés sur l'augmentation du champ réceptif du réseau, identifié comme un facteur clef expliquant la qualité des résultats obtenus. En effet, l'attribution d'une classe aux pixels nécessite de disposer d'un large champ de vision. Pour ce faire, plusieurs travaux (Chen, Papandreou et al. 2016 ; Chen, Papandreou et al. 2018 ; Chen, Zhu et al. 2018 ; Zhao, Shi et al. 2017) ont intégré des convolutions dilatées à leur architecture. Ces convolutions contiennent des trous dans leur fenêtre de convolution permettant, pour un même nombre de paramètres, de couvrir une surface de l'image plus grande. Néanmoins, en introduisant ces trous, le réseau peut avoir plus de mal à remarquer les détails de petites tailles se trouvant dans son champ réceptif et donc, à segmenter des motifs complexes comme certains contours des objets. Par conséquent, Deeplab (Chen, Papandreou et al. 2018) utilise une couche nommée "A-trous Spatial Pyramid Pooling" (ASPP) qui, pour la carte de caractéristiques de plus faible résolution, effectue plusieurs convolutions dilatées en parallèle avec un taux de dilatation croissant et combine la sortie de ces couches par concaténation. De cette façon, le réseau a accès à la fois à un contexte large et à des détails subtils. L'ASPP est présentée Figure 3.2.

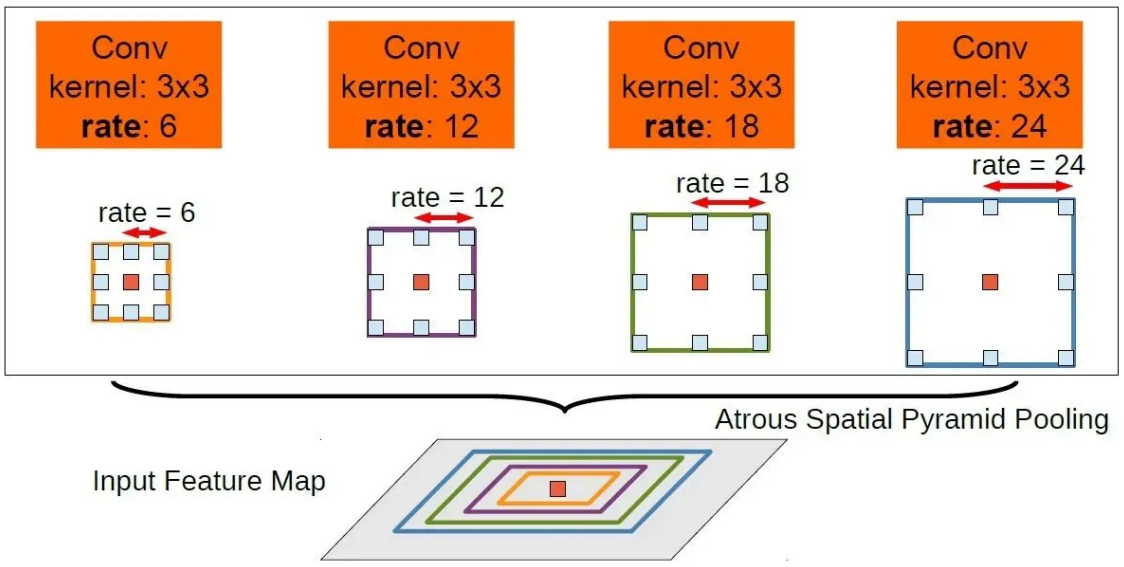


FIGURE 3.2 – A-trous Spatial Pyramid Pooling (ASPP) utilisé par Deeplab (Chen, Papandreou et al. 2018) pour augmenter le champ réceptif tout en gardant accès aux détails de l'image

3.2.2 Deep learning pour la segmentation d'images médicales

La plupart des recherches conduites en imagerie cardiaque par apprentissage profond partent d'une architecture de type U-net et introduisent certaines modifications. Les réseaux neuronaux récurrents spécifiquement adaptés aux vidéos (convLSTM et convGRU) (Shi, Chen et al. 2015 ; Ballas et al. 2016) sont souvent utilisés pour permettre au réseau de prédire des résultats temporellement cohérents sur une séquence d'images (Wang et Zhang 2021 ; Zhang, Icke et al. 2018 ; Smistad et al. 2021 ; Poudel, Lamata et Montana 2016 ; Savioli et al. 2018).
De nombreuses méthodes utilisent le "Spatial Transformer Network" (STN) présenté par Jaderberg et al. 2015 pour recaler les images avant la segmentation (Ni et al. 2023 ; Gong et al. 2022 ; Chartsias et al. 2020 ; Vigneault et al. 2018). L'idée est d'obtenir une orientation adaptée ou de corriger certains artefacts pour faciliter la tâche du réseau de segmentation.
Un autre axe majeur de recherche est l'utilisation d'images cardiaques ayant des orientations différentes en entrée du réseau. On cherche ainsi à tirer parti d'informations complémentaires extraites de ces orientations (Zhao, Hu et al. 2022 ; Galazis et al. 2022 ; Li, Wang, Zhang et al. 2020 ; Chen, Biffi et al. 2019).

Les architectures multi-tâches ont également montré leur efficacité pour la segmentation cardiaque. Ces architectures reposent sur l'apprentissage de plusieurs tâches afin d'améliorer la capacité de généralisation du réseau à partir de peu de données. En effet, à la suite des travaux de Myronenko 2019, certains auteurs ont intégré une branche de reconstruction en plus de la branche de segmentation pour introduire davantage de régularisation (Chang et al. 2022 ; Habijan et al. 2021). L'apprentissage multi-tâches est également utilisé pour combiner segmentation et classification (Chen, Bai et Rueckert 2019 ; Dangi, Yaniv et Linte 2019 ; Peng et al. 2021 ; Zhang, Karanikolas et al. 2018) ainsi que segmentation et estimation de mouvement (Zhao, Feng et al. 2020 ; Qin, Bai et al. 2018 ; Xue et al. 2022). D'autres approches pré-entrainent un réseau de reconstruction des annotations de segmentation et utilisent ce réseau de façon à obtenir une fonction de coût supplémentaire pour régulariser le réseau de segmentation. L'idée est d'introduire des contraintes de formes ("shape constraints") de sorte que les prédictions du réseau tiennent compte de l'anatomie des patients (Yue et al. 2019 ; Oktay, Ferrante et al. 2018 ; Chen, Lyu et al. 2023 ; Brahim et al. 2021).

Plusieurs travaux ont montré l'intérêt d'introduire des modules d'attention au sein de toutes les couches convolutionnelles du réseau afin de rééchelonner les canaux des cartes de caractéristiques (Q. Wang et al. 2020 ; Woo et al. 2018 ; Hu, Shen et al. 2019). L'idée est de passer de la carte de caractéristiques de taille $C \times H \times W$ à une carte d'attention de taille $C \times 1 \times 1$ à l'aide d'une couche d'average ou max pooling. Celle-ci permet d'identifier les canaux les plus importants et de déterminer des poids d'attention par l'intermédiaire d'une fonction sigmoïde. Ces poids sont alors utilisés pour rééchelonner les cartes de caractéristiques initiales. Le principe est décrit Figure 3.3. Ces modules permettent au réseau d'apprendre l'interdépendance entre les caractéristiques afin de les recalibrer de façon dynamique. Le réseau peut ainsi se concentrer sur les caractéristiques les plus informatives et réduire l'influence des caractéristiques bruitées. Ces modules consomment peu de ressources mémoire et peuvent être intégrés facilement au sein de blocs résiduels. Ils sont aussi souvent uti-

lisés en conjonction de modules d'attention spatiale et ont été appliqués avec succès à la segmentation cardiaque (Qayyum et al. 2020 ; Arega et Bricq 2020 ; Liu et Yang 2019 ; Chen, Zhou et al. 2022).

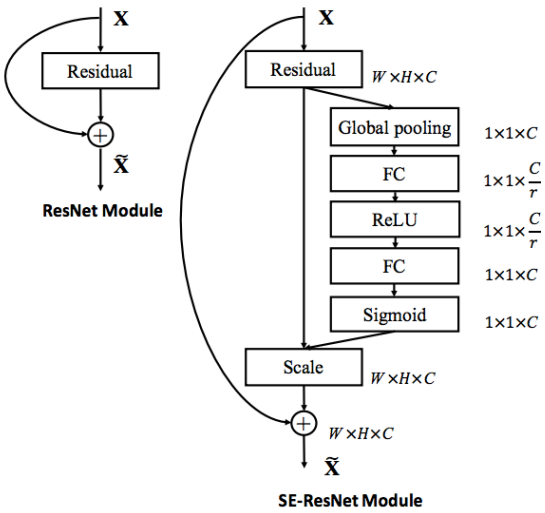


FIGURE 3.3 – Le module "Squeeze and excitation" (Hu, Shen et al. 2019) permettant de porter une attention différente à chaque canal des cartes de caractéristiques.

Plus récemment, les transformers ont été largement utilisés pour la segmentation d'images du cœur. Ces architectures sont le plus souvent intégrées au sein d'un U-net. Du fait de l'absence de larges jeux de données publics, il est difficile de pré-entrainer les transformers de façon efficace pour la segmentation. Par conséquent, la plupart des travaux emploient soit un nombre limité de couches transformer à la plus basse résolution (Chen, Lu et al. 2021 ; Wang, Wang, Liang et al. 2022 ; Xu, Wu et al. 2021), soit des versions alternatives du transformer capables d'effectuer de l'attention aux plus hautes résolutions grâce à une consommation mémoire réduite. Le Swin Transformer (Z. Liu et al. 2021 ; Liu, Ning et al. 2022) a, par exemple, été souvent utilisé pour la segmentation 3D (Grzeszczyk, Płotka et Sitek 2022 ; Maurya et al. 2022 ; Tang, Yang et al. 2022) ou 2D (Fu et al. 2022 ; Yang, Liu et Liang 2024 ; A et al. 2023) car, en effectuant l'attention au sein de fenêtres de petite taille, il rend possible l'utilisation de la self-attention sur des cartes de caractéristiques de haute résolution. Le principe d'attention de Swin est décrit Figure 3.4. D'autres approches modifient le processus de self-attention afin d'en réduire la complexité, permettant ainsi d'utiliser ces blocs d'attention à plusieurs niveaux de l'encodeur et du décodeur. Ainsi, certaines approches emploient des convolutions avec stride ou des projections linéaires pour réduire le nombre de tokens devant être traités par la couche de self-attention (Gao, Zhou et Metaxas 2021 ; Huang, Deng et al. 2021). D'autres ont recours à un processus d'attention linéaire qui calcule l'attention entre la key et la value plutôt qu'entre la query et la key et utilise une fonction de normalisation afin d'approximer la fonction softmax (Fan et al. 2023 ; Azad et al. 2022).

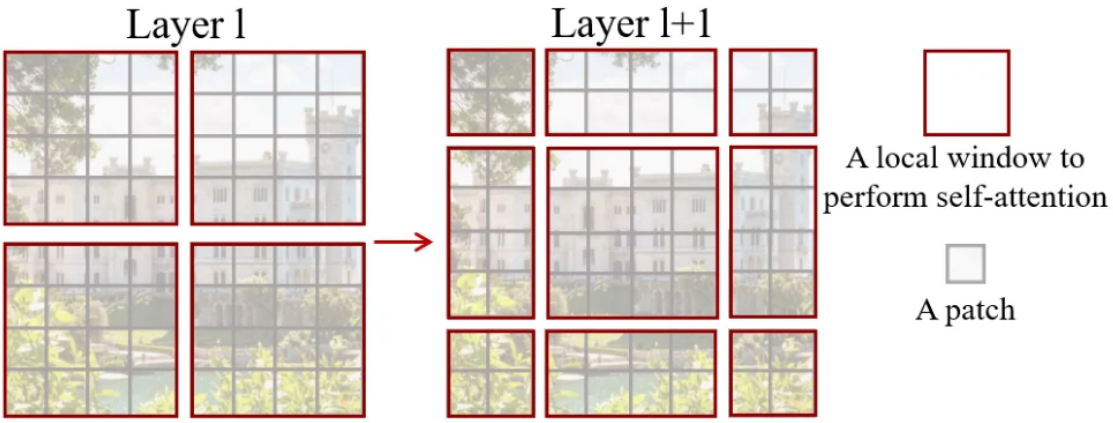


FIGURE 3.4 – Swin Transformer (Z. Liu et al. 2021) effectue l'attention au sein de fenêtres puis décale ces fenêtres et effectue l'attention une nouvelle fois au sein des fenêtres décalées.

3.3 Mesures de performance des segmentations

Nous présentons dans cette section les principales métriques utilisées pour évaluer la qualité des résultats de segmentation. Ces métriques permettent de mesurer la similarité ou la distance entre une segmentation binaire prédite X et la segmentation de vérité terrain Y.

Le score de Dice

Le score de Dice se définit de la façon suivante :

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{3.13}$$

Dans le cadre de la classification binaire, le score de Dice se définit comme suit :

$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.14}$$

où TP sont les vrais positifs, *i.e.* les pixels prédits comme appartenant à l'objet et qui appartiennent effectivement à l'objet. FP sont les faux positifs, *i.e.* les pixels prédits comme appartenant à l'objet alors qu'ils n'en font pas parti. FN sont les faux négatifs, *i.e.* les pixels prédits comme n'appartenant pas à l'objet alors qu'ils appartiennent en réalité à l'objet. Le score de Dice est compris entre 0 et 1 où une valeur de 1 indique une segmentation parfaite.

L'indice de Jaccard (IOU)

On peut mesurer la qualité d'une segmentation binaire à l'aide du score de Jaccard J, définit comme le rapport de l'intersection sur l'union entre la segmentation prédite et la vérité terrain :

$$J(Y, X) = \frac{|Y \cap X|}{|Y \cup X|} = \frac{|Y \cap X|}{|Y| + |X| - |Y \cap X|} = \frac{TP}{TP + FP + FN} \tag{3.15}$$

Comme le score de Dice, l'indice de Jaccard est compris entre 0 et 1. Par ailleurs, on a la relation suivante avec le score de Dice :

$$J = \frac{DSC}{2 - DSC} \tag{3.16}$$

Par conséquent, le score de Dice et l'indice de Jaccard possèdent les mêmes propriétés. En particulier, ces mesures ne tiennent pas compte de la distance entre les pixels ce qui les rend moins appropriées que les mesures de distance lorsque l'objectif de la segmentation est d'estimer avec précision les contours des structures. En outre, puisque ces métriques intègrent le nombre total de pixels appartenant à l'objet au dénominateur, elles sont sensibles à la taille de l'objet segmenté. En effet, une erreur de segmentation de quelques pixels pour un petit objet diminuera le score de Dice et l'indice de Jaccard de façon plus prononcé que pour un objet plus large. Il s'agit d'un inconvénient des mesures reposant sur le calcul du "chevauchement" (overlap). En effet, on peut considérer qu'il serait approprié d'attribuer un meilleur score à la segmentation d'un petit objet lorsque le pourcentage de pixels correctement classifié est le même que pour un grand objet.

Le score de Dice et l'indice de Jaccard donnent un score de 0 lorsque TP vaut 0. Autrement dit, quelle que soit la distance entre la segmentation prédite et la segmentation de vérité terrain, ces mesures rapportent la même valeur lorsqu'il n'y a pas d'intersection entre ces deux masques de segmentations. Dans le cas d'absence d'intersection le score de Dice et l'indice de Jaccard sont donc moins informatifs que les mesures de distance.

La distance de Hausdorff

La distance de Hausdorff (HD) rapporte la distance maximale parmi les distances entre les points d'un ensemble non-vide X et le point le plus proche d'un autre ensemble non-vide Y. Dans le cadre de la segmentation, X et Y sont les ensembles composés, respectivement, des points de contour de l'objet prédits par le modèle et les véritables points de contours de cet objet. Mathématiquement, la distance de Hausdorff est donc définie comme suit :

$$HD(X, Y) := \max \left\{ \sup_{x \in X} d(x, Y),\ \sup_{y \in Y} d(y, X) \right\} \tag{3.17}$$

avec $d(a, B) := \inf_{b \in B} d(a, b)$ et $d(a, b)$ la distance euclidienne entre les points $a \in A$ et $b \in B$. Plutôt que de rapporter la distance maximale, il est également possible de rapporter la plus petite distance supérieure à 95% des distances, on parle alors de distance de Hausdorff 95 (HD95). Cela permet d'éviter d'avoir une mesure trop sensible aux cas extrêmes.

Average Symmetric Surface Distance

L'Average Symmetric Surface Distance (ASSD) rapporte la moyenne des distances entre tous les points de X et le point le plus proche de Y. Mathématiquement, en reprenant les notations de la partie 3.3 on a donc :

$$\text{ASSD}(X, Y) = \frac{1}{|X| + |Y|} \left(\sum_{x \in X} d(x, Y) + \sum_{y \in Y} d(y, X) \right) \tag{3.18}$$

La Figure 3.5 présente la différence entre la HD, HD95 et l'ASSD.

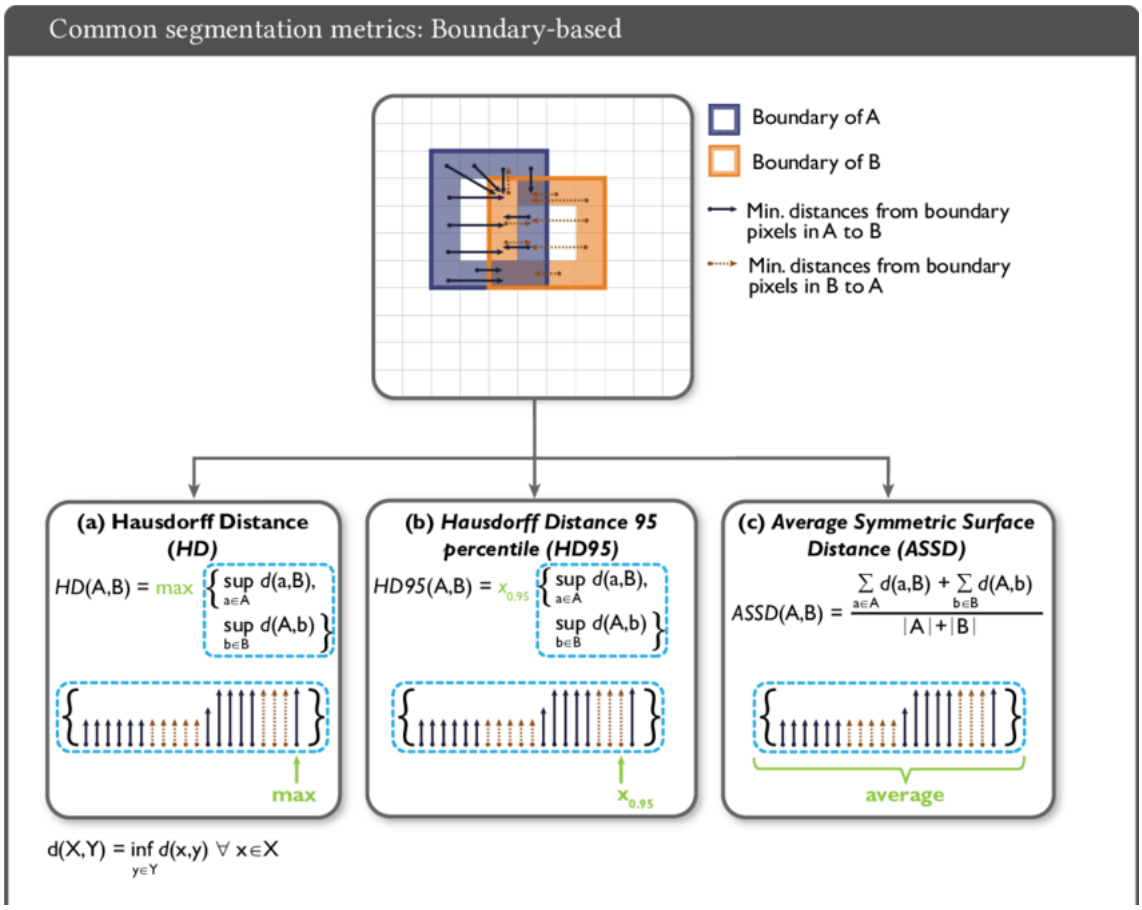


FIGURE 3.5 – Méthode de calcul de la HD, HD95 et ASSD (Reinke et al. 2022)

3.4 positionnement

De nos jours, de nombreux algorithmes de segmentation reposent sur l'apprentissage profond, et notamment l'architecture U-net (Ronneberger, Fischer et Brox 2015) (section 3.2) composée d'un encodeur, d'un décodeur et de "skip-connections" entre l'encodeur et le décodeur. Ce design fusionne efficacement les informations haute résolution de l'encodeur avec les caractéristiques sémantiquement riches du décodeur. Cependant, une simple concaténation des cartes de caractéristiques de l'encodeur et du décodeur s'est révélée sous-optimale (Wang, Wang, Zhong et al. 2021 ; Cao, Ma et al. 2022 ; Zhou, Siddiquee et al. 2018 ; Pang et al. 2019 ; Wang, Cao et al. 2021). En effet, les cartes de caractéristiques de l'encodeur avec des détails locaux très précis, contiennent moins de caractéristiques sémantiquement riches que les cartes de caractéristiques du décodeur et sont généralement plus bruitées. A l'inverse, les cartes de caractéristiques en provenance du décodeur sont très riches sémantiquement mais très peu précises au niveau des contours des formes. Ce phénomène est connu sous le nom d'écart sémantique. Il est donc intéressant de réutiliser les cartes de caractéristiques de l'encodeur dans le décodeur car cela permet de réintroduire des informations de haute résolution.

Traditionnellement, des convolutions ont été utilisées pour essayer de combler cet écart sémantique (Wang, Chen et al. 2022 ; Wang, Wang, Zhong et al. 2021 ; Ibtehaz et Rahman 2020) car elles permettent d'augmenter progressivement le champ réceptif. Certains mécanismes de filtrage ont également été mis en œuvre pour fil-

trer les caractéristiques transmises au décodeur en se basant sur les caractéristiques provenant de tous les niveaux de l'encodeur (Li, Zhao et al. 2020), ou seulement du niveau suivant (Islam et al. 2017). La plupart des méthodes qui ont spécifiquement tenté de réduire l'écart sémantique utilise encore des convolutions ou repose sur la self-attention. Les rares méthodes qui utilisent la cross-attention s'appuient soit sur des couches transformer entières (Wang, Cao et al. 2021 ; Peiris et al. 2021) soit utilisent une attention spatiale complète (Petit et al. 2021), qui sont toutes deux coûteuses en termes de calcul.

En conséquence, ce travail se concentre sur la conception d'une nouvelle architecture de réseau capable de mieux gérer l'écart sémantique entre les données de l'encodeur et du décodeur. Des travaux récents (Wang, Cao et al. 2021 ; Petit et al. 2021) ont montré que les mécanismes de cross-attention semblent être la solution idéale pour déterminer, à partir des données sémantiquement riches du décodeur, où porter l'attention pour obtenir des informations de localisation précises dans les données de l'encodeur. Cependant, ces méthodes sont très coûteuses en termes de nombre de paramètres et sont donc sujettes à une mauvaise généralisation lorsque le nombre de données est limité. L'architecture proposée surmonte ce problème et améliore les performances de segmentation évaluées sur des ensembles de données publics et privés en effectuant l'attention au sein de fenêtres plus petites. Afin d'évaluer les forces et les faiblesses de la méthode, nous avons réalisé une étude détaillée des résultats de segmentation. Sur le plan clinique, nous montrons que la méthode proposée améliore également la précision des indices quantitatifs volumiques nécessaires pour diagnostiquer les maladies cardiovasculaires. Nous avons également étudié la capacité de généralisation de l'algorithme sur de nouvelles bases de données (données hors distribution) pour vérifier que la méthode pourrait être utilisée dans différents centres, sans nécessiter de réapprentissage.

La première partie de cette étude présente l'architecture du réseau et comprend une explication détaillée des Swin Filtering Blocks (SFB) utilisés pour combler l'écart sémantique. Cette section décrit également les données et le protocole suivi pour les expériences. Ensuite, les résultats obtenus avec cette architecture de réseau sont présentés : une étude ablative, une comparaison avec la littérature, l'étude des performances de généralisation et l'apport de l'entrainement sur toutes les images du cycle cardiaque. Enfin, nous discutons nos résultats avant de conclure.

Chapitre 4

Segmentation d'IRM cardiaque par réseau de neurones avec attention pour l'estimation de biomarqueurs

4.1 Méthode

Cette section détaille nos principales contributions, le jeu de données utilisé et les validations effectuées comme résumé Figure 4.1 sous forme de diagramme en blocs.

4.1.1 Architecture du réseau

Le réseau proposé, appelé SFB-net (Swin Filtering Block network), utilise l'architecture U-net (Ronneberger, Fischer et Brox 2015) avec un encodeur, un décodeur et des "skip-connections" entre les deux, comme illustré Figure 4.2. Des blocs convolutionnels, représentés en bleu, ont été utilisés tout au long du réseau et ont été doublés dans l'encodeur par rapport au décodeur pour améliorer la capacité d'encodage du modèle (Myronenko 2019). Ces blocs contiennent 2 convolutions, chacune suivie d'une couche de "batch-normalization" et d'une activation Gelu (Gaussian Error Linear Unit (Hendrycks et Gimpel 2023)). Nous utilisons Gelu plutôt que Relu dans tout le réseau pour rester cohérent avec la couche transformer qui est généralement implémentée avec cette fonction d'activation. De plus, Gelu ne souffre pas de l'effet "dying Relu" où l'activation donne toujours des sorties nulles pour les entrées négatives, empêchant le réseau d'ajuster ses poids (Lu, Shin et al. 2020 ; Arnekvist et al. 2020) (section 2.2.1). Le nombre de filtres a été doublé à chaque couche de l'encodeur et divisé par deux pour les couches correspondantes du décodeur. De plus, nous proposons d'utiliser des convolutions avec stride au lieu de couches de pooling pour réduire la dimension des cartes de caractéristiques, car cela permet au réseau d'effectuer cette opération plus facilement avec des paramètres apprenables. Le nombre de sous-échantillonnages est faible, avec une carte de caractéristiques 8 fois plus petite que la taille de l'image en entrée pour la résolution la plus faible. Le sur-échantillonnage a été réalisé à l'aide de convolutions transposées 2D. Pour compenser la faible profondeur du réseau, ce qui pourrait réduire son champ réceptif, nous introduisons une couche transformer conventionnelle au point le plus étroit (représentée en violet). Cela permet au réseau de tirer parti de l'information contextuelle globale. Les couches transformers n'ont pas été utilisées dans l'encodeur et le

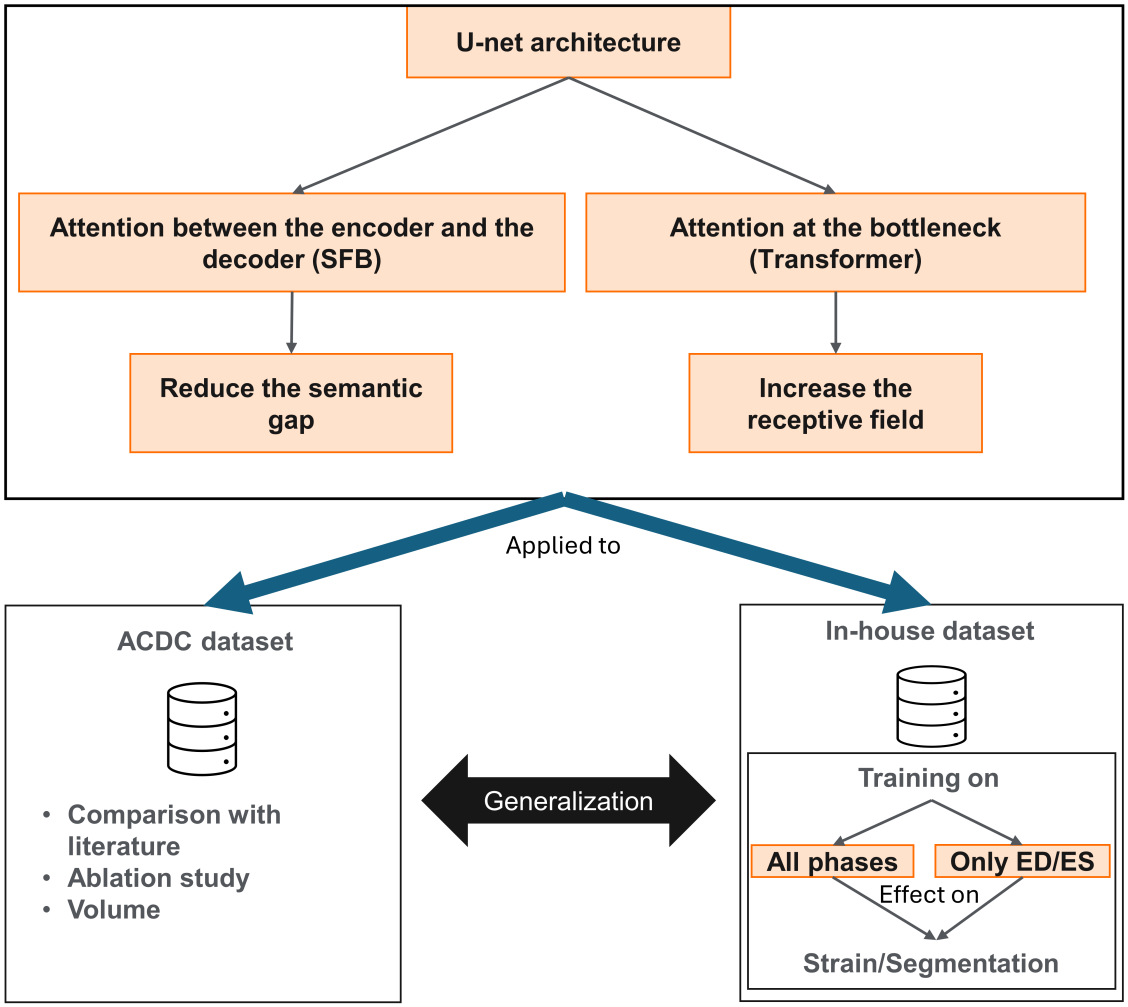


FIGURE 4.1 – Diagramme en blocs de l'approche proposée. Notre réseau est testé sur deux ensembles de données et nous évaluons les conséquences de l'entraînement uniquement avec les images des phases ED et ES par rapport à toutes les images du cycle cardiaque. SFB : Swin Filtering Block.

décodeur, car il a été démontré que les convolutions à des résolutions plus élevées conduisaient à de meilleures performances (Wang, Xie, Lin et al. 2022 ; Xiao et al. 2021 ; Dai, Liu et al. 2021). En effet, les convolutions généralisent mieux ce qui a été appris aux images non vues que les transformers et extraient plus efficacement l'information locale se situant à des résolutions plus élevées. Un résumé de l'architecture de notre réseau se trouve en annexe (tableau 1).

La deep-supervision a été appliquée à chaque niveau du décodeur. Plus précisément, les segmentations de vérité terrain sont sous-échantillonnées pour correspondre aux résolutions à chaque niveau du décodeur. Les poids de la fonction de coût, $\alpha_i \forall i \in 0, 1, 2$ pour chaque résolution, sont divisés par deux lorsque la taille de l'image est sous-échantillonnée et normalisés de sorte que la somme des poids soit égale à 1. La fonction de coût finale est la somme pondérée des erreurs obtenues à chaque niveau du décodeur et est définie comme suit :

$$\mathcal{L} = \alpha_1 \times \mathcal{L}_{H,W} + \alpha_2 \times \mathcal{L}_{\frac{H}{2},\frac{W}{2}} + \alpha_3 \times \mathcal{L}_{\frac{H}{4},\frac{W}{4}} \tag{4.1}$$

Où H et W représentent respectivement la hauteur et la largeur de l'image d'entrée.

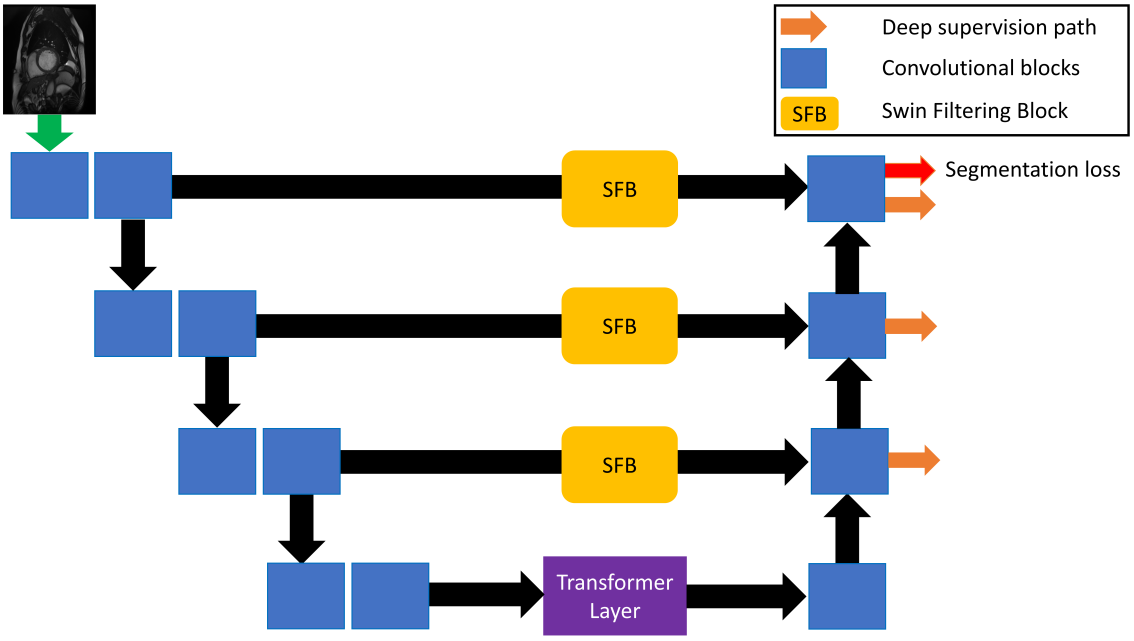


FIGURE 4.2 – Représentation de SFB-net. La deep-supervision est utilisée dans le décodeur. Les blocs convolutionnels sont représentés en bleu, les Swin Filtering blocks (SFB) en jaune, et la couche transformer conventionnelle en violet.

Avec :

$$\alpha_i = \frac{\frac{1}{2^{i-1}}}{\sum_{j=1}^{3} \alpha_j} \quad \forall i \in \{1, 2, 3\} \tag{4.2}$$

Pour une comparaison équitable avec les études précédentes (Zhou, Guo et al. 2022 ; Isensee et al. 2018), la fonction de coût $\mathcal{L}$ à chaque niveau de résolution est définie comme la somme de l'entropie croisée et du critère de Dice :

$$\mathcal{L} = L_{\text{dice}} + L_{\text{CE}} \tag{4.3}$$

avec

$$L_{\text{dice}} = -\frac{2}{|K|} \sum_{k \in K} \left(\frac{\sum_{i \in I} u_i^k v_i^k}{\sum_{i \in I} u_i^k + \sum_{i \in I} v_i^k} \right) \tag{4.4}$$

$$L_{\text{CE}} = -\frac{1}{|I|} \sum_{i \in I} \sum_{k \in K} v_i^k \log(u_i^k) \tag{4.5}$$

où u est la sortie softmax du réseau et v est un codage one-hot de la carte de segmentation de vérité terrain. u et v ont tous les deux une forme $I \times K$ avec $i \in I$ représentant le nombre de pixels dans le patch/batch d'entraînement et $k \in K$ représentant les classes.

4.1.2 Swin Filtering Blocks

Nous proposons également d'introduire un mécanisme de filtrage au niveau des skip connections entre l'encodeur et le décodeur, comme illustré par les blocs jaunes Figure 4.2 et décrit plus en détails Figure 4.3. L'objectif est de permettre au décodeur de filtrer les informations non pertinentes provenant de l'encodeur. Plus précisément, les cartes de caractéristiques de l'encodeur contiennent du bruit qui pourrait être

éliminé avant la concaténation avec les cartes de caractéristiques du décodeur. Pour ce faire, les informations locales contenues dans les cartes de caractéristiques haute résolution de l'encodeur et situées dans les zones soulignées par des cartes de caractéristiques sémantiquement riches du décodeur ont été mises en avant, tandis que les réponses dans les autres zones bruitées ont été atténuées. Nous avons utilisé un processus d'attention au sein de fenêtres et de fenêtres décalées (Liu, Lin et al. 2021) plutôt que la Multi-Head-Self-Attention (MHSA), (Vaswani et al. 2017), de façon à réduire la consommation mémoire et le temps d'entrainement. La "Windowed Multi Head Self Attention" (W-MHSA) réalise une attention au sein de fenêtres de M par M patchs. La "Shifted Window Multi-Head Self Attention" (SW-MHSA), décale les fenêtres de $\lfloor \frac{M}{2} \rfloor$ patchs dans les directions x et y afin que l'attention puisse être réalisée entre des patchs appartenant à différentes fenêtres. La W-MHSA se définit comme suit :

$$\text{W-MHSA}\left(Q, K, V\right) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V \tag{4.6}$$

où Q, K et $V \in \mathbb{R}^{M^2 \times d}$ sont respectivement la query, key et value de dimension d, et $B \in \mathbb{R}^{M^2 \times M^2}$ est le biais de position relative apprenable ajouté à chaque tête qui encode la position relative entre les patchs. Q, K et V sont des tenseurs générés à partir d'une carte de caractéristiques F à l'aide de projections linéaires distinctes comme suit :

$$Q_F = FA_Q^T + b_Q \tag{4.7}$$

$$K_F = FA_K^T + b_K \tag{4.8}$$

$$V_F = FA_V^T + b_V \tag{4.9}$$

avec $b_i \in \mathbb{R}^d$ le biais et $A_i \in \mathbb{R}^{d \times d}$ une matrice de poids pour $i \in \{Q, K, V\}$. Q_F, K_F et V_F sont la query, key et value générées à partir de la carte de caractéristiques F. Les blocs W-MHSA et SW-MHSA sont utilisés pour réaliser la cross-attention entre les cartes de caractéristiques de l'encodeur et du décodeur. La cross-attention utilise le même procédé que la self-attention mais avec la query, key et value provenant de cartes de caractéristiques différentes. Étant donné que les cartes de caractéristiques provenant de l'encodeur sont redimensionnées en fonction de celles du décodeur, la value est choisie comme provenant de l'encodeur tandis que la query et la key proviennent du décodeur :

$$\text{CA}_{\text{out}} = \text{SW-MHSA}\left(Q_{F_{\text{dec}}}, K_{F_{\text{dec}}}, \text{W-MHSA}\left(Q_{F_{\text{dec}}}, K_{F_{\text{dec}}}, V_{F_{\text{enc}}}\right)\right) \tag{4.10}$$

Où F_{dec} et F_{enc} sont les cartes de caractéristiques provenant respectivement du décodeur et de l'encodeur. Le résultat est passé à une fonction sigmoïde σ pour générer des poids w variant de 0 à 1 utilisés pour redimensionner la carte de caractéristiques de l'encodeur :

$$w = \sigma\left(\text{conv}\left(\text{CA}_{\text{out}}\right)\right) \tag{4.11}$$

où conv est une convolution standard avec un noyau de taille 1 par 1 suivie d'une couche de batch-normalisation. Enfin, la carte de caractéristiques de l'encodeur redimensionnée F_{out} est obtenue en appliquant le produit de Hadamard $\odot$ entre les poids calculés w et la carte de caractéristiques originale de l'encodeur :

$$F_{\text{out}} = F_{\text{enc}} \odot w \tag{4.12}$$

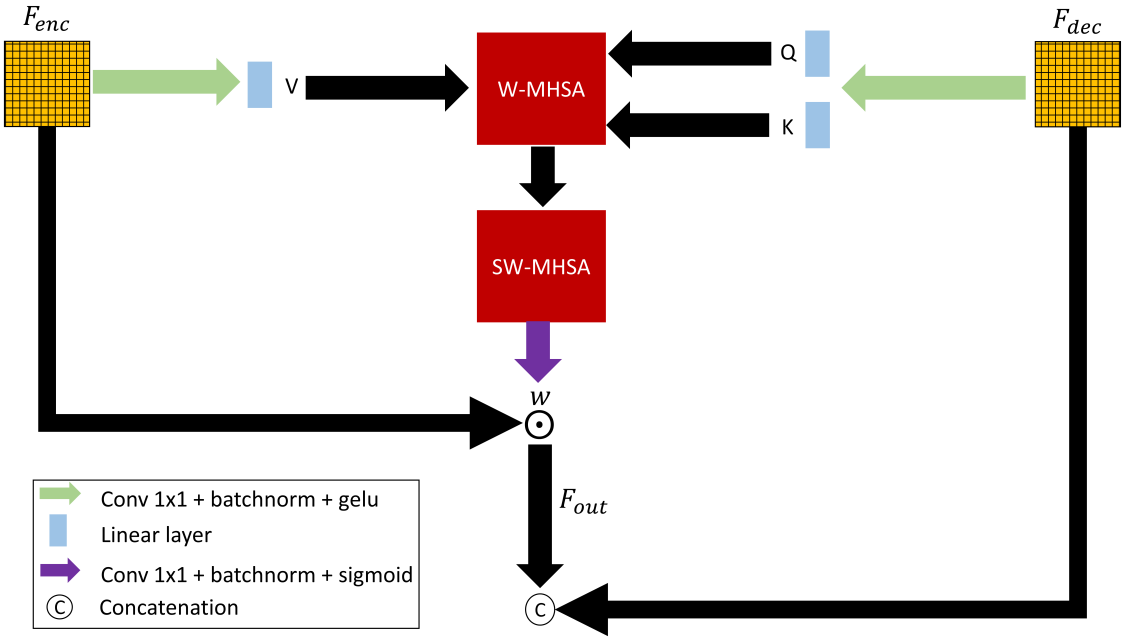


FIGURE 4.3 – Représentation schématique des Swin Filtering Blocks (SFB) utilisés entre l'encodeur et le décodeur. w sont les poids obtenus par la cross-attention et utilisés pour filtrer la carte de caractéristiques de l'encodeur. W-MHSA : Windowed Multi Head Self Attention, SW-MHSA : Shifted Window Multi-Head Self Attention.

4.1.3 Données IRM et population étudiée

Les résultats suivants sont obtenus à partir des jeux de données ACDC (section 2.5.2) et Quorum (section 2.5.5). La version de Quorum traité par le logiciel Medis est utilisé pour l'étude de la capacité de généralisation du réseau (section 4.2.4). La version traité par CardioTrack est employé pour évaluer les conséquences d'un entrainement sur toutes les phases du cycle cardiaque (section 4.2.5).

4.1.4 Détails d'implémentation

Pour le jeu de données ACDC, une validation croisée à 5 folds a été utilisée pour évaluer les performances du modèle. Pour le jeu de données Quorum, nous utilisons 80 % des patients pour l'entraînement et 20 % pour les tests. 20 % des patients de l'ensemble d'entraînement sont utilisés pour la validation. Le nombre de patients dans les ensembles d'entraînement et de test est fourni Tableau 4.1.

TABLE 4.1 – Nombre de patients et d'images dans les ensembles d'entraînement et de test pour ACDC et Quorum. Pour ACDC, le nombre moyen de coupes 2D par fold est rapporté.

Dataset	**Train (# images)**	**Test (# images)**
ACDC (1 fold)	80 (1521.6)	20 (380.4)
Quorum (section 4.2.4)	124 (2629)	39 (780)
Quorum (section 4.2.5)	174 (22743)	54 (6375)

SFB-net est implémenté avec PyTorch et entraîné sur une Tesla V100 SXM2 de 16 Go. Le framework nnU-net (Isensee et al. 2018) est utilisé comme point de

départ pour ce travail. L'optimiseur AdamW et un rythme de décroissance cosinus sont utilisés pour l'entraînement. Le pas d'apprentissage initial et le weight decay sont tous deux fixés à 0,0001. Pour une comparaison équitable avec d'autres études (Zhou, Guo et al. 2022 ; Isensee et al. 2018), le nombre d'epochs d'entraînement est fixé à 1000. Chaque epoch est constituée de 250 itérations (échantillonnage aléatoire des images d'entraînement pour former un batch. 250 est la valeur par défaut du framework nnU-net). La taille du batch est de 10 pour ACDC et de 6 pour le jeu de données interne. Les poids du modèle après la dernière epoch sont sélectionnés pour l'inférence.
Nous avons conservé les prétraitements de nnU-net : avant l'entraînement, toutes les images sont rééchantillonnées pour avoir une taille de pixel égale à la médiane sur le jeu de données. Comme nous utilisons un réseau 2D, aucun rééchantillonnage n'est effectué le long de l'axe z du volume. Les images 2D sont rognées au centre pour avoir une taille de 224x224 pixels pour ACDC et de 288x288 pixels pour le jeu de données Quorum. Avant de les passer au réseau, les images sont normalisées pour qu'elles aient une moyenne de 0 et un écart-type de 1. Un large éventail d'augmentations de données a été appliqué dynamiquement pendant l'entraînement pour les jeux de données ACDC et Quorum : rotation, mise à l'échelle, ajustement gamma, ajustement de la luminosité, mirroring, modification du contraste, simulation de basse résolution, bruit et flou. Plus d'informations sur l'augmentation de données sont disponibles en annexe .1.1. L'augmentation des données au moment du test est aussi utilisée car c'est le paramètre par défaut de nnU-net.

4.2 Résultats

4.2.1 Résultats et étude ablative

Les résultats en termes de score de Dice, d'intersection sur l'union (IOU), de distance de Hausdorff (HD) et de distance de surface symétrique moyenne (ASSD) pour le jeu de données ACDC sont présentés dans le Tableau 4.2. Nous avons également mené une étude ablative pour évaluer de façon plus précise l'apport des différents composants de SFB-net. Celui-ci a été comparé à trois variantes :

— SFB-net sans SFBs (SFB).
— SFB-net sans transformer : SFB-net où la couche transformer au bottleneck est remplacée par une convolution avec un noyau de taille 3×3 pixels (512 filtres en entrée, 512 en sortie).
— SFB-net sans d-s : SFB-net sans deep-supervision.

TABLE 4.2 – Etude ablative sur ACDC.
Étude ablative réalisée sur le jeu de données ACDC. Le score de Dice, le score d'Intersection sur l'Union (IOU), la distance de surface symétrique moyenne (ASSD) et la distance de Hausdorff (HD) sont obtenus avant post-traitement sur 5 folds et rapportés comme moyenne ± écart-type. Les valeurs en gras correspondent aux meilleures performances pour la structure cardiaque. SFB = Swin Filtering Block, d-s = deep-supervision. Les p-value rapportées se réfèrent à la comparaison entre le SBF-net de référence et la méthode de la ligne courante. Elles sont calculées à l'aide du test de rangs signés de Wilcoxon.

	Modèles	Moyenne	VG	MYO	VD
Dice	SFB-net	**92.45 ± 3.15**	**95.04 ± 4.29**	**90.83 ± 2.99**	91.50 ± 6.45
	SFB-net sans SFBs	92.42 ± 3.31 p=0.1140	94.86 ± 4.65 p=0.1224	90.77 ± 2.90 p=0.2114	91.63 ± 6.47 p=0.3008
	SFB-net sans transformer	92.26 ± 3.35 p=0.0159	94.75 ± 4.62 p=0.0825	90.62 ± 3.20 p=0.0011	91.41 ± 6.47 p=0.3870
	SFB-net sans d-s	92.41 ± 3.22 p=0.3236	94.74 ± 4.86 p=0.2544	90.73 ± 3.12 p=0.8079	**91.75 ± 5.86** p=0.8544
IOU	SFB-net	**86.36 ± 4.98**	**90.83 ± 6.96**	**83.33 ± 4.86**	84.91 ± 9.80
	SFB-net sans SFBs	86.29 ± 5.16 p=0.1093	90.54 ± 7.37 p=0.1195	83.22 ± 4.72 p=0.1966	85.12 ± 9.70 p=0.3031
	SFB-net sans transformer	86.04 ± 5.28 p=0.0147	90.35 ± 7.34 p=0.0845	82.99 ± 5.17 p=0.0012	84.77 ± 9.91 0.3783
	SFB-net sans d-s	86.26 ± 5.11 p=0.3165	90.35 ± 7.69 p=0.2565	83.18 ± 5.06 p=0.8041	**85.25 ± 9.09** p=0.8649
ASSD (mm)	SFB-net	**0.55 ± 0.49**	**0.46 ± 0.75**	**0.43 ± 0.30**	0.76 ± 1.00
	SFB-net sans SFBs	0.56 ± 0.55 p=0.1357	0.51 ± 0.84 p=0.4581	0.45 ± 0.35 p=0.0986	0.72 ± 0.99 p=0.5231
	SFB-net sans transformer	0.60 ± 0.51 p=0.0128	0.54 ± 0.72 p=0.0718	0.50 ± 0.47 p=0.0006	0.76 ± 0.98 p=0.3964
	SFB-net sans d-s	0.58 ± 0.56 p=0.0928	0.55 ± 0.93 p=0.3230	0.48 ± 0.46 p=0.3147	**0.70 ± 0.85** p=0.2721
Hausdorff distance (mm)	SFB-net	**9.24 ± 6.29**	**6.73 ± 7.48**	**7.99 ± 8.02**	**12.99 ± 11.04**
	SFB-net sans SFBs	10.22 ± 8.13 p=0.1037	7.78 ± 9.78 p=0.2465	8.66 ± 9.44 p=0.2332	14.23 ± 13.96 p=0.4159
	SFB-net sans transformer	11.49 ± 10.96 p=0.0350	9.18 ± 14.15 p=0.0419	11.51 ± 17.66 p=0.0176	13.78 ± 11.63 p=0.0755
	SFB-net sans d-s	10.55 ± 8.91 p=0.0586	8.13 ± 10.66 p=0.0945	9.21 ± 11.32 p=0.3231	14.30 ± 13.93 p=0.0209

Les trois variantes d'ablation ont environ 21 millions de paramètres. Cette étude ablative a montré que pour la plupart des variantes, une légère baisse des performances est observée, bien qu'elle n'ait pas atteint de signification statistique, comme en témoigne l'augmentation légère de l'ASSD et de la HD ainsi que la légère diminution du score Dice et de l'IOU par rapport à SFB-Net. La Figure 4.4 illustre un exemple de carte de caractéristiques provenant de l'encodeur avant et après avoir été filtré par le SFB. La carte de caractéristiques a été prise à la résolution la plus élevée du réseau et moyennée le long de la dimension des caractéristiques. On peut voir que les caractéristiques qui ne correspondent pas aux structures cardiaques ont une contribution moindre dans cette nouvelle carte. Enfin, le remplacement du transformer au bottleneck par une seule convolution a entraîné une détérioration significative des performances, comme en témoigne l'augmentation plus marquée de l'ASSD et de la HD ainsi qu'une notable diminution du score de Dice et de l'IOU par rapport à SFB-Net. La Figure 4.5 illustre les résultats de segmentation avant post-traitement de SFB-net et de SFB-net avec la couche transformer au bottleneck remplacée par une convolution. La Figure 4.6 montre la distribution de l'ASSD à travers les niveaux de coupes dans le volume cardiaque pour chaque structure cardiaque. Le nombre de coupes avec un ASSD ≥ 5 mm était faible et similaire pour le VG et le myocarde, mais plus élevé pour le VD surtout pour les coupes les plus basales. Les coupes les moins bien segmentées sur ACDC sont également illustrées Figure 4.7. Ces coupes se situent dans la partie la plus à la base du coeur et présentent des caractéristiques communes avec un ventricule droit commencant à se diviser en deux partie. Pour ces coupes, le réseau éprouve des difficultés à déterminer si le ventricule droit doit être segmenté car cette structure n'apparait pas segmenté pour tous les patients dans la vérité terrain.

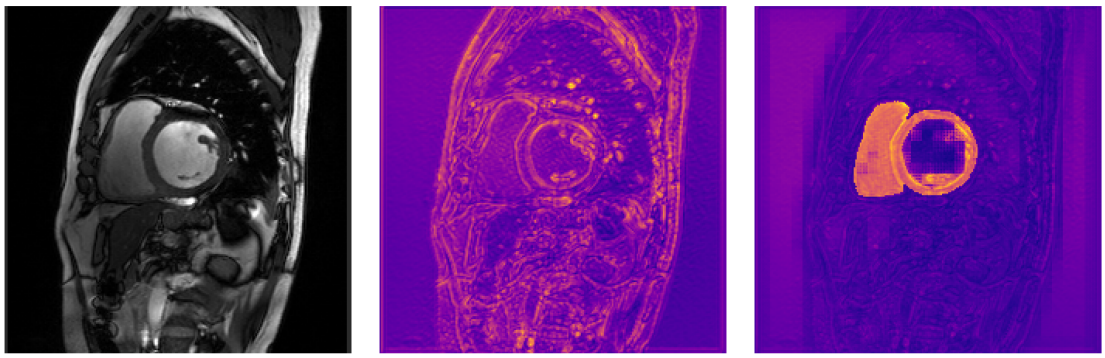


FIGURE 4.4 – (a) Image passée en entrée du réseau. (b) et (c) cartes de caractéristiques provenant de l'encodeur avant et après avoir été filtrée par le SFB respectivement. La carte de caractéristiques a été extraite à la résolution la plus élevée et moyennée sur la dimension des caractéristiques.

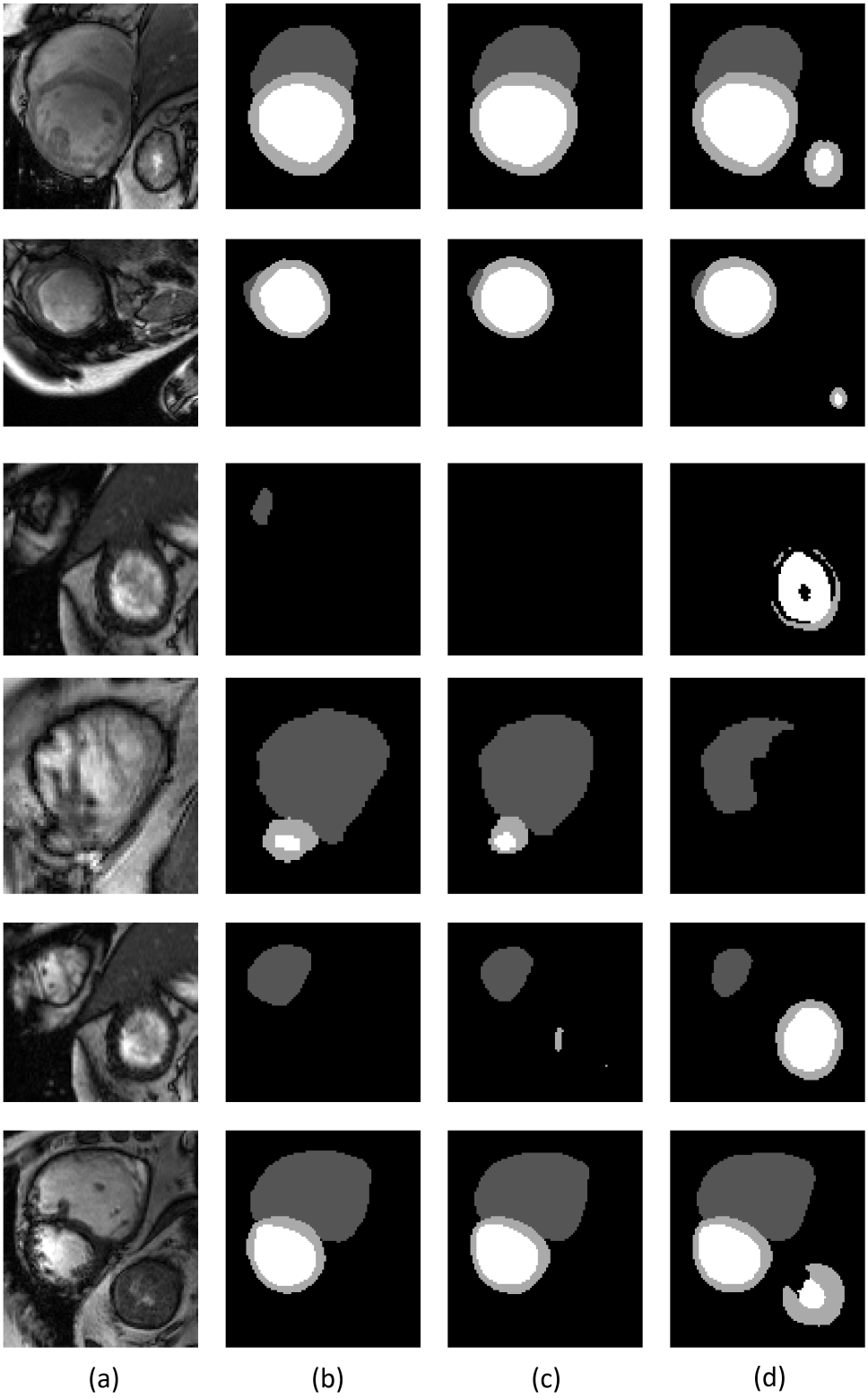
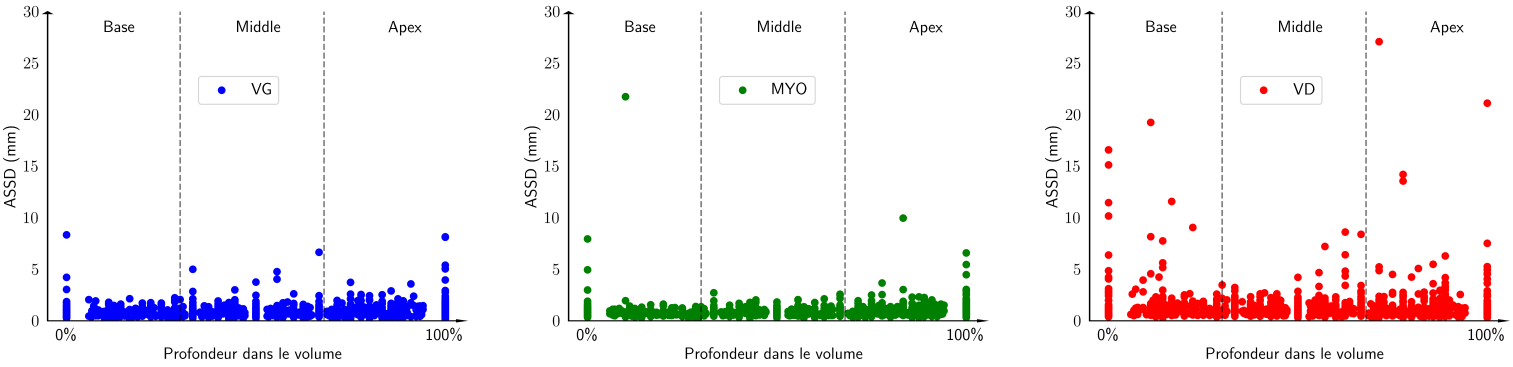


FIGURE 4.5 – Comparaison des résultats de segmentation entre SFB-net et SFB-net sans transformer sur ACDC. (a) Image originale à segmenter, (b) segmentation de vérité terrain, (c) prédictions de SFB-net, (d) prédictions de SFB-net avec la couche transformer remplacée par une convolution avec noyau de taille 3×3 pixels.



(a) (b) (c)

FIGURE 4.6 – ASSD en fonction du niveau de coupe dans le volume cardiaque exprimé en pourcentage (0 % pour la coupe la plus basale, 100 % pour la coupe la plus apicale). (a) ventricule gauche, (b) myocarde, (c) ventricule droit. Mieux vu en zoomant.

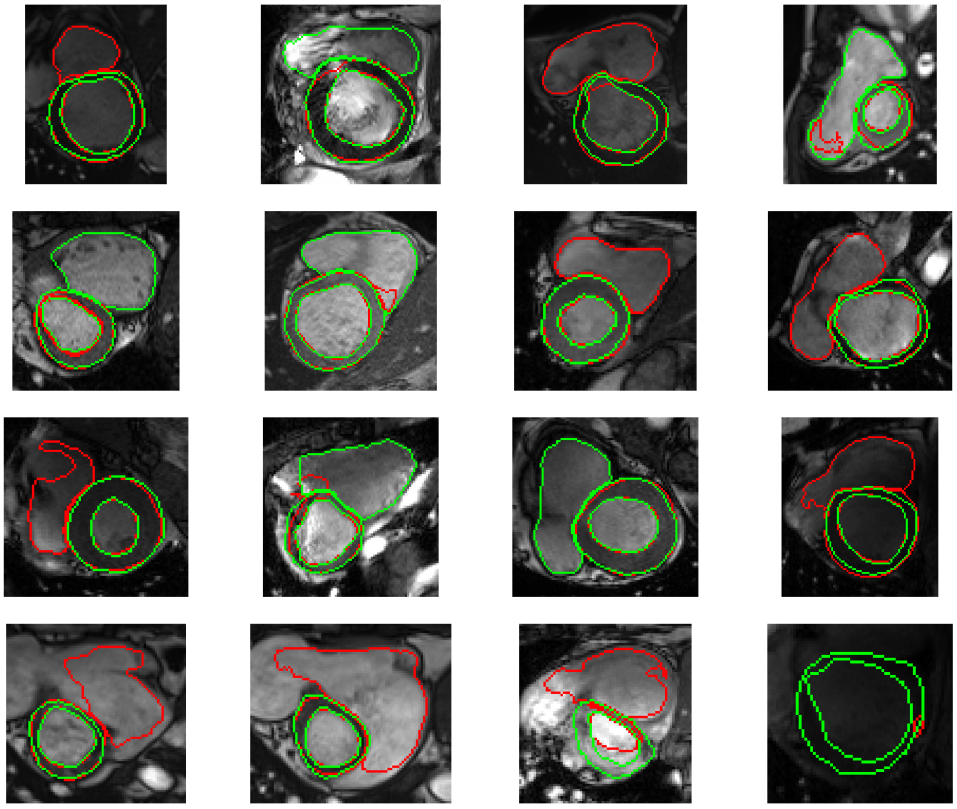


FIGURE 4.7 – Illustration de segmentations de mauvaise qualité identifiées par leur ASSD élevée, où les contours de vérité terrain sont en vert et les prédictions en rouge.

4.2.2 Comparaison à l'etat de l'art

Le Tableau 4.3 présente les résultats de comparaison sur ACDC entre notre approche SFB-net et 6 méthodes récentes de la littérature (à savoir nnU-net, 2021

(Isensee et al. 2018) ; Ω-net, 2018 (Vigneault et al. 2018) ; TransUnet, 2021 (Chen, Lu et al. 2021) ; SwinUnet, 2022 (Cao, Wang et al. 2021) ; Unetr, 2022 (Hatamizadeh et al. 2021) ; nnFormer, 2021 (Zhou, Guo et al. 2022)). Les résultats rapportés proviennent des articles correspondant à ces méthodes, sauf pour nnU-net qui a été réimplémenté. Ces comparaisons révèlent que SFB-net atteint le meilleur score de Dice moyen (92,49 %), ainsi que le meilleur score de Dice pour le myocarde (90,85 %), tandis que les scores de Dice du ventricule gauche (95,08 %, 5e / 7) et du ventricule droit (91,53 %, 2e / 7) sont légèrement inférieurs aux résultats de la littérature.

TABLE 4.3 – Comparaison à l'état de l'art sur ACDC.
La métrique rapportée est le score de Dice. Seuls les résultats de nnU-net et SFB-net ont été recalculés. Les autres résultats sont extraits des articles. Les valeurs en gras correspondent aux performances les plus élevées pour la structure cardiaque considérée.

Méthode	Moyenne	VG	MYO	VD
nnU-net (Isensee et al. 2018)	91.75	94.40	90.18	90.67
Ω-net (Vigneault et al. 2018)	92.16	95.40	89.10	**92.00**
TransUnet (Chen, Lu et al. 2021)	89.71	95.73	84.54	88.86
SwinUnet (Cao, Wang et al. 2021)	90.00	**95.83**	85.62	88.55
Unetr (Hatamizadeh et al. 2021)	88.61	94.02	86.52	85.29
nnFormer (Zhou, Guo et al. 2022)	92.06	95.65	89.58	90.94
SFB-net	**92.49**	95.08	**90.85**	91.53

4.2.3 Paramètres volumétriques

Les associations entre les indices volumétriques prédits et de vérité terrain pour chaque structure cardiaque du jeu de données ACDC sont présentées dans le Tableau 4.4. Les statistiques du graphique de Bland-Altman sont égalements raportées dans ce tableau. En rapportant le biais moyen de Bland et Altman à la valeur de vérité terrain moyenne on obtient une valeur de biais relative en pourcentage : $d_r = \frac{\bar{d}}{\bar{Y}}$ avec $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, n le nombre de volumes, et Y_i la valeur de vérité terrain pour un volume spécifique. Les limites d'agréments (LoA) se définissent de la façon suivante :

$$LoA = \bar{d} \pm 1.96 \times \sigma \tag{4.13}$$

avec σ l'écart type des différences. Les limites d'agréments forment un intervalle comprenant 95% des différences entre les valeurs prédites et les valeurs de vérité terrain pour un paramètre volumétrique.
Les valeurs de vérité terrain ont été calculées à partir des annotations de segmentation. Les corrélations entre les mesures prédites et de vérité terrain étaient élevées ($\rho > 0.9$) pour tous les paramètres, avec des biais de Bland-Altman ($\bar{d}$) faibles ($< 2\%$ sauf pour le volume d'éjection du ventricule droit : biais = -2,27 %) et des limites d'agrément étroites. Les p-values sont toutes supérieures au seuil de significativité fixé à 0.05, sauf pour le volume télédiastolique du VG (p=0.0034). Cela montre qu'il n'y a pas de différence significative par rapport aux valeurs de vérité terrain.

TABLE 4.4 – Associations entre les paramètres quantitatifs volumétriques prédits et de vérité terrain sur le jeu de données ACDC.
Coefficients de corrélation ρ et biais moyen de Bland-Altman d, ainsi que les limites d'agrément (LoA) estimées entre les indices quantitatifs prédits et réels sur ACDC. Les biais moyens relatifs d_r sont rapportés en pourcentage de la valeur réelle correspondante. Les p-values sont calculées à l'aide du test de rangs signés de Wilcoxon entre les valeurs prédites et réelles.

Paramètre	ρ	d_r (%)	$\bar{d}$	p-value
Volume télédiastolique du VG (ml)	1.0	-0.96	-1.585 [-11.295, 8.126]	0.0034
Volume télésystolique du VG (ml)	1.0	-1.26	-1.249 [-14.675, 12.177]	0.0908
Fraction d'éjection du VG (%)	0.98	0.26	0.122 [-7.439, 7.683]	0.9347
Volume d'éjection du VG (ml)	0.96	-0.51	-0.335 [-14.101, 13.430]	0.2471
Masse télédiastolique du MYO (g)	0.99	1.04	1.358 [-15.034, 17.751]	0.2542
Masse télésystolique du MYO (g)	0.99	0.44	0.657 [-17.546, 18.859]	0.4023
Volume télédiastolique du VD (ml)	0.98	-1.05	-1.607 [-20.574, 17.361]	0.0709
Volume télésystolique du VD (ml)	0.97	-0.10	-0.083 [-26.639, 26.473]	0.3153
Fraction d'éjection du VD (%)	0.90	-0.59	-0.276 [-15.597, 15.045]	0.5274
Volume d'éjection du VD (ml)	0.89	-2.27	-1.523 [-28.548, 25.501]	0.2726

4.2.4 Performances de généralisation

Le tableau 4.5 compare les performances de notre modèle entraîné et testé soit sur le jeu de données Quorum, soit sur ACDC et quelques exemples qualitatifs sont montrés Figure 4.8. Le jeu de données ACDC contient des coupes basales où le VD apparaît en deux parties, comme illustré Figure 4.9. Ces coupes n'existent pas dans la base de données Quorum. Par conséquent, afin d'éviter des résultats anormalement bas pour le VD sur ces coupes, les métriques du VD lors des tests sur ACDC ont été calculées sans prendre en compte les 2 coupes les plus basales. Les modèles testés sur le même jeu de données que celui sur lequel ils ont été entraînés ont montré de meilleurs résultats que les autres modèles. La diminution du score de Dice est plus prononcée pour le VD que pour les autres structures (le modèle entraîné sur ACDC et testé sur Quorum atteint un score de Dice inférieur de 5.12, 4.25 et 1.44 points pour le VD, le MYO et le VG respectivement par rapport aux tests sur le jeu de données ACDC). La Figure 4.10a montre le graphique de fréquence cumulée des scores de Dice de chaque modèle. Environ 20% des volumes ont un score de Dice inférieur à 92 pour les modèles entraînés et testés sur le même jeu de données, contre environ 60% pour les autres. La Figure 4.10b montre le score de Dice des coupes 2D par rapport à leur profondeur dans les volumes. On peut voir que l'écart de performance est de 5 points de Dice pour les coupes basales et intermédiaires dans les volumes. Cependant, la différence de performance est plus prononcée pour les coupes apicales, surtout lors des tests sur le jeu de données Quorum. En effet, pour cet ensemble de test, le modèle entraîné sur ACDC a un score de Dice inférieur de 30 points par rapport au modèle entraîné sur le jeu de données Quorum pour les coupes les plus apicales.

TABLE 4.5 – Performances de généralisation

Les scores de Dice, IOU, ASSD et HD sont rapportés sous la forme moyenne ± écart-type. Les p-values sont calculées entre le modèle entraîné sur l'ensemble de données Quorum et testé sur ACDC, et le modèle à la fois entraîné et testé sur ACDC (troisième et quatrième ligne pour chaque métrique). Le test des rangs signés de Wilcoxon est utilisé pour calculer ces p-values. Lors des tests sur ACDC, les résultats du VD pour les 2 coupes les plus basales n'ont pas été pris en compte.

	train	test	Moyenne	VG	MYO	VD
Dice	ACDC	Quorum	88.99 ± 4.44	93.64 ± 3.56	86.59 ± 5.25	86.75 ± 6.72
	Quorum	Quorum	92.14 ± 3.69	95.12 ± 3.10	89.22 ± 5.33	92.08 ± 3.95
	ACDC	ACDC	92.60 ± 2.94	95.08 ± 4.17	90.84 ± 2.98	91.87 ± 5.89
	Quorum	ACDC	89.66 ± 4.03	93.76 ± 4.87	87.33 ± 4.23	87.90 ± 8.10
			< 0.0001	< 0.0001	< 0.0001	< 0.0001
IOU	ACDC	Quorum	80.71 ± 6.62	88.24 ± 5.94	76.69 ± 7.40	77.19 ± 9.96
	Quorum	Quorum	85.76 ± 5.77	90.84 ± 5.31	80.90 ± 7.71	85.55 ± 6.47
	ACDC	ACDC	86.57 ± 4.62	90.90 ± 6.78	83.36 ± 4.85	85.44 ± 8.82
	Quorum	ACDC	81.87 ± 5.96	88.61 ± 7.82	77.75 ± 6.29	79.24 ± 11.53
			< 0.0001	< 0.0001	< 0.0001	< 0.0001
ASSD (mm)	ACDC	Quorum	0.71 ± 0.42	0.51 ± 0.33	0.60 ± 0.31	1.03 ± 0.88
	Quorum	Quorum	0.42 ± 0.25	0.37 ± 0.29	0.44 ± 0.28	0.44 ± 0.27
	ACDC	ACDC	0.43 ± 0.38	0.42 ± 0.58	0.42 ± 0.25	0.45 ± 0.71
	Quorum	ACDC	0.70 ± 0.57	0.62 ± 0.73	0.75 ± 0.78	0.74 ± 0.86
			< 0.0001	< 0.0001	< 0.0001	< 0.0001
HD (mm)	ACDC	Quorum	9.79 ± 3.25	6.02 ± 2.01	8.75 ± 3.69	14.43 ± 7.12
	Quorum	Quorum	7.08 ± 2.17	5.00 ± 1.91	6.73 ± 2.95	9.49 ± 4.75
	ACDC	ACDC	8.20 ± 4.19	6.04 ± 4.39	7.14 ± 4.81	11.42 ± 8.44
	Quorum	ACDC	10.37 ± 5.21	7.26 ± 5.18	11.44 ± 9.74	12.41 ± 7.09
			< 0.0001	0.0288	< 0.0001	< 0.0001

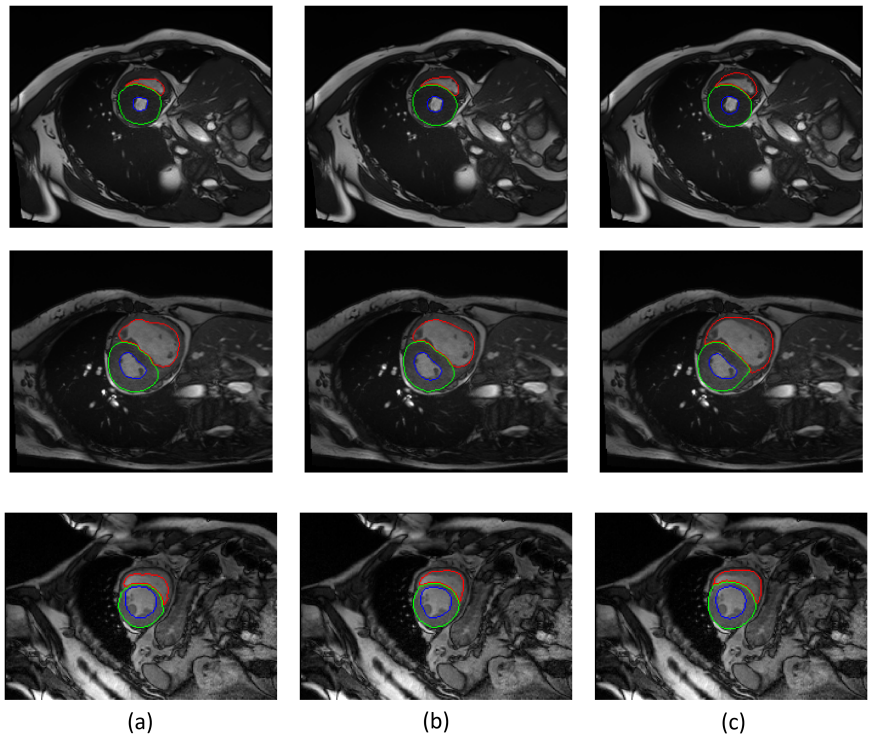


FIGURE 4.8 – Exemple de coupes segmentées pour les 3 volumes les moins performants en termes de score de Dice sur l'ensemble de données ACDC. Les contours du VD sont en rouge, les contours du myocarde en vert et ceux du VG en bleu. (a) Annotations de référence, (b) prédictions du modèle entraîné sur ACDC et (c) prédictions du modèle entraîné sur Quorum. Mieux vu en zoomant.

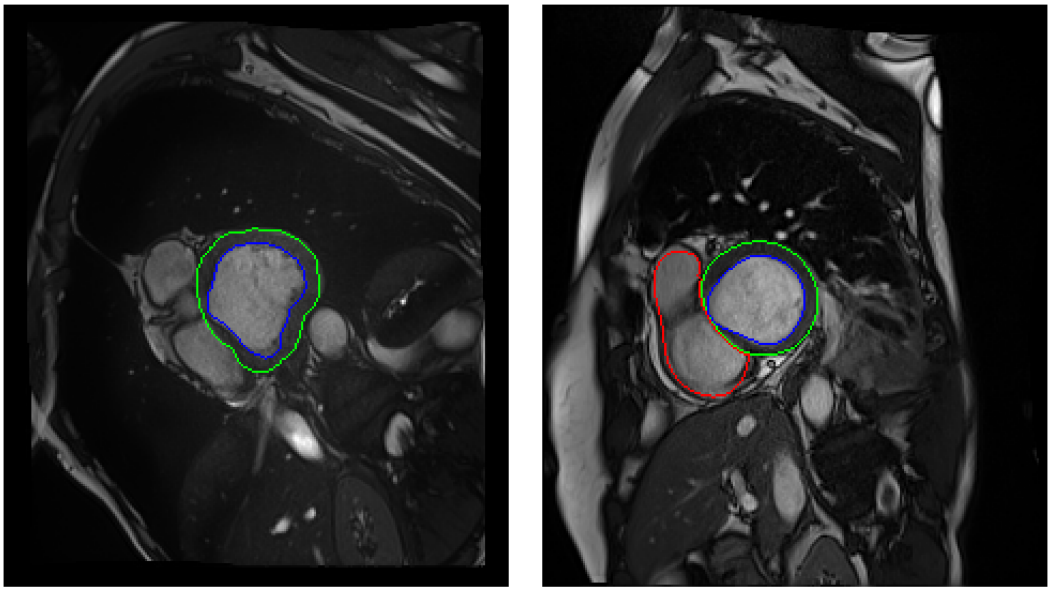
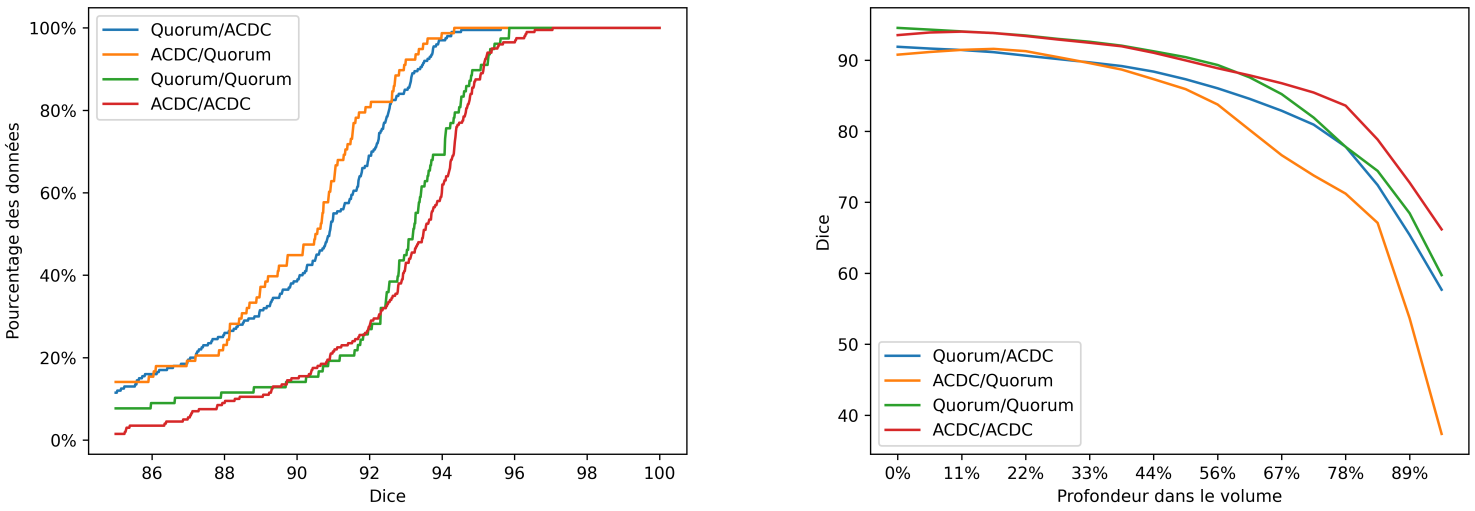


FIGURE 4.9 – Coupes basales problématiques dans ACDC (Rouge : annotation VD, bleu : VG, vert : myocarde). (a) : Les volumes ACDC contiennent des coupes basales où le VD semble être divisé en deux parties distinctes, par conséquent aucune annotation n'est présente. (b) Dans le jeu de données Quorum, ces coupes n'existaient pas. À la place, le VD dans les coupes les plus basales présentait une séparation moins visible qui n'entraînait pas la suppression de l'annotation.



(a) (b)

FIGURE 4.10 – Performances de généralisation pour les modèles entraînés/testés sur ACDC ou le jeu de données Quorum. (a) Graphique de fréquence cumulées des scores de Dice. (b) Scores de Dice de chaque coupe prise individuellement par rapport à la profondeur relative dans les volumes. Les résultats ont été interpolés sur le nombre maximum de coupes dans un volume. 0 % représente la tranche la plus basale et 100 % la plus apicale.

4.2.5 Utilisation de toutes les images du cycle cardiaque

Dans le tableau 4.6, les résultats de notre modèle, entraîné sur toutes les phases du jeu de données Quorum, sont comparés aux performances du même modèle mais entraîné uniquement sur les phases ED et ES. Ce dernier modèle a montré une légère détérioration des performances pour toutes les métriques de segmentation (Dice moyen : -0,23, ASSD moyen : +0,01 mm, HD moyen : +0,15 mm). Cependant, l'entraînement uniquement avec les phases ED et ES a eu un impact plus prononcé sur les mesures de déformations radiales et circonférentielles du VG ainsi que sur les mesures de déformations circonférentielles du VD. En effet, les corrélations sont respectivement de 0,63, 0,76 et 0,12 contre 0,72, 0,84 et 0,57 pour le modèle entraîné sur toutes les phases. Pour l'indice de pic de déformation, l'impact est moins perceptible avec la même corrélation radiale pour le VG et une corrélation circonférentielle du VG (VD) de 0,88 (0,87) contre 0,90 (0,82) lors de l'entraînement sur toutes les phases. En ce qui concerne les valeurs de pic de déformation circonférentielle du VD, l'entraînement uniquement avec les phases ED et ES conduit à davantage de valeurs aberrantes, comme indiqué par les limites d'agréments plus larges ([-34,91 ; 33,41] contre [-13,03 ; 11,54] pour le modèle entraîné avec toutes les phases). La Figure 4.11 affiche les scores de Dice des deux algorithmes par rapport au numéro de phase dans le cycle cardiaque. Les phases ont été triés de façon à commencer par l'ED et les résultats ont été interpolés sur le nombre maximum de phases dans l'ensemble de test. On peut voir qu'en moyenne, il y a un écart de 0,2 points de Dice pour les phases proches de l'ED ou de l'ES, tandis que la différence est d'environ 0,5 points de Dice pour les phases les plus éloignées de l'ED ou de l'ES (par exemple autour des phases numéro 35 à 40).

TABLE 4.6 – Comparaison d'un entrainement sur toutes les phases et seulement sur les phases ED et ES.

Le modèle entraîné sur toutes les phases du cycle cardiaque du jeu de données Quorum est comparé au même modèle, mais entraîné uniquement sur les images ED et ES. Les scores de Dice, ASSD, IOU et HD sont rapportés sous forme de moyenne ± écart type (en mm).

	Metrique	Toutes las phases	Seulement ED et ES
Segmentation	Dice moyen	93.21 ± 1.58	92.98 ± 1.70
	IOU moyen	87.46 ± 2.63	87.08 ± 2.81
	ASSD moyen (mm)	0.14 ± 0.08	0.15 ± 0.09
	HD myen (mm)	4.12 ± 1.20	4.27 ± 1.21
Déformation radiale VG	Correlation pic ES	0.72	0.63
	Biais moyen pic ES [LoA]	$1.22[-28.75; 31.16]$	$6.317[-29.35; 41.99]$
	Biais relatif moyen pic ES (%)	1.97	10.19
	Correlation indices pic ES	0.84	0.84
	Biais moyen indices pic ES [LoA]	$0.07[-3.49; 3.64]$	$0.21[-3.41; 3.82]$
	Biais relatif moyen indices pic ES (%)	0.54	1.52
Déformation circonférentielle VG	Correlation pic ES	0.84	0.76
	Biais moyen pic ES [LoA]	$-0.273[-4.00; 3.45]$	$-0.663[-5.14; 3.81]$
	Biais relatif moyen pic ES (%)	1.45	3.52
	Correlation indices pic ES	0.90	0.88
	Biais moyen indices pic ES [LoA]	$-0.10[-3.01; 2.82]$	$-0.01[-3.16; 3.14]$
	Biais relatif moyen indices pic ES (%)	-0.70	-0.09
Déformation circonférentielle VD	Correlation pic ES	0.57	0.12
	Biais moyen pic ES [LoA]	$-0.74[-13.03; 11.54]$	$-0.75[-34.91; 33.41]$
	Biais relatif moyen pic ES (%)	4.92	4.97
	Correlation indices pic ES	0.82	0.87
	Biais moyen indices pic ES [LoA]	$-0.17[-6.18; 5.84]$	$0.34[-4.93; 5.62]$
	Biais relatif moyen indices pic ES (%)	-1.17	2.34

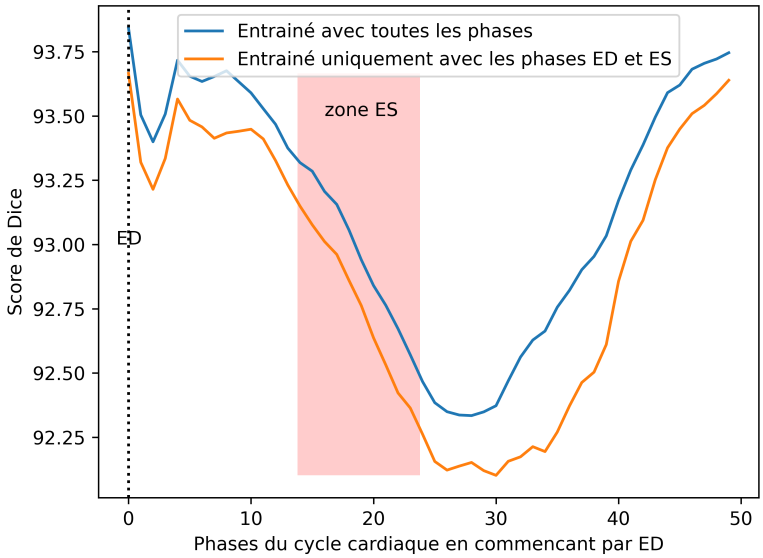


FIGURE 4.11 – Score de Dice moyen en fonction de la phase du cycle cardiaque pour notre modèle entraîné sur toutes les phases et notre modèle entraîné uniquement avec les phases ED et ES. Les résultats sont interpolés sur la longueur maximale d'une séquence dans l'ensemble de test.

4.3 Discussion

Bien que visuellement, nos Swin Filtering Blocks semblent être capables de réduire l'écart sémantique, cela ne s'est pas matérialisé par une augmentation significative des performances de segmentation. La couche transformer au bottleneck, qui, comme les SFB, repose également sur l'attention, a apporté une augmentation significative des performances par rapport à l'utilisation d'une convolution, démontrant que les mécanismes d'attention peuvent effectivement contribuer à améliorer les performances de segmentation. De plus, l'architecture proposée permet de prédire avec précision des indices quantitatifs volumétriques cliniques et généralise bien aux données provenant de différents centres et fabricants. Enfin, l'entraînement uniquement sur les phases ED et ES entraîne une légère diminution des performances de segmentation, mais l'implication sur les mesures de déformation radiale et circonférentielle est plus marquée.

L'utilisation d'une couche transformer au bottleneck plutôt qu'une convolution permet au réseau de prédire des segmentations plus cohérentes. En effet, il est plus rare que des structures cardiaques soient prédites à deux positions différentes. Cela peut s'expliquer par le champ réceptif plus grand de ces couches qui a permis de bénéficier d'une information contextuelle plus étendue. Cette constatation est également conforme à une étude précédente (Raghu et al. 2021) qui a montré que les architectures transformer préservent mieux les informations spatiales des données d'entrées à travers le réseau que les réseaux de neurones convolutionnels (CNN). La capacité à segmenter chaque structure cardiaque comme une seule composante confirme également que les transformers, comme les humains, s'appuient davantage sur les formes pour prendre des décisions, contrairement aux convolutions qui uti-

lisent principalement les textures (Tuli et al. 2021). En examinant les cartes de caractéristiques filtrées par les SFB, on peut remarquer que les réponses bruitées en dehors des structures cardiaques ont été réduites tandis que les zones importantes ont été mises en évidence, suggérant que les SFB ont contribué à réduire l'écart sémantique. De manière intéressante, dans ces cartes, la zone de la cavité ventriculaire gauche n'est pas aussi lumineuse que les autres zones du cœur. Cela peut venir du fait que, puisque le VG est encerclé par le myocarde, le réseau peut se contenter d'apprendre à délimiter le myocarde et en déduire ensuite la forme du VG.

Le graphique de l'ASSD en fonction du niveau de la coupe dans le volume a indiqué une baisse des performances pour les coupes près de la base ou de l'apex du cœur, en particulier pour la cavité ventriculaire droite. Bien que les résultats pour cette structure soient plus bas dans tout le volume, l'augmentation de l'ASSD est plus prononcée et apparait plus tôt vers l'apex, ce qui peut résulter de la diminution plus rapide de la taille de la structure et de sa géométrie complexe et individuellement variable, rendant ainsi sa segmentation plus difficile pour le réseau. Ce résultat est conforme aux conclusions de Bernard et al. 2018 qui ont identifié des difficultés similaires avec les coupes situées aux extrémités des volumes, notant que ce problème se produit également pour les experts cliniques. En ce qui concerne les coupes basales, les erreurs de segmentation sont principalement présentes pour la cavité ventriculaire droite, où les annotations peuvent sembler incohérentes pour le réseau, en raison de la présence d'autres structures telles que l'artère pulmonaire. Les acquisitions étant conventionnellement alignées sur l'axe du VG, l'obliquité des coupes pour le VD et la présence de l'artère pulmonaire peuvent conduire à une division du ventricule droit en deux parties dans les coupes inférieures. Cela peut avoir conduit le réseau à des difficultés à identifier la première coupe dans la pile à partir de laquelle la cavité ventriculaire droite doit être segmentée. Cela a également été confirmé par les résultats visuels des coupes les plus mal segmentées, puisque la plupart d'entre elles sont situées près de la base avec une cavité ventriculaire droite apparaissant souvent à moitié brisée et donc non annotée par l'expert car trop basale. Cela se reflète également dans les corrélations calculées pour la fraction d'éjection du VD et le volume d'éjection, qui sont plus faibles que pour les autres structures. Les autres indices quantitatifs cliniques se situent dans la même plage que ceux rapportés précédemment dans la littérature (Kawel-Boehm et al. 2020).

Les performances de généralisation sont satisfaisantes avec une légère diminution des performances lors du test sur un jeu de données différent de celui sur lequel le modèle a été entraîné. En effet, le modèle entraîné sur ACDC montre une petite diminution des performances lorsqu'il est testé sur le jeu de données Quorum, avec une augmentation de la distance moyenne et maximale entre les contours prédits et la vérité terrain de moins d'un pixel (augmentation de l'ASSD et de l'HD de 0,22 mm et 1,19 mm respectivement avec une taille moyenne de pixel de 1,43 mm^2). Le modèle entraîné sur Quorum montre des performances de segmentation satisfaisantes sur l'ensemble de données ACDC. Cependant, pour ce modèle, le masque prédit pour la cavité ventriculaire droite est plus grand que la vérité terrain et la prédiction du modèle entraîné sur ACDC. Cette différence peut résulter de conventions d'annotation différentes pour cette cavité entre les deux jeux de données. Les performances sur des données hors distribution sont robustes pour la plupart des coupes dans les volumes mais diminuent dans les coupes les plus apicales, ce qui est cohérent avec Campello et al. 2021 et Martín-Isla et al. 2023. Cela peut également

s'expliquer par la différence de conventions d'annotation, car l'ensemble de données ACDC contient des coupes apicales sans annotation pour toutes les classes, tandis que le jeu de données Quorum contient toujours au moins une classe.

En ce qui concerne les performances sur l'ensemble du cycle cardiaque, notre modèle entraîné uniquement sur les phases ED et ES montre des performances légèrement inférieures au modèle entraîné sur toutes les phases. Cependant, l'écart en termes de scores de Dice entre les deux modèles semble se creuser à mesure que la distance par rapport aux phases ED et ES augmente. Il est à noter cependant que la baisse de performance affecte toutes les phases du cycle cardiaque, suggérant que l'entraînement avec plus de phases améliore les performances pour toutes les phases du cycle, y compris les phases ED et ES. Cela résulte probablement de la diversité accrue des données d'entraînement. Étant donné que l'annotation manuelle de chaque phase du cycle cardiaque est chronophage, nos résultats montrent l'intérêt de l'utilisation d'un logiciel capable de fournir, de façon automatique ou semi-automatique, une segmentation de vérité terrain sur l'ensemble du cycle cardiaque. En outre, si les différences de performances de segmentation sont faibles entre les deux modèles, les conséquences plus importantes sur les mesures de déformations suggèrent que les métriques traditionnelles de segmentation ne sont pas idéales pour mesurer la capacité d'un algorithme à être utilisé dans un contexte clinique.

4.3.1 Limitations

Notre méthode présente quelques limitations. Premièrement, le jeu de données utilisé pour comparer les performances du modèle entraîné sur toutes les phases et celui entraîné uniquement avec les phases ED et ES ne contient que des annotations de segmentation pour 3 coupes dans le volume. Par conséquent, il n'était pas possible de calculer des paramètres volumétriques pour l'ensemble du cycle cardiaque. De plus, notre architecture qui utilise des modèles 2D n'extrait pas d'information temporelle et a été appliqué à chaque image du cycle cardiaque de manière indépendante lors de l'inférence. Par conséquent, il y a probablement des possibilités d'amélioration, en particulier pour les mesures de déformation.

Une autre limitation concerne la plus faible qualité de segmentation pour les coupes les plus basales ou apicales. Pour les coupes les plus basales, cela peut être lié à la forme du ventricule droit, qui diffère fortement de sa forme dans les coupes intermédiaires. Pour les coupes les plus apicales, certaines structures peuvent être absentes, ce qui entraîne des valeurs de distance importantes si le réseau prédit leur présence. Il est à noter que dans le jeu de données ACDC, le cœur est complètement absent dans certaines coupes qui auraient probablement été exclues de l'analyse dans un cadre clinique.

Enfin, bien que le réseau montre des résultats satisfaisants en généralisant à d'autres jeux de données, les performances pourraient être encore améliorées en utilisant des techniques d'adaptation de domaine. Cela pourrait être exploré dans des travaux futurs.

4.4 Conclusion

Une nouvelle architecture d'apprentissage profond reposant sur l'attention a été introduite pour segmenter les structures cardiaques à partir d'images IRM ciné petit-

axe sur deux jeux de données différents. Le modèle montre une capacité de généralisation satisfaisante bien qu'il reste des possibilités d'amélioration pour les coupes les plus apicales. Les paramètres volumétriques calculés sont proches des paramètres de référence et en accord avec la littérature, indiquant que l'algorithme peut être utilisé dans un contexte médical pour aider au diagnostic de pathologies cardiaques. L'utilisation de toutes les phases du cycle cardiaque plutôt que seulement les phases ED et ES entraîne un important gain de précision pour l'estimation de la déformation et une légère amélioration des performances de segmentation, ce qui encourage l'utilisation d'outils capables de fournir des annotations de vérité terrain pour l'ensemble du cycle cardiaque.

Troisième partie

Estimation du mouvement cardiaque

Chapitre 5

Etat de l'art

Cet état de l'art présente tout d'abord les méthodes permettant d'estimer le flux optique entre deux images, que ce soit par des approches traditionnelles ou des approches utilisant l'apprentissage. Il introduit ensuite les méthodes utilisant plus de deux images pour l'estimation du flux dans des vidéos. Une section spécifique est ensuite dédiée au calcul du flux optique dans les images médicales. L'état de l'art se termine par une présentation des principales mesures utilisées dans la littérature pour mesurer les performances des flux optiques estimés.

5.1 Calcul du flux optique avec les méthodes traditionnelles

Le membre de droite de l'équation 2.34 peut être développé en l'approximant par une série de Taylor en (x, y, t) :

$$\begin{aligned} I(x+\Delta x, y+\Delta y, t+\Delta t) \approx\ & I(x,y,t) \\ & + \frac{\partial I(x,y,t)}{\partial x}\Delta x \\ & + \frac{\partial I(x,y,t)}{\partial y}\Delta y \\ & + \frac{\partial I(x,y,t)}{\partial t}\Delta t + \epsilon \end{aligned} \tag{5.1}$$

avec ϵ les termes d'ordre 2 et plus de la série de Taylor. En reportant l'équation 2.34 dans l'équation 5.1 et en divisant par Δt des deux côtés on obtient :

$$0 \approx \frac{\partial I(x,y,t)}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I(x,y,t)}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I(x,y,t)}{\partial t} + \mathcal{O}(\epsilon\Delta t)$$

où $\epsilon\Delta t$ est un terme d'ordre Δt (en supposant que Δx et Δy varient comme Δt). Lorsque $\Delta t \to 0$ on obtient :

$$0 \approx \frac{\partial I(x,y,t)}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I(x,y,t)}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I(x,y,t)}{\partial t} \tag{5.2}$$

$$0 \approx \frac{\partial I(x,y,t)}{\partial x}V_x + \frac{\partial I(x,y,t)}{\partial y}V_y + \frac{\partial I(x,y,t)}{\partial t} \tag{5.3}$$

où V_x et V_y représentent le déplacement en x et y *i.e.* le flux optique de $I(x, y, t)$ et $(\frac{\partial I(x,y,t)}{\partial x}, \frac{\partial I(x,y,t)}{\partial y}, \frac{\partial I(x,y,t)}{\partial t})$ sont les dérivées de l'image au pixel (x, y, t). En posant $I_x = \frac{\partial I(x,y,t)}{\partial x}, I_y = \frac{\partial I(x,y,t)}{\partial y}, I_t = \frac{\partial I(x,y,t)}{\partial t}$ il vient :

$$I_x V_x + I_y V_y = -I_t \tag{5.4}$$

ou avec une écriture matricielle :

$$\vec{\nabla} I \cdot \vec{V} = -I_t \tag{5.5}$$

Cependant, les deux composantes de $\vec{V}$ sont inconnues. En l'état, nous avons donc une seule équation pour deux inconnues, ce qui implique une infinité de solutions. Il est donc nécessaire d'introduire des contraintes supplémentaires pour trouver le flux optique. Par exemple Lucas et Kanade 1981 partent du principe que $\vec{V}$ est constant dans un voisinage local autour de chaque pixel. Cela permet d'introduire des équations supplémentaires à l'équation 5.5 et de résoudre le système d'équations par la méthode des moindres carrés. Horn et Schunck 1981 préfèrent utiliser une contrainte de régularité du flux optique. Le problème du calcul du flux optique revient alors à minimiser une énergie comprenant une composante d'attache aux données et une composante de régularisation du flux optique.
D'autres méthodes ne se servent pas de l'équation 5.5 pour calculer le flux optique. C'est le cas des méthodes de "block-matching" (Immanuel, Bala et George 2011 ; Dabov et al. 2006 ; Chen, Hung et Fuh 2001) qui cherchent à identifier les blocs de pixels les plus similaires entre deux images. Pour un bloc candidat dans une image, l'algorithme de "block matching" calcule une distance entre tous les pixels de ce bloc et tous les pixels d'un bloc se trouvant dans un voisinage du bloc courant dans l'image précédente.
Certaines approches s'appuient sur l'information de phase des images pour obtenir le mouvement des pixels entre deux images (Foroosh, Zerubia et Berthod 2002 ; Hoge 2003 ; Erturk 2003). Ces méthodes s'appuient sur la propriété de la transformée de Fourier indiquant qu'une translation spatiale se traduit par une translation linéaire de la phase dans le domaine fréquentiel avec une magnitude inchangée :

$$f(x - a) = e^{-ia\omega} \hat{f}(\omega) \tag{5.6}$$

où a est une constante, f un signal, $\hat{f}$ la transformée de Fourier de ce signal et $\omega = 2\pi\xi$ avec ξ la fréquence. Par conséquent, les méthodes de corrélation de phases calculent la corrélation entre la transformée de Fourier des deux images ("Cross-power spectrum") puis, effectuent la transformée inverse. Une fois la transformée inverse calculée, les coordonnées de la valeur maximale dans l'image de corrélation de phase ainsi obtenue donne l'information sur le mouvement d'un voisinage de pixels.
Néanmoins, toutes ces méthodes présentent d'importantes limites. L'algorithme de Lucas-Kanade n'est pas adapté pour les grands déplacements ainsi que pour calculer le flux optique dans des zones homogènes (gradient spatial faible). L'algorithme de Horn-Schunk est sensible au bruit et, du fait de l'hypothèse de régularité du flux optique, est peu adapté pour les zones à fort gradient comme les contours des objets. Les algorithmes de corrélation de phase ne peuvent estimer que des mouvements translationnels rigides et sont moins efficaces pour les zones relativement homogènes. Enfin, les méthodes de "block matching" nécessitent de longs temps de calculs et

sont dépendantes de la taille de la région de recherche. Par conséquent, les travaux les plus récents se concentrent sur l'estimation du flux optique par des algorithmes d'apprentissage profond.

5.2 Calcul du flux optique par apprentissage profond

En dehors du cadre spécifique de l'imagerie médicale, la plupart des algorithmes d'apprentissage profond pour l'estimation du flux optique s'appuie sur l'existence d'une vérité terrain pour entrainer leur modèle en cherchant à minimiser l'erreur par rapport à cette vérité. C'est le cas des modèles entrainés sur KITTI (Geiger et al. 2013), Flying chairs (Fischer et al. 2015), MPI Sintel (Butler et al. 2012), FlyingThings3D (Mayer et al. 2016) ou encore Middleblury (Scharstein et Szeliski 2002).
Très souvent, les architectures s'articulent autour d'un volume de coût qui calcule la corrélation entre les patchs de deux cartes de caractéristiques, renseignant ainsi sur leur similarité spatiale et sémantique. Etant donné 2 cartes de caractéristiques F_1 et $F_2 \in \mathbb{R}^{H\times W\times D}$, le volume de coût se calcule grâce à un produit scalaire entre les patchs de coordonnées x dans F_1 et $x + d$ dans F_2 :

$$c(x, x + d) = F_1(x) \cdot F_2(x + d) \tag{5.7}$$

On obtient ainsi un volume 4D de taille $H \times W \times H \times W$ contenant la relation entre tous les patchs de F_1 et tous les patchs de F_2. La complexité du calcul du volume de coût est donc $\mathcal{O}((HW)^2)$. De ce fait, ce calcul peut difficilement s'appliquer aux cartes de caractéristiques de hautes résolutions. En pratique, pour un patch de coordonnée x dans F_1, on calcule la correspondance uniquement avec un nombre restreint de patchs $x + d$ de F_2 se situant dans un voisinage de taille d autour de x (Sun, Yang et al. 2018 ; Hui, Tang et Loy 2018 ; Fischer et al. 2015). On obtient alors un tenseur de taille $H \times W \times D^2$ avec $D = 2d + 1$. Le volume de coût est généralement traité avec des convolutions 2D (Sun, Yang et al. 2018 ; Hui, Tang et Loy 2018 ; Fischer et al. 2015) mais d'autres travaux utilisent des transformers (Huang, Shi et al. 2022 ; Shi, Huang, Li et al. 2023), des modules d'attention (Zhang, Woodford et al. 2021) ou des convolutions 4D séparables (Yang et Ramanan 2019).

Les architectures d'apprentissage profond pour l'estimation du flux optique se distinguent par l'emploi ou non de pyramides multi-résolution. Ainsi, avant les travaux de Teed et Deng 2020, des cartes de caractéristiques à plusieurs échelles étaient extraites à l'aide de CNN puis, ces caractéristiques étaient concaténées avec celles du décodeur (Fischer et al. 2015 ; Ilg et al. 2016) ou employées pour calculer le volume de coût à chaque résolution (Sun, Yang et al. 2018 ; Hui, Tang et Loy 2018). Ce calcul permet au réseau d'estimer le flux optique final de façon progressive en ayant accès aux petits comme aux larges déplacements. Ranjan et Black 2016 ne calculent pas de volume de coût mais entrainent 3 réseaux différents à chaque niveau de la pyramide pour estimer un résidu par rapport au flux optique prédit au niveau inférieur. Cette architecture permet d'accroitre la précision pour les grands déplacements puisque l'estimation du mouvement par chaque réseau peut s'appuyer sur l'estimation du réseau du niveau précédent dans la pyramide. De nombreuses

études utilisent également le concept de "warping " des caractéristiques qui consiste à recaler les cartes de caractéristiques correspondant aux deux images à chaque niveau du décodeur à l'aide d'une estimation intermédiaire du flux optique (Sun, Yang et al. 2018 ; Hui, Tang et Loy 2018 ; Hur et Roth 2019). De cette façon, le mouvement résiduel à estimer par les couches suivantes est réduit.

Les travaux de Teed et Deng 2020 ont introduit une nouvelle architecture qui se distingue des travaux précédents et qui fait aujourd'hui référence puisqu'elle est souvent reprise dans les travaux les plus récents. Plutôt que d'utiliser les cartes de caractéristiques extraites à tous les étages de l'encodeur, seule celle de plus faible résolution est utilisée. Les cartes de caractéristiques des deux images générées par un encodeur siamois sont utilisées pour calculer le volume de coût. Puis, RAFT s'appuie sur un convGRU pour itérativement raffiner le flux optique. Le convGRU prend en entrée des caractéristiques de corrélations extraites du volume de coût, des caractéristiques de contexte extraites par un "context encoder" ayant des poids différents de l'encodeur siamois et le flux optique courant. Le flux optique est mis à jour avec le résidu obtenu en sortie du convGRU (Ballas et al. 2016) à chaque itération $f_{k+1} = f_k + \Delta f_k$ où f_k est le flux à l'itération k et Δf_k le résidu. Après L itérations, le flux $f_L \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 2}$ est sur-échantillonné avec une opération de "convex upsampling" différente des décodeurs classiques puisqu'elle permet de sur-échantillonner un flux optique qui a seulement 2 canaux.
La plupart des travaux précédemment cités dans cette section font usage d'un raffinement progressif du flux optique, généralement avec un décodeur pyramidal. Seuls les travaux de Ilg et al. 2016 et Hur et Roth 2019 utilisent ce principe pour une seule résolution. Cependant, pour ce faire, ils utilisent séquentiellement plusieurs réseaux ayant chacun plusieurs millions de paramètres, ce qui est lent et limite le nombre d'itérations. Les concepts de "context encoder", "convex upsampling" et le raffinement itératif par réseau récurrent ont été repris par de nombreux travaux d'estimation du flux optique par apprentissage profond. En particulier, certains travaux ont porté sur l'ajout de modules d'attention à RAFT (Ferede et Balasubramanian 2023 ; Xu, Yang et al. 2021 ; Jiang et al. 2021). Dernièrement, les transformers ont été employés, soit en amont du calcul du volume de coût 4D de façon à avoir des caractéristiques plus discriminatives (Zhao, Zhao et al. 2022 ; Sui et al. 2022 ; Lu, Wang et al. 2023 ; Xu, Zhang et al. 2022), soit en aval, pour post-traiter ce volume en le considérant comme une séquence de tokens (Huang, Shi et al. 2022 ; Shi, Huang, Li et al. 2023).

5.3 Flux optique dans les vidéos

Les flux de mouvement sont généralement estimés à l'aide de paires d'images, bien que quelques travaux utilisent plus de deux images en entrée. Ainsi, certaines méthodes employent des couches RNN pour propager l'information de flux de manière temporelle (Qin, Bai et al. 2018 ; Li, Wei et al. 2020). Shi, Huang, Bian et al. 2023 encodent l'information de mouvement vers l'avant et vers l'arrière pour un triplet d'images. Les informations de mouvement sont ensuite propagées au triplet d'images adjacent en utilisant une fonction de warping au sein d'un processus itératif. De même, Liu, Lyu et al. 2019 estiment également le flux de mouvement avant et arrière à partir d'une image centrale dans un triplet d'images. Pendant l'entraîne-

ment, cinq images sont passées en entrée afin d'estimer les cartes d'occlusion entre l'image centrale et ses deux images voisines. Ding et al. 2020 utilisent les flux de mouvement estimés pour imposer la cohérence temporelle entre les segmentations prédites, permettant ainsi d'utiliser des images non annotées dans une vidéo. Yan et al. 2019 utilisent un réseau séparé pour estimer plusieurs flux de mouvement dans une vidéo. Ces mouvements sont ensuite employés pour localiser la cavité ventriculaire gauche et améliorer les résultats de segmentation. Plusieurs travaux ont essayé d'estimer le flux de mouvement entre des images non adjacentes en se basant sur les flux entre des images voisines (Wu, Liu et al. 2023 ; Ye, Kanski, Yang, Chang et al. 2021 ; Janai et al. 2017). Ren et al. 2018 utilisent également deux réseaux pour fusionner les flux de mouvement entre 3 images afin d'obtenir des flux plus précis entre les images adjacentes. Harley, Fang et Fragkiadaki 2022 ont conçu un processus de mise à jour itératif au moment de l'inférence pour mettre à jour la trajectoire d'un ensemble de points tout en tenant compte de l'occlusion dans la vidéo. Lu, Cai et al. 2020 expliquent que la supervision des fonctions de coût intermédiaires dans un processus itératif sur des vidéos peut réduire le problème de propagation d'erreur, conduisant ainsi à des flux de mouvement plus précis.

5.4 Apprentissage profond pour le calcul du flux optique sur des images médicales

Dans l'imagerie cardiaque, l'estimation du flux optique par apprentissage profond a souvent pour but d'effectuer le recalage entre deux images. Du fait de la difficulté à obtenir des vérités terrain de flux optique, la plupart des méthodes ont recours à l'apprentissage non-supervisé (Balakrishnan et al. 2019 ; Ye, Kanski, Yang, Axel et al. 2024 ; Wang, Yang et Papanastasiou 2022 ; Ye, Kanski, Yang, Chang et al. 2021 ; Vos, Berendsen, Viergever, Sokooti et al. 2019 ; Vos, Berendsen, Viergever, Staring et al. 2017). Un réseau U-net apprend alors à prédire le champ de déformations horizontale et verticale ainsi qu'en profondeur dans le cas de recalage de volumes. Ce champ de déformations est ensuite utilisé, via un STN (Jaderberg et al. 2015), pour recaler l'image en mouvement sur l'image fixe (on parle aussi de "warping" en anglais).
Les fonctions de coût pour l'apprentissage du champ de déformations intègrent généralement une composante de régularisation spatiale permettant d'obtenir un champ de déformations régulier. La fonction de coût généralement utilisée pour le recalage d'images par apprentissage profond est définie par :

$$\mathcal{L}_{\text{us}}(f, m, \phi) = \mathcal{L}_{\text{sim}}(f, m \circ \phi) + \lambda \mathcal{L}_{\text{smooth}}(\phi) \tag{5.8}$$

où $\mathcal{L}_{\text{us}}$ est la fonction de coût non-supervisée, f est l'image fixe, m est l'image en mouvement, ϕ est le champ de déformations et $\mathcal{L}_{\text{sim}}$ est une fonction mesurant la similarité entre l'image fixe et l'image en mouvement recalée à l'aide de ϕ grâce à l'opération de "warping" $\circ$. On utilise généralement l'erreur quadratique moyenne (MSE), la corrélation croisée normalisée (NCC) (Avants et al. 2008) ou encore une mesure d'information mutuelle (MI) (Maes et al. 1997) pour $\mathcal{L}_{\text{sim}}$. Généralement la composante de régularisation consiste à minimiser le gradient spatial du flux optique de sorte que $\mathcal{L}_{\text{smooth}}(\phi) = \sum_p \|\nabla\phi(p)\|_2^2$ avec p un pixel du flux optique. Néanmoins, une telle forme de régularisation n'intègre aucune contrainte physique.

Par conséquent, plusieurs travaux ont travaillé à l'intégration d'une composante de régularisation tenant compte de contraintes biomécaniques de manière à obtenir un mouvement plus réaliste (Lee et Genet 2019 ; Qin, Wang et al. 2020 ; Lu, Ahn et al. 2021 ; Zhang, You et al. 2022 ; Lu, Jin et al. 2023).
De nombreuses recherches se sont également orientées vers l'utilisation de composantes de "cycle consistency" dans la fonction de coût (Kim, Kim, Park et al. 2021 ; Lu, Yang et al. 2019 ; Kim, Kim, Lee et al. 2019 ; Kuang 2019). Concrètement, on estime à la fois la transformée ϕ permettant de transformer m vers f, ainsi que la transformée ϕ^{-1} allant de f vers m. En appliquant ϕ puis ϕ^{-1}, on peut mesurer l'écart avec l'image de départ m :

$$\mathcal{L}_{\text{cycle}}(f, m, \phi) = \mathcal{L}_{\text{sim}}(m, (m \circ \phi) \circ \phi^{-1}) \tag{5.9}$$

Cette composante a pour but d'obtenir une transformation cohérente avec la transformation inverse afin de préserver la topologie entre l'image en mouvement et l'image déformée.

On utilise souvent le déterminant de la Jacobienne d'un pixel pour mesurer la préservation locale de la topologie et la régularité du champ de déformations/flux de mouvement. Le déterminant de la Jacobienne renseigne sur la nature de la déformation appliquée à un pixel (ou voxel en 3D). Un déterminant positif indique un accroissement de la taille du pixel si le déterminant est supérieur à 1 et une réduction s'il est inférieur à 1. Un déterminant de 0 indique une réduction du nombre de dimensions. Les pixels dont le déterminant de la Jacobienne est négatif indiquent une inversion locale de l'orientation (on parle de "pixel folding" en anglais). Cette inversion indique un mouvement non physiquement possible. Le modèle a alors prédit un mouvement irrégulier qui ne préserve pas la forme des structures anatomiques, le mouvement n'est localement pas continu (Figure 5.1). Les composantes $\mathcal{L}_{\text{smooth}}$ et $\mathcal{L}_{\text{cycle}}$ ont pour conséquence de réduire le nombre de pixels ayant un déterminant de la Jacobienne négatif.

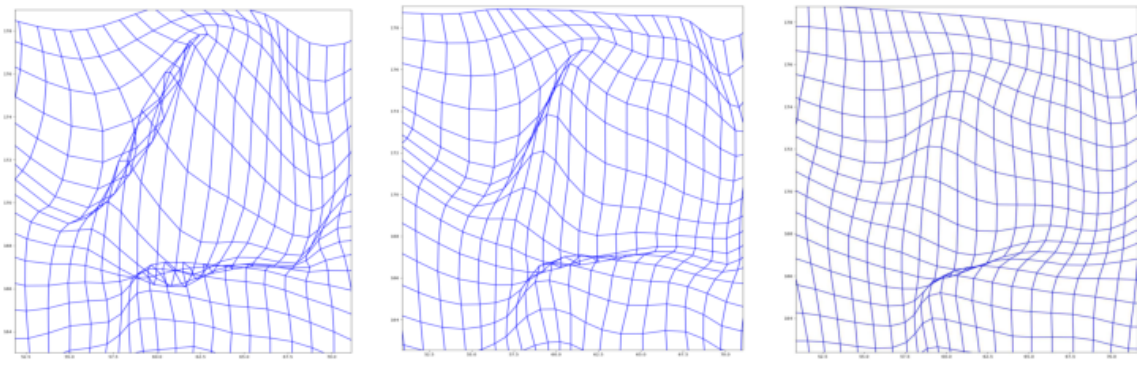


FIGURE 5.1 – Application d'une déformation à une grille. La déformation appliquée est obtenue avec un λ (equation 5.8) de plus en plus élevé de la gauche vers la droite. Lorsque λ est faible (à gauche), on a des inversions de l'orientation (déterminant de la Jacobienne négatif). A mesure que l'on augmente λ, les inversions sont moins visibles. Image issue de (Kuang 2019).

Dalca et al. 2018 introduisent un post-traitement permettant de s'assurer que le champ de déformations produit est un difféomorphisme, c'est à dire qu'il représente

un mouvement physiquement possible sans pixel ayant un déterminant de la Jacobienne négatif. Les déformations diffeomorphiques sont différentiables, inversibles et préservent la topologie. Pour obtenir un champ de déformations difféomorphique, il est nécessaire de calculer la carte exponentielle permettant de passer de l'algèbre de Lie au groupe de Lie. Pour ce faire, les auteurs utilisent la méthode "scaling and squaring" (Arsigny et al. 2006) pour intégrer le champ de déformation de façon itérative. À la suite de ces travaux, de nombreuses recherches traitant du recalage d'images cardiaques reprendront cette technique pour éliminer les pixels avec déterminant de la Jacobienne négatif (Ghadim et Azarnoush 2023 ; Krishnaswamy et al. 2023 ; Sheikhjafari et al. 2022 ; Krebs, Delingette et al. 2019 ; Krebs, Mansi et al. 2018 ; Ye, Kanski, Yang, Chang et al. 2021).
Lorsque des annotations de segmentation sont disponibles, plusieurs auteurs ont montré l'intérêt de les utiliser dans la fonction de coût en recalant les étiquettes de l'image en mouvement vers les étiquettes de l'image fixe. Il est alors possible de calculer l'écart entre les étiquettes de l'image fixe et celles obtenues en warpant par le mouvement estimé :

$$\mathcal{L}_{\text{seg}}(s_f, s_m, \phi) = f_{\text{seg}}(s_f, s_m \circ \phi) \tag{5.10}$$

où s_f est l'annotation de segmentation de l'image fixe, s_m l'annotation de segmentation de l'image en mouvement et f_{seg} une fonction mesurant la précision de la segmentation comme par exemple le critère de Dice ou l'entropie croisée. On s'assure ainsi que le réseau apprend une déformation qui préserve les contours de certaines structures, indépendamment de l'intensité des pixels de l'image. (Balakrishnan et al. 2019 ; Hu, Modat, Gibson, Ghavami et al. 2018 ; Hu, Modat, Gibson, Li et al. 2018 ; Qin, Bai et al. 2018 ; Hering et al. 2018).
Enfin, de façon similaire aux méthodes d'estimation du flux optique par raffinement itératif (Teed et Deng 2020 ; Ranjan et Black 2016), plusieurs approches ont recours à des méthodes itératives ou récursives pour progressivement estimer le champ de déformation permettant de recaler l'image en mouvement avec l'image fixe. Les premières itérations estiment un flux de mouvement grossier alors que les itérations suivantes prédisent des déplacements plus petits. Un réseau de neurones apprend l'estimation de mouvement à chaque itération en prenant en entrée l'image fixe ainsi que l'image en mouvement recalée avec le flux de mouvement courant. Chaque flux de mouvement ainsi obtenu est alors agrégé avec les flux précédents par la somme (Blendowski, Hansen et Heinrich 2021 ; Sandkühler et al. 2019) ou le warping (Zhao, Dong et al. 2019). Ces méthodes reposent donc sur la composition de flux de mouvement de manière à simplifier l'apprentissage du réseau qui doit estimer des flux avec un mouvement de plus en plus faible. Le flux optique entre les deux images est vu comme la composition de flux de mouvements plus faibles. Néanmoins, ces méthodes ont pour objectif d'estimer le flux optique entre deux images et n'ont pas pour but de composer les flux de mouvements entre les images d'une vidéo.

Il convient ici de rappeler qu'uniquement le backward warping est implémenté dans les frameworks d'apprentissage profond (Pytorch, Tensorflow) pour le STN. En effet, le forward warping n'assure pas que chaque pixel de l'image cible a une correspondance dans l'image source ce qui peut créer des trous (application injective). A l'inverse, le backward warping établit une correspondance entre chaque pixel de la cible et au moins un pixel de l'image source (application surjective), évitant ainsi

des valeurs non définies dans l'image cible. Une description du warping forward et backward se trouve Figure 5.2. Il en résulte que l'on se sert du mouvement allant de l'image t vers l'image $t+1$ pour "warper" l'image $t+1$ vers l'image t.

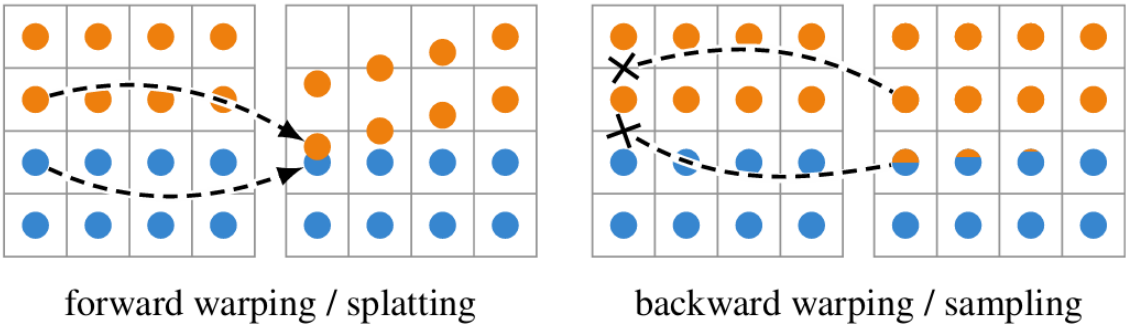


FIGURE 5.2 – A gauche le forward warping peut créer des trous (pixel en haut à gauche) et plusieurs pixels de l'image source peuvent être projetés vers le même pixel de l'image cible. A droite, le backward warping assure que chaque pixel de l'image cible ait une correspondance dans l'image source. Image issue de Niklaus et Liu 2020

5.5 Mesures de performance de flux optique

Il existe plusieurs mesures permettant d'évaluer la performance des méthodes d'estimation du flux optique.

End Point Error (EPE)

Pour un pixel i, l'EPE mesure la distance euclidienne entre le flux optique de vérité terrain $\mathbf{f}_i = (u, v)$ et le flux optique prédit $\hat{\mathbf{f}}_i = (\hat{u}, \hat{v})$:

$$\text{EPE}_i = \sqrt{(\hat{u} - u)^2 + (\hat{v} - v)^2} \tag{5.11}$$

avec (u, v) le déplacement de vérité terrain dans le sens vertical et horizontal pour ce pixel et $(\hat{u}, \hat{v})$ le déplacement prédit pour ces mêmes directions. L'EPE mesure donc l'erreur de déplacement d'un pixel. En calculant la moyenne de l'EPE pour tous les pixels d'une image, on obtient l'EPE moyen : $\text{AEPE} = \frac{1}{N}\sum_{i=1}^{N} \text{EPE}_i$ où N est le nombre de pixels dans l'image.

F1-all

Le F1-all représente le pourcentage de cas extrêmes ("outliers") rapporté au nombre total de pixels. Un cas extreme est un pixel pour lequel le mouvement estimé a une EPE supérieure à 3 pixels ou à 5% de la magnitude du flux optique de vérité terrain :

$$\text{F1-all}(\%) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(EPE_i > \max(3, 0.05 \times \|f_i\|)) \times 100 \tag{5.12}$$

où $\mathbb{1}$ est la fonction indicatrice.

Average Angular Error (AAE)

L'erreur angulaire mesure l'angle entre le mouvement prédit et le mouvement de vérité terrain. En calculant la moyenne de ces angles pour tous les pixels on obtient l'AAE :

$$\text{AAE} = \frac{1}{N}\sum_{i=1}^{N} \arccos\left(\frac{\hat{f}_i \cdot f_i}{\|\hat{f}_i\|\|f_i\|}\right) \tag{5.13}$$

La similarité cosinus étant comprise dans l'intervalle $[-1; 1]$, en prenant l'arc cosinus, on a une fonction bien définie qui donne une mesure comprise entre 0 et π.

Structural Similarity Index (SSIM)

Le SSIM n'est pas une mesure de flux optique car il mesure la similarité entre deux images et ne calcule donc pas de métrique concernant le déplacement des pixels. Cependant, le SSIM peut donner une indication sur la qualité du mouvement prédit en l'absence de vérité terrain de flux optique, ce qui est souvent le cas dans l'imagerie médicale. Pour ce faire, on mesure à l'aide du SSIM la similarité entre l'image en mouvement recalée ("warped") à l'aide du flux optique prédit, notée $\hat{F}$, et l'image fixe notée F :

$$\text{SSIM}(F, \hat{F}) = \frac{(2\mu_F\mu_{\hat{F}} + C_1)(2\sigma_{F\hat{F}} + C_2)}{(\mu_F^2 + \mu_{\hat{F}}^2 + C_1)(\sigma_F^2 + \sigma_{\hat{F}}^2 + C_2)} \tag{5.14}$$

où μ et σ sont la moyenne et la variance des deux images et C_1, C_2 sont des constantes pour stabiliser la division quand le dénominateur est très faible. $\sigma_{F\hat{F}}$ est la covariance entre F et $\hat{F}$. A la différence du PSNR et du MSE, le SSIM mesure la similarité structurelle plutôt que la similarité au niveau des pixels. Cela permet d'avoir une mesure de similarité plus proche de la perception humaine.

5.6 positionnement

La plupart des méthodes qui essayent d'estimer la déformation cardiaque à l'aide d'une approche par apprentissage profond utilisent des données volumiques 3D (Morales, Boomen et al. 2021 ; V. Graves et al. 2023 ; Alvarez-Florez et al. 2023 ; Masutani et al. 2023 ; Wang, Sun et al. 2023). Cela a l'avantage d'obtenir une déformation cohérente pour toutes les coupes du volume. Néanmoins, du fait d'une importante consommation mémoire, cela nécessite de n'utiliser que deux volume à chaque itération durant l'entrainement. Par conséquent, ces approches ne peuvent utiliser plus de deux phases du cycle cardiaque, ce qui peut induire l'estimation de mouvements irréguliers et non cohérents temporellement.
Par ailleurs, les méthodes d'estimation du mouvement cardiaque par apprentissage profond reposent généralement sur l'utilisation d'annotations de segmentation afin que le réseau apprenne à prédire des mouvements qui préservent la forme des structures cardiaques (Balakrishnan et al. 2019 ; Morales, Boomen et al. 2021 ; Morales, Cirillo et al. 2023 ; Zhang, You et al. 2022 ; Hering et al. 2018 ; Hu, Modat, Gibson, Li et al. 2018 ; Hu, Modat, Gibson, Ghavami et al. 2018). Cela limite grandement le nombre d'images pouvant être utilisées durant l'entrainement car il est difficile d'obtenir des annotations de segmentation pour toutes les phases du cycle cardiaque. En effet, seules les phases de télé-diastole et télé-systole sont généralement accompagnées d'annotations de segmentation (Campello et al. 2021 ; Bernard et al. 2018 ;

Radau et al. 2009).
Pour répondre à ces deux limitations, nous proposons une approche qui tire parti de toutes les images du cycle cardiaque tout en utilisant les annotations de segmentation de la phase télé-diastolique et télé-systolique. Cette méthode semi-supervisée permet au réseau d'effectuer des prédictions tenant compte de l'information temporelle, tout en préservant les contours des structures cardiaques. En outre, les méthodes proposées reposent sur l'utilisation de couches transformers et sur le calcul du volume de coût, ce qui est rarement présent dans les architectures d'estimation du mouvement dans le domaine médicale.

Chapitre 6

Agrégation de flux optique

6.1 Motivation

Dans le domaine de l'imagerie médicale, les études d'estimation du flux de mouvement se sont principalement concentrées sur les techniques de recalage d'images non supervisées. Ces méthodes tentent généralement d'optimiser à la fois une erreur de similarité entre l'image déformée et l'image fixe, et une erreur de régularisation pour obtenir un champ de déformation lisse (Balakrishnan et al. 2019 ; Qin, Wang et al. 2020 ; Yu, Chen et al. 2020). Les erreurs de similarité consistent généralement à minimiser la différence absolue ou quadratique entre l'image fixe et l'image déformée. Les fonctions de perte de corrélation croisée (Liu, Yang et al. 2022) et d'information mutuelle (Qiu et al. 2021) sont également régulièrement utilisées comme mesure de similarité. L'erreur de régularisation consiste souvent à minimiser la norme du gradient du champ de vitesse (Wang et Zhang 2020 ; Balakrishnan et al. 2019). Ainsi, ces méthodes ne nécessitent aucun flux de déformation de vérité terrain et sont entièrement non supervisées.

Cependant, selon (Balakrishnan et al. 2019 ; Hu, Modat, Gibson, Li et al. 2018 ; Hu, Modat, Gibson, Ghavami et al. 2018), lorsque des annotations de segmentation sont disponibles, l'ajout d'une erreur de segmentation aux deux composantes précédentes de la fonction de coût conduit à des résultats plus précis. Cette erreur de segmentation consiste à mesurer la différence entre la segmentation de l'image fixe et la segmentation de l'image en mouvement déformée par le flux de mouvement estimé. Ces algorithmes sont limités par le nombre d'images utilisées pour l'entraînement, car peu d'images possèdent une annotation de segmentation de vérité terrain. Par exemple, dans le domaine de l'imagerie cardiaque, seules les phases de fin de diastole et de fin de systole sont généralement annotées (Bernard et al. 2018 ; Campello et al. 2021). Ces phases correspondent au moment du cycle cardiaque où le cœur est le plus relâché et contracté respectivement et présentent un mouvement plus important que toutes autres paires de phases de la vidéo. En conséquence, apprendre à prédire ce flux de mouvement est plus difficile que pour une paire de phases plus rapprochées dans la vidéo et pour laquelle le mouvement est plus faible. Cependant, et comme le montre ce travail, estimer le mouvement entre des images adjacentes et composer itérativement ce mouvement conduit à l'intégration des erreurs de recalage et est également sous-optimal. Pour résoudre ce problème, nous présentons une nouvelle méthode de recalage où les flux sont d'abord calculés entre des images adjacentes, puis agrégés à l'aide d'un réseau d'agrégation pour éviter de multiples interpolations.

Notre approche est comparée à plusieurs algorithmes de l'état de l'art pour l'estimation du flot optique et le recalage d'images. Comparée à ces algorithmes, l'approche proposée obtient de meilleures performances sur l'ensemble de ces métriques.

6.2 Méthode

Soit $F_{t-1,t} \in \mathbb{R}^{W \times H \times 2}$ le flux de mouvement représentant le déplacement vertical et horizontal des pixels de l'image $I_{t-1} \in \mathbb{R}^{W \times H}$ vers l'image I_t. Étant donné une séquence vidéo de T images $S = \{I_1, ..., I_T\}$, où seules les images de fin de diastole (ED) et de fin de systole (ES) I_{ED} et I_{ES} possèdent des segmentations de vérité terrain, nous souhaitons estimer le mouvement $F_{1,t}$ de chaque pixel entre l'image en télédiastole I_{ED} et chacune des autres images $I_t \; \forall t \in [1; T] \backslash ED$. En effet, la déformation des structures cardiaques est une mesure relative à la phase télédiastolique de la séquence.
Par ailleurs, la solution consistant à composer les mouvements obtenus entre images adjacentes donne des résultats imparfaits, comme montré en section 6.3.4. Il est donc important d'estimer directement les mouvements des pixels par rapport à l'image de référence. L'hypothèse sous-jacente de ce travail est que, plus la quantité de mouvement entre deux images est petite, plus il est facile d'estimer le déplacement des pixels. Sur la base de ce principe, un algorithme pourrait être conçu pour estimer $F_{1,t}$ en s'appuyant sur des flux de mouvement entre images voisines $F_{t-1,t}$, car ces derniers devraient être plus facile à estimer.
Dans ce travail, un tel algorithme est conçu autour de 2 réseaux neuronaux f_1 et f_2 ayant la même architecture mais paramétrés par des poids différents. f_1 estime le mouvement entre des images adjacentes $F_{t-1,t} \; \forall t \in [2; T]$. Ensuite, f_2 fusionne ces flux de mouvement de manière successive afin d'obtenir le mouvement entre des images non adjacentes $F_{1,t}$.

6.2.1 Processus itératif d'agrégation des mouvements

Pour commencer, le premier réseau f_1 prend en entrée une séquence S et produit un ensemble de $T-1$ flux de mouvement $F_{t-1,t} \forall t \in [2; T]$ entre les images adjacentes. Ensuite, à partir de $F_{1,2}$, qui est directement obtenu en sortie du premier réseau f_1, le second réseau f_2 agrège de manière itérative les flux de mouvement et construit $F_{1,t}$ en utilisant $F_{1,t-1}$ et le flux élémentaire $F_{t-1,t}$.

Pour être plus précis, considérons deux instants consécutifs $t-1$ et t dans le cycle cardiaque. Le flux $F_{t-1,t}$ est estimé entre les images correspondantes I_{t-1} et I_t en utilisant le réseau f_1 :

$$F_{t-1,t} = f_1(I_{t-1}, I_t). \tag{6.1}$$

En utilisant ce flux, un warping permet de calculer l'image recalée $R_{t,t-1}$ obtenue après déformation de I_t vers I_{t-1} ainsi que l'erreur de recalage :

$$R_{t,t-1} = I_t \circ F_{t-1,t} \tag{6.2}$$

$$E_{t,t-1} = R_{t,t-1} - I_{t-1} \tag{6.3}$$

Ce warping est réalisé avec le "spatial transformer" décrit par Jaderberg et al. 2015.
Dans un second temps, le processus itératif d'agrégation de mouvements calcule le

flux entre la première image et l'image t en utilisant le réseau f_2 et l'équation :

$$F_{1,t} = F_{1,t-1} + f_2(X_{1,t-1}, X_{t-1,t}) \quad \forall t \in [3; T] \tag{6.4}$$

où

$$X_{a,b} = [F_{a,b}; I_a; I_b; R_{b,a}; E_{b,a}] \tag{6.5}$$

6.2.2 Architecture des réseaux

f_1 et f_2 reposent sur la même architecture de type U-Net comme décrit Figure 6.1, mais utilisent des poids différents. L'architecture du réseau f est donc décrite une seule fois pour une paire de tenseurs d'entrées de même dimension Y_1 et Y_2 (I_{t-1} et I_t pour f_1, $X_{1,t-1}$ et $X_{t-1,t}$ pour f_2).

Un encodeur f_{enc} est utilisé deux fois pour extraire les caractéristiques $F_{Y_1} = f_{enc}(Y_1)$ et $F_{Y_2} = f_{enc}(Y_2)$ séparément. Ensuite, à la résolution la plus faible, une couche avec des processus d'attention est utilisée en parallèle sur chaque caractéristique de manière à ce que les cartes de caractéristiques se prêtent attention l'une l'autre. Toute cette couche est commune aux deux branches du réseau. Elle est constituée d'un bloc de self-attention, suivi d'un bloc de cross-attention, et d'un MLP. Les blocs d'attention sont similaires à celui proposé par Vaswani et al. 2017. Pour la self attention, Query, Key et Value correspondent à l'entrée F_Y. Pour la cross attention, la key et la value sont composées de la seconde séquence à qui porter attention tandis que la query correspond à l'entrée F_Y.

Seule la carte de caractéristiques $F_{Y_1}^{trans}$ correspondant à la carte de caractéristiques F_{Y_1} après la couche transformer est passée au décodeur f_{dec} afin d'obtenir le flux de mouvement final en sortie. Les cartes de caractéristiques de bas niveau de même résolution estimées par l'encodeur pour les deux entrées Y_1 et Y_2 sont concaténées et passent par deux couches de convolution pour réduire de moitié le nombre de caractéristiques. Ensuite, elles sont concaténées une deuxième fois avec les cartes de caractéristiques correspondantes provenant du décodeur (architecture de type U-net). Un tableau détaillant les couches de notre architecture est disponible en annexe (section .2)

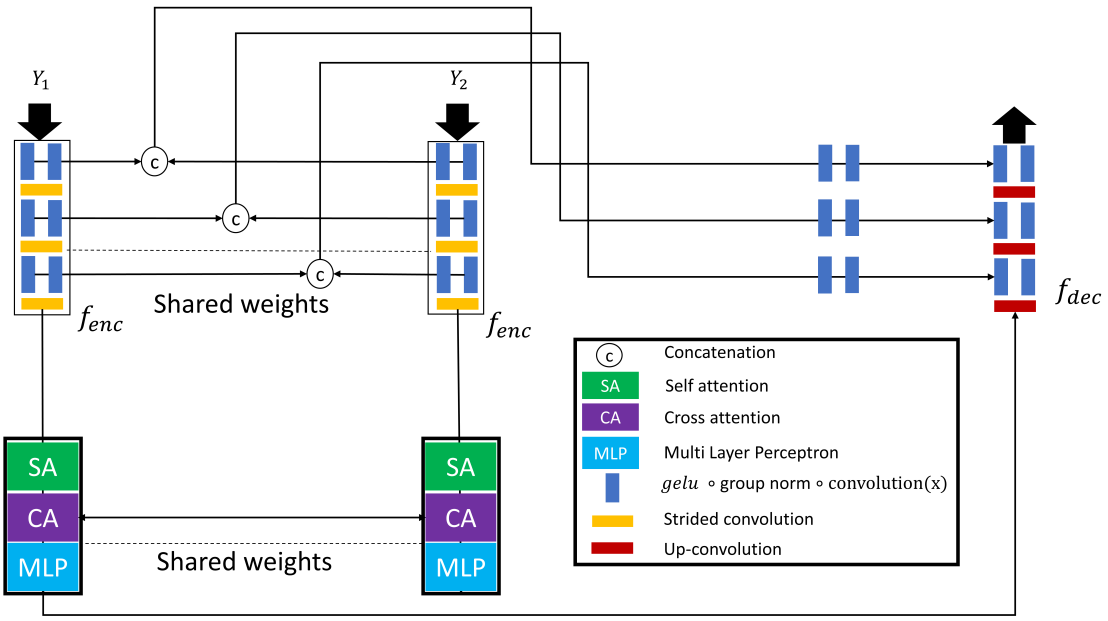


FIGURE 6.1 – Architecture de f_1 et f_2.

6.2.3 Organisation des données pour l'entrainement et l'inférence

Un processus similaire est suivi pour l'entraînement et l'inférence et tire parti de la nature cyclique du cycle cardiaque. La séquence cine-IRM, couvrant l'ensemble du cycle cardiaque, est divisée en deux moitiés S_F ou S_B correspondant au mouvement vers l'avant et vers l'arrière, commençant par l'image ED et se terminant par l'image ES comme décrit Figure 6.2. Pour l'entraînement, en raison des contraintes de mémoire GPU, nous sélectionnons aléatoirement S_F ou S_B. Ensuite, $N-2$ images sont échantillonnées uniformément dans la séquence tout en gardant I_{ED} comme première image et I_{ES} comme dernière afin d'obtenir une séquence de longueur N. De manière générale, comme sur ACDC par exemple, seules les images correspondant aux phases ED et ES de la séquence ont une segmentation de vérité terrain. Pour l'inférence, S_F et S_B sont traités en entier par le modèle sans utiliser de mécanisme d'échantillonnage, c'est-à-dire que toutes les phases du cycle sont considérées.

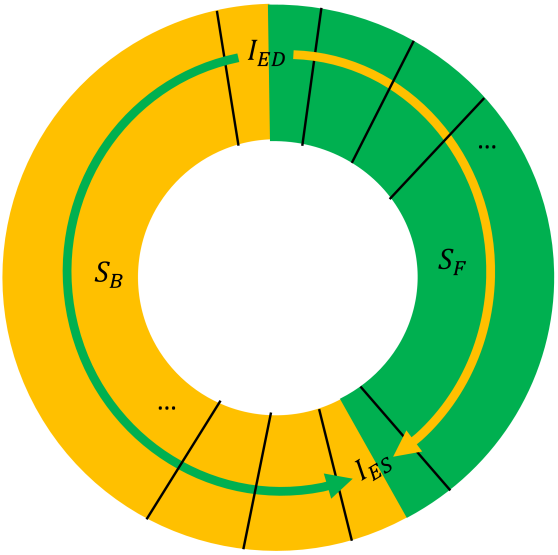


FIGURE 6.2 – S_F et S_B dans le cycle cardiaque.

6.2.4 Fonctions de coût

La fonction de coût $\mathcal{L}$ est composée de plusieurs parties.
Afin d'évaluer la similarité entre deux images I et J, la fonction de corrélation croisée normalisée (NCC) est utilisée. En particulier, la NCC pour un pixel p est définie de la façon suivante :

$$NCC(I,J)(p) = \frac{\left(\sum_{p_i \in W}[I(p_i) - \bar{I}(p)][J(p_i) - \bar{J}(p)]\right)^2}{\left(\sum_{p_i \in W}[I(p_i) - \bar{I}(p)]^2\right)\left(\sum_{p_i \in W}[J(p_i) - \bar{J}(p)]^2\right)} \tag{6.6}$$

où $\bar{I}(p)$ et $\bar{J}(p)$ sont les valeurs moyennes à l'intérieur d'une fenêtre W de taille w^2 centrée autour du pixel p. Comme Ye, Kanski, Yang, Chang et al. 2021 et Balakrishnan et al. 2019, w est fixé à 9. La fonction de corrélation croisée normalisée (NCC) est moins sensible aux variations d'intensité entre les images que la fonction d'erreur quadratique moyenne (MSE), ce qui donne de meilleures performances dans nos expériences. Cette fonction permet de suivre les pixels à travers les images et garantit la précision du processus de recalage.
La composante de la fonction de coût qui évalue la similarité entre l'image recalée $R_{t,t-1}$ et I_{t-1} est définie comme suit :

$$\mathcal{L}_{sim}(I_{t-1}, R_{t,t-1}) = \frac{1}{|\Omega|}\sum_{p \in \Omega} 1 - \text{NCC}(I_{t-1}, R_{t,t-1})(p) \tag{6.7}$$

où Ω est l'ensemble des pixels p d'une image.

Une erreur de segmentation $\mathcal{L}_{seg}$ est également utilisée entre la segmentation de vérité terrain de l'image ED Y_{ED} et la segmentation obtenue en déformant l'annotation de segmentation ES Y_{ES} en utilisant le flot $F_{1,N}$. Cette fonction de coût est uniquement utilisée pour le mouvement allant de ED vers ES car seules les segmentations de ES et ED sont données.

$$\mathcal{L}_{seg}(Y_{ED}, Y_{ES}, F_{1,N}) = f_{seg}(Y_{ED}, Y_{ES} \circ F_{1,N}) \tag{6.8}$$

f_{seg} est une combinaison du critère de Dice et de l'entropie croisée comme proposé dans Isensee et al. 2018 et comme fait pour la segmentation section (4.1).

Cette fonction de coût de régularisation est appliquée à tous les flux $F_{t-1,t}$ et $F_{1,t}$, $\forall\, t \in [2; N]$ et implique donc les réseaux f_1 et f_2 respectivement :

$$\mathcal{L}_{smooth}(F) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \|\nabla F(p)\|_2^2 \tag{6.9}$$

Cette fonction de coût de régularisation est appliquée à toutes les sorties de f_1 et f_2, $F_{t-1,t}$ et $F_{1,t}$ respectivement $\forall t \in [2; N]$.

Ainsi, la fonction de coût finale $\mathcal{L}$ est définie comme suit :

$$\begin{aligned}\mathcal{L}(I, R, F, Y) = &\frac{1}{N-1} \sum_{t=2}^{N} \lambda_1 \mathcal{L}_{sim}(I_{t-1}, R_{t,t-1}) + \lambda_2 \mathcal{L}_{smooth}(F_{t-1,t}) \\ &+ \frac{1}{N-1} \sum_{t=2}^{N} \lambda_1 \mathcal{L}_{sim}(I_1, R_{t,1}) + \lambda_2 \mathcal{L}_{smooth}(F_{1,t}) \\ &+ \lambda_3 \mathcal{L}_{seg}(Y_{ED}, Y_{ES}, F_{1,N})\end{aligned} \tag{6.10}$$

Dans nos tests, les meilleurs résultats ont été obtenus avec $\lambda_1 = 0.5$, $\lambda_2 = 1.0$, $\lambda_3 = 1.0$.

6.2.5 Détails d'implémentation

Le modèle est implémenté avec Pytorch et un GPU NVIDIA V100 de 16G. Les deux réseaux f_1 et f_2 sont entraînés simultanément de manière end-to-end. L'optimiseur AdamW est utilisé avec un pas d'apprentissage et un "weight decay" de 10^{-4}. Le nombre de caractéristiques est doublé à chaque couche de sous-échantillonnage ("strided convolutions") de l'encodeur et réduit de moitié à chaque "up-convolution" du décodeur. Le nombre maximal de caractéristiques à la plus faible résolution est de 512. Notre modèle est entraîné avec une taille de batch de 1 et des couches de "group-normalisation" (avec 8 groupes) sont utilisées dans tout le réseau. Le nombre d'images par séquence pendant l'entraînement est fixé à $N = 12$. L'augmentation de données inclut : "flipping", rotation, zoom, translation, ajustement du contraste, injection de bruit gaussien, flou, "sharpening" et modification de l'intensité. Tous les modèles sont entraînés pendant 180 epochs avec 250 itérations par epoch. Notre modèle contient environ $25M$ de paramètres.

6.3 Résultats

6.3.1 Jeu de données et pré-traitements

Nous utilisons la version du jeu de données Quorum traité par le logiciel CardioTrack pour effectuer nos expériences (section 2.5.5). Seules les annotations de segmentation de fin diastole et de fin systole sont utilisées pour l'entraînement tandis que toutes les annotations sont utilisées pour calculer les métriques sur l'ensemble

de test.
80% des patients ont été utilisé pour l'entraînement et le reste pour les tests. 20% de l'ensemble d'entraînement a été utilisé pour la validation. La séparation est effectuée de manière aléatoire et les patients dans chacun des 3 ensembles sont différents. Un réseau de segmentation binaire pré-entraîné est utilisé pour rogner chaque image à une taille de 192x192 pixels autour des structures cardiaques. Avant d'être traité par le modèle, chaque image d'une séquence d'entrée est normalisée en fonction de la moyenne et de l'écart type de la séquence.

6.3.2 Mesures d'évaluation

Des mesures de suivi de point, de similarité, de segmentation et de régularisation sont calculées pour évaluer le modèle.

Pour la mesure de suivi de points, les points de contour de l'image de référence en télédiastole sont déformés en utilisant les flux de mouvement prédits. L'End Point Error (EPE, équation 5.11) est calculée entre les points prédits $\hat{p}$ et les points de référence p pour chaque image.
La moyenne des EPE sur une séquence entière est ensuite calculée pour obtenir l'EPE moyenne (AEPE). Soit N_k le nombre de points de contour suivis dans une séquence pour la structure k avec $k \in [1;3]$ (RV, MYO, LV). Soit $\widehat{p_{k,t}}$ la position estimée de l'un de ces points dans l'image I_t et $p_{k,t}$ sa position de vérité terrain correspondante. $\widehat{p_{k,t}}$ est définie comme :

$$\widehat{p_{k,t}} = p_{k,1} + F_{1,t}(p_{k,1}) \tag{6.11}$$

où $F_{1,t}(p_{k,1})$ fait référence à l'échantillonnage de $F_{1,t}$ à l'emplacement $p_{k,1}$. Ensuite, l'EPE moyenne pour une séquence de T images et une structure spécifique k, AEPE_k, est calculée comme suit :

$$\text{AEPE}_k = \frac{1}{N_k(T-1)} \sum_{t=2}^{T} \sum_{p_{k,t}=1}^{N_k} EPE(p_{k,t}, \widehat{p_{k,t}}) \tag{6.12}$$

L'AEPE sur une séquence est ensuite calculée comme $\text{AEPE} = \frac{1}{3}\sum_{k=1}^{3} \text{AEPE}_k$. Les résultats rapportés sont la moyenne de toutes les séquences. Le F1-all, définit comme le pourcentage de points pour lesquels $EPE > 3$ est également calculé.

Les points déformés sont transmis au logiciel CardioTrack qui calcule automatiquement les déformations radiales et circonférentielles du ventricule gauche et la déformation du VD (strain). Ces strain prédits sont comparées aux strains de référence. Les corrélations avec la valeur au pic de référence (r_v) et les indices de référence (r_i) sont rapportées pour la phase télésystolique. Nous mesurons également la distance Euclidienne entre les courbes de déformations (dist) calculée comme suit :

$$\text{dist}(\widehat{c}, c) = \frac{1}{T}\sqrt{\sum_{t=1}^{T}(c(t) - \widehat{c}(t))^2} \tag{6.13}$$

où $c(t)$ et $\widehat{c(t)}$ sont respectivement les courbes de déformation (strain) prédites et de références pour la phase t. Lors du calcul des corrélations, le coefficient directeur

de la droite de régression est également rapporté.

Pour la mesure de similarité, l'Indice de Similarité Structurelle (SSIM, équation 5.14) entre l'image de fin diastole (ED) et l'ensemble des images recalées de chaque séquence est calculé. Cet indice est calculé uniquement pour les pixels se trouvant au sein du coeur. Les annotations de segmentation sont utilisées pour calculer le SSIM par structure cardiaque.

En ce qui concerne l'évaluation de la segmentation, la segmentation de chaque image t d'une séquence est recalée vers l'image ED en utilisant le flux de mouvement prédit $F_{1,t}$. Cela est fait pour chaque phase du cycle et chaque coupe du volume. Ensuite, le score de Dice, la distance de Hausdorff (HD) et la distance de surface symétrique moyenne (ASSD) sont calculés sur l'ensemble du volume et moyennés sur tous les volumes (les recalages sont effectués coupe par coupe, mais les métriques de segmentation sont calculées pour l'ensemble du volume).

Pour l'évaluation de la régularisation, le pourcentage moyen de pixels avec un Jacobien $det(J_F(p))$ négatif est calculé sur les flux de mouvements 2D pour les pixels à l'intérieur du cœur. Les masques de segmentation de référence sont utilisés pour calculer ce nombre de pixels avec $det(J_F(p)) < 0$ par structure cardiaque.

Pour chaque type de métrique, les résultats du recalage entre ED et ES (image la plus éloignée de ED), ainsi que la moyenne des recalages entre ED et toutes les images de la séquence sont rapportés. Le test des rangs signés de Wilcoxon est utilisé pour évaluer la signification statistique des résultats et une valeur $p < 0,05$ est considérée comme statistiquement significative.

6.3.3 Méthodes de référence utilisées pour comparaison

L'approche proposée est comparée à plusieurs versions semi-supervisées de VoxelMorph présentées par Balakrishnan et al. 2019. Ces modèles sont ré-entraînés à partir de zéro sur notre jeu de données en utilisant l'implémentation TensorFlow disponible en ligne et uniquement avec les images annotées pour la phase télédiastolique (ED) et télésystolique (ES) car les annotations de segmentation sont nécessaires. Les hyperparamètres optimaux rapportés dans l'article sont conservés. Lors de l'inférence, comme les modèles VoxelMorph ne prennent que 2 images en entrée, nous itérons sur la séquence et, pour chaque instant t, la première image de la séquence $I_1 = I_{ED}$ et l'image courante I_t sont passées au réseau. Le nombre de filtres dans le modèle original de VoxelMorph est augmenté de façon à ce que le réseau contienne environ 25 millions de paramètres, soit le même nombre que notre modèle. Cela conduit à des performances légèrement meilleures que la configuration par défaut. Les performances du modèle entraîné avec la fonction de coût MSE (**VM-MSE**), la fonction de coût NCC (**VM-NCC**) et la version difféomorphique sont présentées (**VM-dif**). Cette version permet d'obtenir 0 pixels avec un Jacobien négatif grâce à un post-traitement consistant à intégrer le champ de déformation de façon itérative (méthode "scaling and squaring" (Arsigny et al. 2006)).

En plus de ces modèles Voxelmorph, des comparaisons avec l'algorithme classique de recalage d'images médicales **SyN** (Symmetric Normalization) sont également pré-

sentées.

Nous présentons également les résultats du modèle plus récent présenté par Zhang, You et al. 2022 (**Bioinformed**) qui utilise une fonction de coût de régularisation qui tient compte de propriétés biomécaniques.

De plus, nous avons également testé **RAFT** (Teed et Deng 2020), entrainé, comme Voxelmorph et Bioinformed, uniquement avec les images de fin diastole et de fin systole de façon à bénéficier des annotations de segmentation. Nous utilisons donc $N = 2$ et une taille de batch de 16. Puisque nous ne disposons pas de la vérité terrain de flux optique pour chaque pixel, nous avons remplacé la fonction de coût $L1$ utilisée dans RAFT par la fonction de coût suivante :

$$\begin{aligned}\mathcal{L}(I, R, F, Y) = \sum_{j=1}^{J} \gamma^{J-j} \big(&\lambda_1 \mathcal{L}_{sim}(I_1, R^j_{ES,ED}) \\ &+ \lambda_2 \mathcal{L}_{smooth}(F^j_{ED,ES}) \\ &+ \lambda_3 \mathcal{L}_{seg}(Y_{ED}, Y_{ES}, F^j_{ED,ES})\big)\end{aligned} \tag{6.14}$$

où J est le nombre total d'itération (c.f Teed et Deng 2020), γ^{J-j} la pondération à chaque itération. $F^j_{ED,ES}$ est le flux optique entre l'image de fin diastole et de fin systole pour l'itération j. $R^j_{ES,ED}$ l'image recalée depuis fin systole vers fin diastole à l'itération j. Les meilleurs résultats sont obtenus avec les poids suivants : $\lambda_1 = 0.5$, $\lambda_2 = 1.0$, $\lambda_3 = 1.0$. Nous avons gardé $\gamma = 0.8$ et $J = 12$ comme indiqué par Teed et Deng 2020.

Enfin, nos résultats incluent également 2 méthodes "naïves" appelées **IterWarpImg** et **IterWarpFlow** qui, à l'inférence, composent itérativement les champs de déformation pour recaler chaque image vers l'image ED. Ces modèles sont entrainés avec $N = 2$ où seules des images adjacentes sont transmises au réseau. En inférence, IterWarpImg déforme itérativement les images/annotations de segmentation depuis l'image courante I_t jusqu'à la première image de la séquence I_1 :

$$R_{t,1} = ((((I_t \circ F_{t-1,t}) \circ F_{t-2,t-1}) \circ \ldots \circ F_{3,2}) \circ F_{2,1}) \tag{6.15}$$

Comme décrit dans Wu, Liu et al. 2023, IterWarpFlow additionne itérativement les champs de mouvement après les avoir recalés afin d'obtenir des champs de déformation entre des images non adjacentes. Ainsi, pour obtenir $R_{t,1}$, l'image I_t recalée vers l'image I_1 on a :

$$F_{1,t} = F_{1,t-1} + (F_{t-1,t} \circ F_{1,t-1}) \tag{6.16}$$

$$R_{t,1} = I_t \circ F_{1,t} \tag{6.17}$$

Ces deux méthodes n'utilisent qu'un seul réseau pour apprendre le mouvement entre des images adjacentes lors de l'entraînement. Toutes les méthodes de références sont entrainées pendant autant d'epochs que notre modèle.

6.3.4 Résultats et comparaison avec l'état de l'art

Les tableaux 6.1 et 6.2 comparent les performances de similarité, de segmentation et de régularisation entre l'approche proposée et les méthodes présentées

dans la section 6.3.3. Tandis que le tableau 6.1 présente les résultats moyens sur toutes les phases, le tableau 6.2 évalue uniquement le mouvement entre ED et ES. Notre modèle a obtenu de meilleurs résultats pour chaque structure cardiaque et pour toutes les métriques de segmentation et de similarité. Les p-values estimées entre notre méthode et RAFT montrent que tous ces résultats sont statistiquement significatifs. Concernant la régularisation ($det(J_F) \leq 0$), les performances sont comparables à celles de Voxelmorph sans le post-traitement difféomorphique ainsi qu'au modèle Bioinformed. Il est à noter que Voxelmorph avec la fonction de coût NCC obtient les seconds meilleurs résultats de similarité tout en ayant un pourcentage limité de pixels avec un Jacobien négatif, surtout pour le mouvement vers la phase de fin-systole. Malgré ces bons résultats, VM-NCC ne préserve pas aussi bien les contours des structures cardiaques que les méthodes concurrentes, comme en atteste les tableaux 6.1 et 6.2. VM-Dif, qui utilise également la fonction de coût MSE, a obtenu 0% de pixels avec un Jacobien négatif grâce à la méthode "scaling and squarring" utilisée pour calculer l'application exponentielle du flux prédit (se référer à Dalca et al. 2019 pour plus d'informations). Ainsi, ce dernier modèle présente les meilleurs résultats en termes de régularité, sans détériorer les performances de segmentation par rapport à VM-MSE. De même, SyN génère également des champs de déformation difféomorphiques entraînant 0% de pixels avec un Jacobien négatif. Les méthodes de recalage itératif n'ont pas atteint les performances de segmentation des autres modèles. Cependant, elles ont obtenu des scores de similarité satisfaisants avec des valeurs de SSIM supérieures à VM-MSE, VM-dif et SyN. Il convient de noter qu'IterWarpFlow a obtenu de mauvaises performances en termes de régularisation du flux de déformation, obtenant les pires résultats de nombre moyen de pixels avec un Jacobien négatif. Cela peut probablement s'expliquer par l'étape d'interpolation requise à chaque fois qu'un champ de déformation est déformé. En revanche, étant donné qu'IterWarpImg ne calcule des champs de déformation qu'entre des images adjacentes, la mesure de régularisation du flux de mouvement est bien meilleure. Il est également à noter que RAFT obtient les meilleures performances de régularisation parmi les méthodes non-difféomorphiques. Cela s'explique probablement par l'absence de décodeur convolutionel et l'usage, à la place, d'une simple couche de "convex upsampling", ce qui limite l'accès aux informations de haute-résolution mais permet d'obtenir des mouvements plus réguliers. Cette forte régularisation explique peut-être les moins bons résultats de similarité obtenus par cette méthode.

TABLE 6.1 – Résultats de segmentation, de régularisation et de similarité pour toutes les phases.

Comparaison avec les méthodes de référence (± écart type). L'annotation de segmentation de chaque image d'une séquence est déformée vers l'annotation de segmentation de ED. Les p-values sont calculées entre notre méthode et RAFT.

	Méthode	Moyenne	VG	MYO	VD
Dice	VM-MSE	94.95 ± 2.24	96.60 ± 1.90	91.76 ± 3.81	96.49 ± 2.01
	VM-NCC	94.69 ± 2.86	96.29 ± 2.90	91.67 ± 4.32	96.10 ± 2.42
	VM-Dif	94.93 ± 2.31	96.59 ± 1.99	91.70 ± 3.87	96.50 ± 2.08
	Bioinformed	95.24 ± 1.70	96.98 ± 1.39	92.20 ± 2.98	96.55 ± 1.64
	RAFT	95.62 ± 1.42	97.27 ± 1.19	92.77 ± 2.60	96.82 ± 1.37
	SyN	93.61 ± 4.52	95.01 ± 5.16	90.18 ± 6.53	95.64 ± 3.24
	IterWarpImg	93.39 ± 3.69	94.50 ± 4.36	89.37 ± 6.01	96.31 ± 1.87
	IterWarpFlow	93.18 ± 3.56	94.46 ± 4.21	88.99 ± 5.84	96.10 ± 1.83
	Notre approche	$\mathbf{96.12 \pm 1.12}$	$\mathbf{97.52 \pm 1.08}$	$\mathbf{93.59 \pm 2.13}$	$\mathbf{97.26 \pm 1.10}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
ASSD (mm)	VM-MSE	0.09 ± 0.07	0.07 ± 0.06	0.13 ± 0.10	0.07 ± 0.07
	VM-NCC	0.11 ± 0.10	0.09 ± 0.13	0.14 ± 0.11	0.09 ± 0.11
	VM-Dif	0.09 ± 0.07	0.07 ± 0.07	0.13 ± 0.10	0.07 ± 0.07
	Bioinformed	0.08 ± 0.05	0.06 ± 0.04	0.12 ± 0.08	0.06 ± 0.05
	RAFT	0.07 ± 0.04	$\mathbf{0.05 \pm 0.03}$	0.10 ± 0.07	0.06 ± 0.04
	SyN	0.16 ± 0.21	0.17 ± 0.31	0.19 ± 0.20	0.12 ± 0.18
	IterWarpImg	0.14 ± 0.13	0.16 ± 0.21	0.19 ± 0.15	0.07 ± 0.06
	IterWarpFlow	0.15 ± 0.13	0.16 ± 0.20	0.20 ± 0.15	0.08 ± 0.06
	Notre approche	$\mathbf{0.06 \pm 0.03}$	$\mathbf{0.05 \pm 0.03}$	$\mathbf{0.09 \pm 0.07}$	$\mathbf{0.05 \pm 0.02}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
HD (mm)	VM-MSE	3.02 ± 1.36	2.69 ± 1.25	2.92 ± 1.50	3.44 ± 1.83
	VM-NCC	3.47 ± 1.90	3.04 ± 1.84	3.29 ± 1.87	4.08 ± 2.56
	VM-Dif	2.96 ± 1.27	2.66 ± 1.21	2.88 ± 1.40	3.33 ± 1.66
	Bioinformed	2.96 ± 1.07	2.61 ± 1.01	2.87 ± 1.25	3.39 ± 1.61
	RAFT	2.61 ± 0.83	2.27 ± 0.78	2.53 ± 0.95	3.05 ± 1.32
	SyN	3.83 ± 2.39	3.52 ± 2.53	3.79 ± 2.59	4.17 ± 2.60
	IterWarpImg	3.81 ± 1.90	3.78 ± 2.19	3.95 ± 2.18	3.69 ± 1.81
	IterWarpFlow	3.91 ± 1.86	3.89 ± 2.14	4.12 ± 2.22	3.71 ± 1.76
	Notre approche	$\mathbf{2.43 \pm 0.75}$	$\mathbf{2.16 \pm 0.81}$	$\mathbf{2.42 \pm 0.97}$	$\mathbf{2.72 \pm 1.11}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
% $det(J_F) \leq 0$ (heart)	VM-MSE	0.22 ± 1.47	0.39 ± 3.00	0.07 ± 0.51	0.20 ± 1.42
	VM-NCC	0.18 ± 0.32	0.32 ± 0.68	0.07 ± 0.22	0.14 ± 0.40
	VM-Dif	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
	SyN	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
	Bioinformed	0.23 ± 1.05	0.39 ± 2.03	0.08 ± 0.41	0.21 ± 1.03
	RAFT	0.04 ± 0.30	0.09 ± 0.66	0.02 ± 0.12	0.03 ± 0.30
	IterWarpImg	0.01 ± 0.02	0.01 ± 0.03	0.00 ± 0.03	0.01 ± 0.04
	IterWarpFlow	0.45 ± 0.79	0.75 ± 1.33	0.21 ± 0.47	0.40 ± 0.90
	Notre approche	0.19 ± 0.45	0.35 ± 0.99	0.10 ± 0.38	0.13 ± 0.44
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SSIM (heart)	VM-MSE	0.73 ± 0.06	0.71 ± 0.07	0.75 ± 0.08	0.73 ± 0.06
	VM-NCC	0.79 ± 0.05	0.77 ± 0.06	0.81 ± 0.06	0.80 ± 0.06
	VM-Dif	0.72 ± 0.06	0.70 ± 0.07	0.74 ± 0.08	0.72 ± 0.07
	SyN	0.71 ± 0.06	0.68 ± 0.07	0.74 ± 0.08	0.71 ± 0.08
	Bioinformed	0.75 ± 0.05	0.73 ± 0.06	0.77 ± 0.07	0.75 ± 0.06
	RAFT	0.70 ± 0.06	0.67 ± 0.08	0.73 ± 0.08	0.69 ± 0.07
	IterWarpImg	0.74 ± 0.06	0.71 ± 0.07	0.76 ± 0.07	0.76 ± 0.06
	IterWarpFlow	0.77 ± 0.05	0.74 ± 0.06	0.79 ± 0.06	0.79 ± 0.06
	Notre approche	$\mathbf{0.84 \pm 0.05}$	$\mathbf{0.83 \pm 0.05}$	$\mathbf{0.84 \pm 0.05}$	$\mathbf{0.85 \pm 0.05}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001

TABLE 6.2 – Résultats de segmentation, de régularisation et de similarité uniquement pour la phase ES.

Comparaison avec les méthodes de référence (± écart type). L'annotation de segmentation de l'image ES est déformée vers l'annotation de segmentation de ED. Les p-values sont calculées entre notre méthode et RAFT.

	Méthode	Moyenne	VG	MYO	VD
Dice	VM-MSE	93.08 ± 1.98	95.06 ± 2.25	89.10 ± 2.99	95.07 ± 2.25
	VM-NCC	90.67 ± 2.92	92.61 ± 4.09	86.51 ± 4.04	92.90 ± 3.11
	VM-Dif	93.08 ± 1.95	95.17 ± 2.21	88.99 ± 3.00	95.07 ± 2.25
	Bioinformed	94.31 ± 1.55	96.33 ± 1.73	90.85 ± 2.36	95.76 ± 1.67
	RAFT	94.57 ± 1.31	96.64 ± 1.53	91.09 ± 2.61	95.98 ± 1.34
	SyN	86.73 ± 5.54	87.39 ± 7.50	81.29 ± 7.74	91.51 ± 4.80
	IterWarpImg	87.57 ± 2.89	88.26 ± 4.30	80.61 ± 5.70	93.83 ± 2.06
	IterWarpFlow	88.09 ± 2.67	89.20 ± 4.01	81.22 ± 5.40	93.86 ± 1.96
	Notre approche	$\mathbf{95.54 \pm 0.97}$	$\mathbf{97.33 \pm 0.93}$	$\mathbf{92.51 \pm 2.32}$	$\mathbf{96.77 \pm 1.04}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
ASSD (mm)	VM-MSE	0.14 ± 0.07	0.12 ± 0.09	0.18 ± 0.07	0.12 ± 0.09
	VM-NCC	0.25 ± 0.16	0.25 ± 0.25	0.26 ± 0.13	0.23 ± 0.19
	VM-Dif	0.14 ± 0.08	0.12 ± 0.09	0.18 ± 0.08	0.12 ± 0.10
	Bioinformed	0.11 ± 0.05	0.08 ± 0.06	0.14 ± 0.06	0.09 ± 0.06
	RAFT	0.10 ± 0.06	0.07 ± 0.05	0.15 ± 0.14	0.08 ± 0.04
	SyN	0.45 ± 0.34	0.57 ± 0.55	0.46 ± 0.28	0.33 ± 0.32
	IterWarpImg	0.35 ± 0.16	0.45 ± 0.30	0.43 ± 0.19	0.16 ± 0.10
	IterWarpFlow	0.32 ± 0.15	0.40 ± 0.26	0.41 ± 0.17	0.15 ± 0.09
	Notre approche	$\mathbf{0.08 \pm 0.05}$	$\mathbf{0.05 \pm 0.03}$	$\mathbf{0.13 \pm 0.14}$	$\mathbf{0.06 \pm 0.03}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
HD (mm)	VM-MSE	4.21 ± 1.23	3.66 ± 1.16	3.86 ± 1.37	5.11 ± 1.87
	VM-NCC	6.10 ± 1.93	5.48 ± 2.06	5.61 ± 2.15	7.21 ± 2.68
	VM-Dif	4.03 ± 1.17	3.59 ± 1.16	3.73 ± 1.16	4.79 ± 1.73
	Bioinformed	3.69 ± 1.24	3.28 ± 1.28	3.54 ± 1.64	4.25 ± 1.66
	RAFT	3.30 ± 0.77	2.78 ± 0.78	3.15 ± 1.27	3.95 ± 1.37
	SyN	7.29 ± 2.27	7.27 ± 2.63	7.44 ± 2.65	7.18 ± 2.44
	IterWarpImg	6.66 ± 1.38	6.80 ± 1.68	7.07 ± 1.75	6.10 ± 1.79
	IterWarpFlow	6.55 ± 1.36	6.61 ± 1.65	7.16 ± 1.79	5.87 ± 1.76
	Notre approche	$\mathbf{2.92 \pm 0.79}$	$\mathbf{2.46 \pm 0.74}$	$\mathbf{2.90 \pm 1.43}$	$\mathbf{3.41 \pm 1.33}$
	p-value	< 0.0001	0.0002	0.0050	0.0003
% $det(J_F) \leq 0$ (heart)	VM-MSE	0.80 ± 2.87	1.50 ± 5.93	0.28 ± 0.94	0.61 ± 2.57
	VM-NCC	0.47 ± 0.41	0.85 ± 0.94	0.20 ± 0.32	0.36 ± 0.66
	VM-Dif	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
	SyN	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$
	Bioinformed	0.80 ± 2.13	1.46 ± 4.39	0.33 ± 0.79	0.61 ± 1.81
	RAFT	0.19 ± 0.63	0.41 ± 1.46	0.08 ± 0.27	0.09 ± 0.47
	IterWarpImg	•	•	•	•
	IterWarpFlow	1.48 ± 1.38	2.31 ± 2.28	0.81 ± 0.94	1.32 ± 1.76
	Notre approche	0.74 ± 0.87	1.41 ± 2.12	0.38 ± 0.67	0.43 ± 0.92
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SSIM (heart)	VM-MSE	0.60 ± 0.09	0.57 ± 0.11	0.61 ± 0.11	0.61 ± 0.10
	VM-NCC	0.63 ± 0.11	0.57 ± 0.14	0.65 ± 0.10	0.66 ± 0.13
	VM-Dif	0.59 ± 0.09	0.55 ± 0.11	0.61 ± 0.11	0.60 ± 0.11
	SyN	0.51 ± 0.14	0.43 ± 0.18	0.58 ± 0.13	0.52 ± 0.19
	Bioinformed	0.64 ± 0.08	0.62 ± 0.10	0.65 ± 0.10	0.65 ± 0.09
	RAFT	0.58 ± 0.09	0.55 ± 0.12	0.62 ± 0.10	0.58 ± 0.12
	IterWarpImg	0.59 ± 0.10	0.51 ± 0.13	0.62 ± 0.11	0.64 ± 0.12
	IterWarpFlow	0.60 ± 0.09	0.52 ± 0.12	0.63 ± 0.10	0.66 ± 0.11
	Notre approche	$\mathbf{0.73 \pm 0.08}$	$\mathbf{0.72 \pm 0.10}$	$\mathbf{0.72 \pm 0.09}$	$\mathbf{0.76 \pm 0.09}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001

La Figure 6.3 présente des résultats de recalage d'images à l'aide du flux optique calculé pour chaque méthode de référence ainsi que pour notre méthode. L'image de fin systole est recalée vers l'image de fin diastole. On peut voir que la méthode IterWarpImg donne une image floutée du fait de l'application successive de la fonction de warping qui effectue une interpolation bilinéaire à chaque étape.

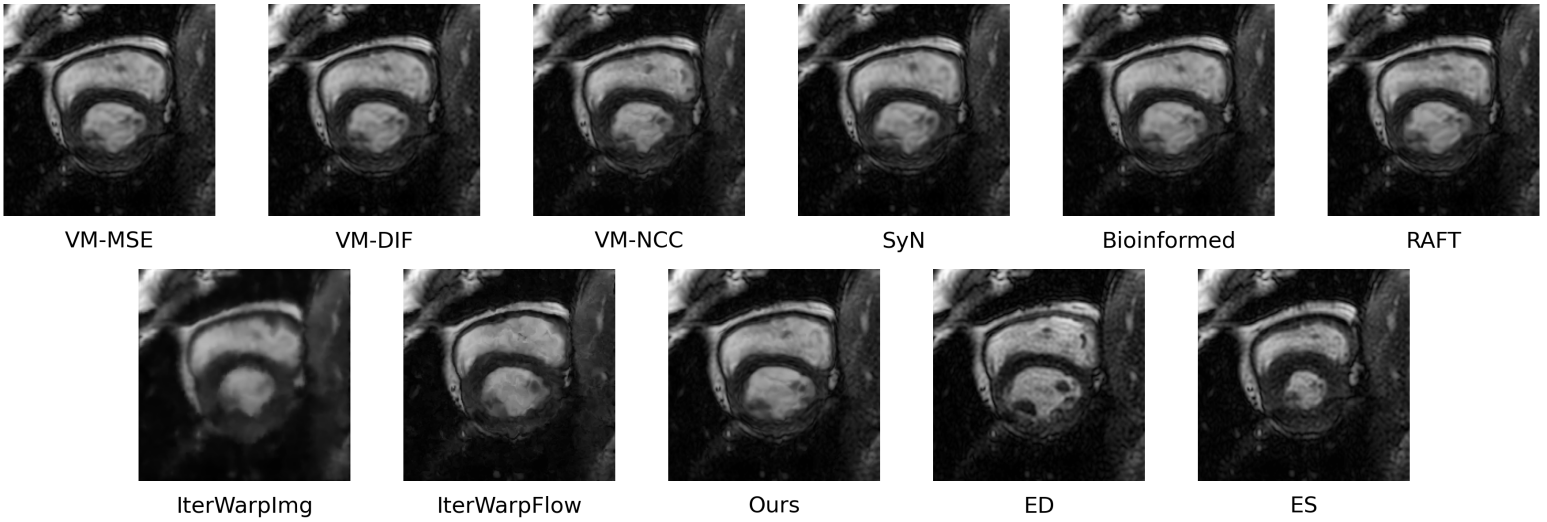


FIGURE 6.3 – Exemple de recalage d'image à l'aide du flux optique calculé pour toutes les méthodes comparées. L'image ES (moving) est recalée vers l'image ED (fixed). Mieux vu en zoomant.

La Figure 6.4 présente le flux optique superposé à l'image de fin diastole pour le même patient que la Figure 6.3 et pour toutes les méthodes sauf IterWarpImg qui ne calcule pas le flux optique entre l'image ED et ES. On peut voir que IterWarpFlow donne de mauvais résultats. De plus notre approche semble produire un flux optique légèrement moins régulier que VM-MSE, VM-Dif, SyN et RAFT.

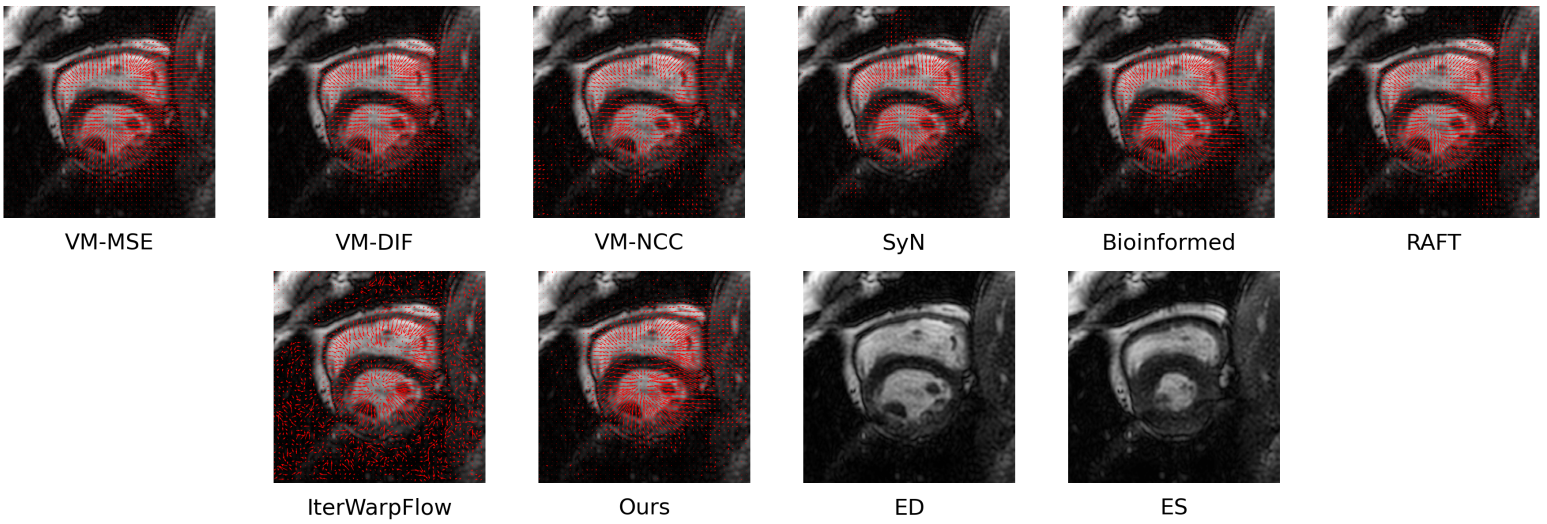


FIGURE 6.4 – Exemple de flux optique superposé à l'image de fin diastole pour toutes les méthodes sauf IterWarpImg qui ne calcule pas le flux optique entre l'image ED et ES. Le flux optique est échantillonné avec un pas de 4 pixels. Mieux vu en zoomant.

La Figure 6.5 présente un exemple de suivi de points entre les images ED et ES. On peut voir que notre approche déplace les points plus près des points de vérité terrain que les autres méthodes. Les méthodes IterWarpFlow et IterWarpImg ne semblent pas donner de résultats satisfaisants pour cette séquence spécifique. Cela vient probablement des recalages successifs des flux optiques et des images en inférence. Ce post traitement introduit une interpolation bilinéaire à chaque étape qui est susceptible de réduire la précision des déplacements.

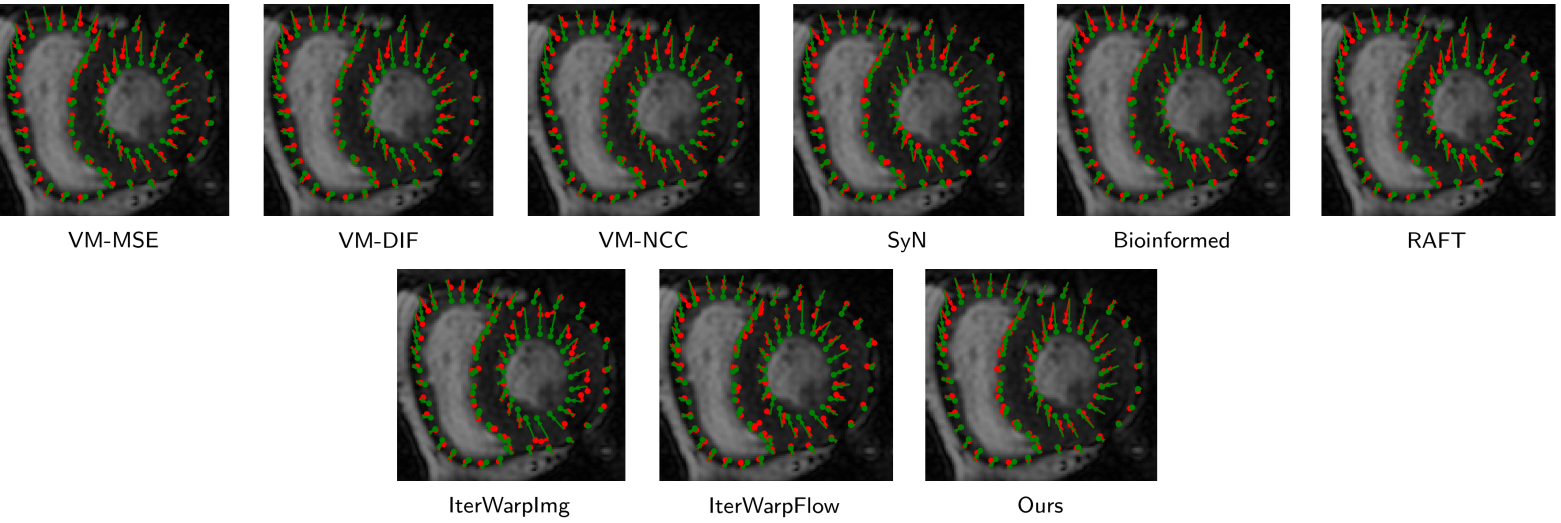


FIGURE 6.5 – Exemple de suivi de points de contour. Les points de contour de la phase ED sont déplacés à l'aide du flux optique allant de ED vers ES. Les points de vérité terrain à ES sont en vert et les points prédits en rouge. Les points sont affichés sur l'image de la phase ES. Mieux vu en zoomant.

La Figure 6.6 présente le résultat de l'application de la déformation prédite par notre approche entre ES et ED à une grille 2D ainsi qu'un grossissement sur une zone contenant des pixels avec un Jacobien négatif. Il est apparent que ces zones se caractérisent par une inversion locale de l'orientation avec des lignes de la grille poussées l'une vers l'autre au point de se croiser.

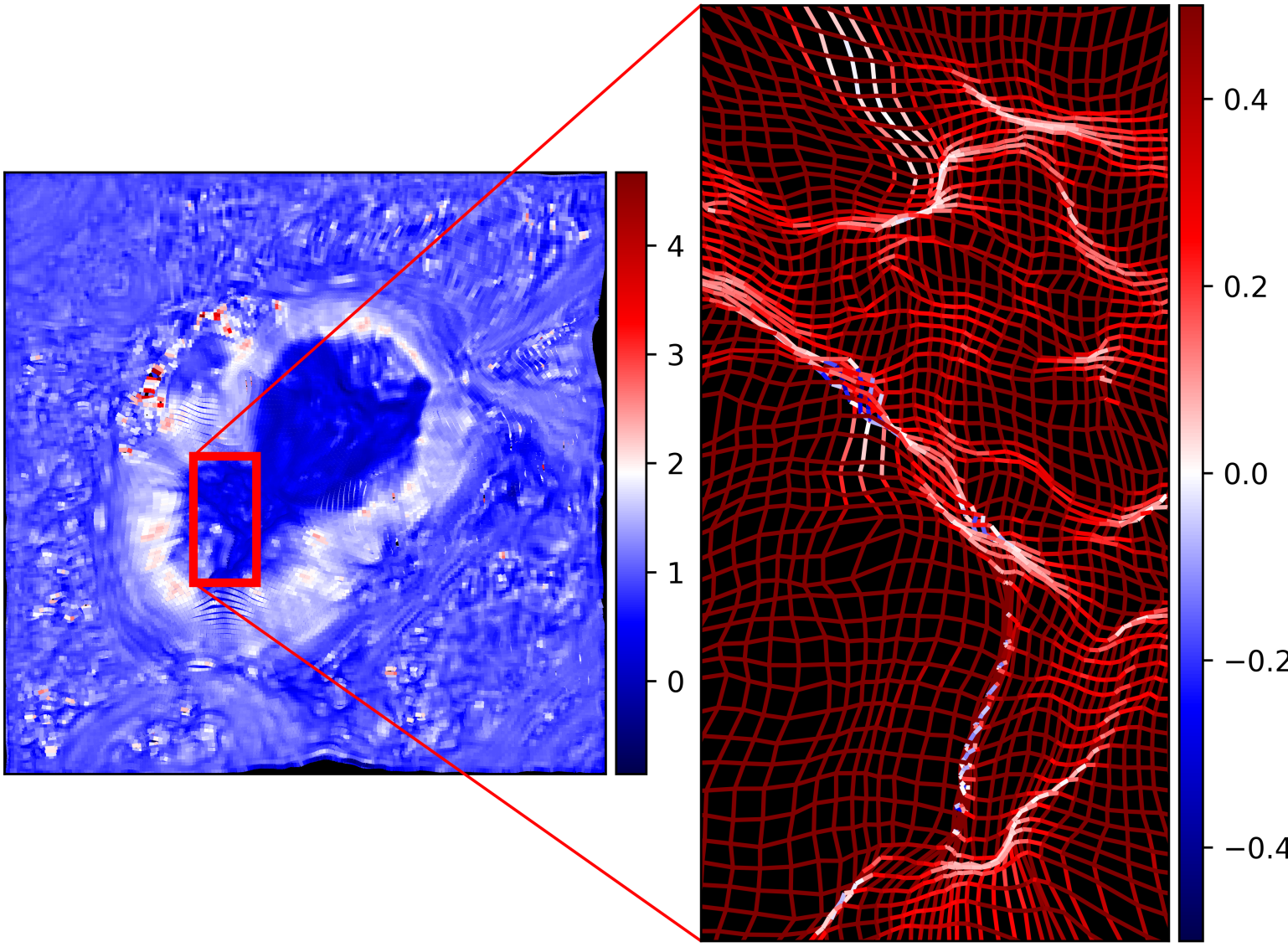


FIGURE 6.6 – Grille 2D déformée par l'application du champ de déformation et zoom sur une zone contenant des pixels avec un Jacobien négatif. Le maillage a été coloré en fonction de la valeur du Jacobien. Mieux vu en zoomant.

La figure 6.7 présente le graphique de fréquences cumulées des EPE moyennes pour l'ensemble des méthodes. Comme on peut le voir dans ce graphique, le nombre de flux optiques avec une EPE moyenne supérieure à 1.5 pixels s'élève à environ 20% pour notre modèle, contre environ 40% pour RAFT, la méthode de référence la plus performante. Ce graphique montre que notre approche contribue à une diminution du nombre de flux de mouvements prédits ayant une erreur de déplacement importante.

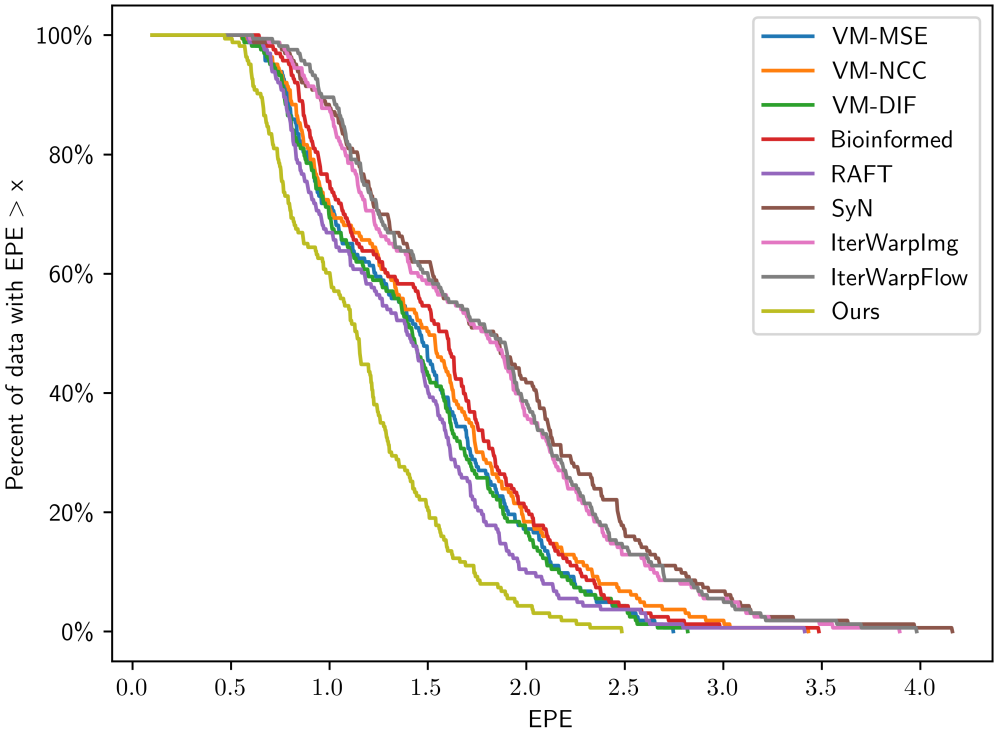


FIGURE 6.7 – Fréquences cumulées des EPE moyennes pour notre approche et les méthodes de références. Les valeurs en ordonnée correspondent au pourcentage des données avec une EPE supérieure à la valeur correspondante sur l'axe des abscisses.

Les résultats de suivi de points sont présentés Tableau 6.3 lorsqu'ils sont estimés sur toutes les images de la séquence et Tableau 6.4 lorsqu'ils sont calculés à partir du plus grand mouvement, entre ED et ES. On peut voir que la méthode proposée obtient de meilleurs résultats dans les deux cas, avec un EPE et un F1-all plus faibles. Parmi les méthodes de référence, RAFT est classée deuxième pour le suivi de points (EPE et F1-all) et les p-values montrent que les résultats sont statistiquement significatifs. Toutes les méthodes parviennent à mieux suivre les points de contours de l'épicarde que ceux de l'endocarde et du ventricule droit. Cela s'explique par la plus faible quantité de mouvement à estimer dans cette zone.

TABLE 6.3 – Résultats de suivi de points pour toutes les phases.
Comparaison avec les méthodes de base (EPE ± écart-type, F1-all ± écart-type). Les p-values sont calculées entre notre méthode et RAFT. EPI = Epicarde, ENDO = Endocarde.

	Methode	Moyenne	VD	EPI	ENDO
EPE (pixels)	VM-MSE	1.44 ± 0.54	1.57 ± 0.68	1.31 ± 0.51	1.44 ± 0.58
	VM-NCC	1.51 ± 0.60	1.65 ± 0.75	1.30 ± 0.53	1.57 ± 0.71
	VM-Dif	1.42 ± 0.54	1.55 ± 0.67	1.31 ± 0.51	1.40 ± 0.58
	Bioinformed	1.53 ± 0.56	1.68 ± 0.67	1.43 ± 0.54	1.48 ± 0.57
	RAFT	1.37 ± 0.52	1.53 ± 0.66	1.29 ± 0.52	1.29 ± 0.49
	SyN	1.82 ± 0.73	1.88 ± 0.85	1.50 ± 0.61	2.09 ± 0.94
	IterWarpImg	1.76 ± 0.69	1.59 ± 0.67	1.40 ± 0.56	2.29 ± 1.02
	IterWarpFlow	1.79 ± 0.68	1.62 ± 0.66	1.43 ± 0.56	2.32 ± 1.02
	Notre approche	$\mathbf{1.15 \pm 0.43}$	$\mathbf{1.26 \pm 0.53}$	$\mathbf{1.06 \pm 0.40}$	$\mathbf{1.13 \pm 0.48}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
F1-all (%)	VM-MSE	10.92 ± 9.08	12.65 ± 11.08	9.01 ± 8.67	11.11 ± 10.57
	VM-NCC	12.08 ± 0.09	13.65 ± 11.00	9.02 ± 8.90	13.58 ± 11.62
	VM-Dif	10.69 ± 9.11	12.56 ± 11.13	9.12 ± 8.92	10.39 ± 10.36
	Bioinformed	11.60 ± 9.98	14.12 ± 11.86	9.94 ± 9.58	10.73 ± 10.72
	RAFT	8.88 ± 8.55	11.50 ± 11.13	7.91 ± 8.30	7.22 ± 8.40
	SyN	16.66 ± 10.29	16.53 ± 11.41	12.52 ± 10.71	20.94 ± 12.02
	IterWarpImg	16.59 ± 10.70	13.56 ± 11.37	10.74 ± 10.00	25.48 ± 14.31
	IterWarpFlow	16.71 ± 10.73	13.65 ± 11.40	10.87 ± 10.08	25.61 ± 14.32
	Notre approche	$\mathbf{5.96 \pm 6.55}$	$\mathbf{7.55 \pm 8.56}$	$\mathbf{4.73 \pm 5.63}$	$\mathbf{5.60 \pm 8.06}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001

TABLE 6.4 – Résultats de suivi de points uniquement pour la phase ES.
Comparaison avec les méthodes de base (EPE ± écart-type, F1-all ± écart-type). Les p-values sont calculées entre notre méthode et RAFT. EPI = Epicarde, ENDO = Endocarde.

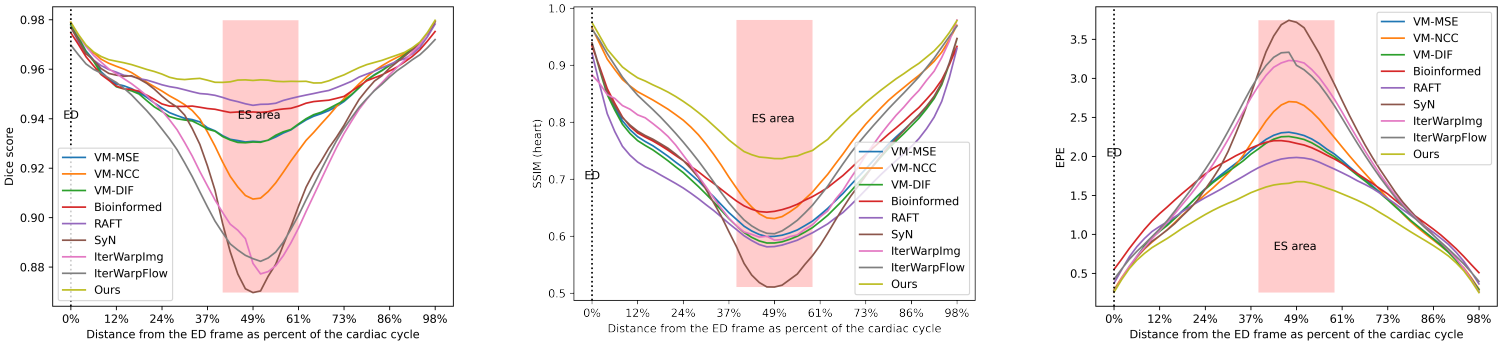
	Methode	Moyenne	VD	EPI	ENDO
EPE (pixels)	VM-MSE	2.30 ± 0.98	2.46 ± 1.33	1.99 ± 0.91	2.45 ± 1.23
	VM-NCC	2.70 ± 1.29	2.87 ± 1.66	2.04 ± 0.99	3.21 ± 1.85
	VM-Dif	2.25 ± 0.96	2.44 ± 1.31	2.00 ± 0.88	2.31 ± 1.23
	Bioinformed	2.17 ± 0.87	2.38 ± 1.11	2.05 ± 0.89	2.08 ± 0.94
	RAFT	1.98 ± 0.84	2.22 ± 1.15	1.83 ± 0.83	1.90 ± 0.89
	SyN	3.74 ± 1.86	3.69 ± 2.21	2.64 ± 1.24	4.89 ± 2.74
	IterWarpImg	3.21 ± 1.57	2.50 ± 1.18	2.30 ± 1.15	4.81 ± 2.77
	IterWarpFlow	3.19 ± 1.34	2.60 ± 1.28	2.24 ± 1.00	4.71 ± 2.27
	Notre approche	$\mathbf{1.71 \pm 0.74}$	$\mathbf{1.87 \pm 1.01}$	$\mathbf{1.56 \pm 0.69}$	$\mathbf{1.69 \pm 0.81}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
F1-all (%)	VM-MSE	24.78 ± 19.20	25.98 ± 20.49	19.56 ± 19.89	28.80 ± 27.08
	VM-NCC	31.18 ± 20.94	29.81 ± 21.29	21.66 ± 21.15	42.08 ± 31.11
	VM-Dif	24.03 ± 19.09	25.87 ± 20.71	19.99 ± 19.96	26.21 ± 26.19
	Bioinformed	22.44 ± 18.23	25.45 ± 19.95	20.15 ± 19.82	21.73 ± 22.49
	RAFT	18.49 ± 16.53	22.02 ± 19.77	15.97 ± 17.23	17.48 ± 20.73
	SyN	45.40 ± 24.17	40.87 ± 25.35	32.45 ± 27.13	62.89 ± 30.55
	IterWarpImg	38.22 ± 23.21	29.42 ± 21.91	25.40 ± 23.83	59.85 ± 32.77
	IterWarpFlow	39.12 ± 20.77	29.74 ± 21.80	23.79 ± 21.15	63.84 ± 29.40
	Notre approche	$\mathbf{13.53 \pm 14.50}$	$\mathbf{15.49 \pm 16.37}$	$\mathbf{11.08 \pm 13.65}$	$\mathbf{14.04 \pm 19.59}$
	p-value	< 0.0001	< 0.0001	< 0.0001	0.0001

Le Tableau 6.5 présente les résultats pour l'estimation de la déformation myocardique et du ventricule droit (VD). Encore une fois, la méthode proposée obtient les meilleurs résultats et RAFT les deuxièmes meilleurs, même si Bioinformed prédit plus précisément que RAFT la phase présentant le pic de déformation circonférentielle pour le myocarde. Les méthodes SyN, IterWarpImg et IterWarpFlow donnent des résultats inférieurs aux méthodes reposant entièrement sur l'apprentissage profond, pour l'estimation de la déformation circonférentielle du myocarde mais sont plus précises pour la déformation du VD. Il est également important de noter que, pour toutes les méthodes, l'estimation de la déformation radiale myocardique est plus difficile que celle de la déformation circonférentielle.

TABLE 6.5 – Résultats de déformation.
Comparaison avec les méthodes de références (corrélation avec la valeur du pic de référence en fin de systole (r_v), corrélation avec les indices de référence (r_i), pentes des droites de régression correspondantes ($slope_v$, $slope_i$) et distance Euclidienne entre les courbes de déformation (dist)).

	Méthode	r_v	$slope_v$	r_i	$slope_i$	dist
Radiale	VM-MSE	0.52	0.46	0.79	0.96	1.97 ± 1.42
	VM-NCC	0.53	0.28	0.61	0.94	2.27 ± 1.61
	VM-Dif	0.56	0.48	0.78	0.92	1.88 ± 1.36
	Bioinformed	0.63	0.55	0.87	0.96	1.56 ± 1.37
	RAFT	0.77	0.40	0.88	0.98	1.67 ± 1.31
	SyN	0.43	0.29	0.47	0.92	2.91 ± 1.97
	IterWarpImg	0.49	0.34	0.53	1.04	3.84 ± 1.90
	IterWarpFlow	0.55	0.45	0.60	1.13	3.66 ± 1.72
	Notre approche	**0.81**	0.54	**0.91**	0.98	$\mathbf{1.31 \pm 1.03}$
circonférentiel	VM-MSE	0.77	0.88	0.83	1.00	0.28 ± 0.12
	VM-NCC	0.71	0.60	0.75	1.13	0.40 ± 0.19
	VM-Dif	0.78	0.95	0.87	1.04	0.27 ± 0.11
	Bioinformed	0.83	0.78	0.90	0.98	0.23 ± 0.10
	RAFT	0.88	0.80	0.87	0.97	$\mathbf{0.19 \pm 0.10}$
	SyN	0.54	0.49	0.63	1.08	0.72 ± 0.30
	IterWarpImg	0.63	0.54	0.79	1.04	0.74 ± 0.24
	IterWarpFlow	0.63	0.54	0.78	1.02	0.69 ± 0.23
	Notre approche	**0.90**	0.88	**0.95**	0.96	0.21 ± 0.09
VD	VM-MSE	0.63	0.63	0.72	0.71	0.49 ± 0.34
	VM-NCC	0.55	0.46	0.73	0.90	0.64 ± 0.42
	VM-Dif	0.67	0.69	0.72	0.69	0.47 ± 0.33
	Bioinformed	0.72	0.75	0.83	0.82	0.44 ± 0.28
	RAFT	0.77	0.70	0.80	0.75	0.39 ± 0.25
	SyN	0.28	0.30	0.62	0.85	0.88 ± 0.60
	IterWarpImg	0.69	0.62	0.82	0.78	0.47 ± 0.33
	IterWarpFlow	0.73	0.68	0.83	0.80	0.48 ± 0.32
	Notre approche	**0.89**	0.89	**0.85**	0.76	$\mathbf{0.30 \pm 0.19}$

La Figure 6.8 présente le score de Dice moyen, le SSIM moyen au sein du coeur et l'EPE moyenne par rapport à la distance à l'image ED en pourcentage du cycle cardiaque. Comme toutes les vidéos n'ont pas le même nombre d'images, les résultats ont été interpolés sur le nombre maximal d'images et moyennés sur toutes les séquences. Ces graphiques montrent que, à mesure que la distance à la phase télédiastolique augmente (et donc la quantité de mouvement), l'écart de performance entre l'approche proposée et les autres modèles s'agrandit, suggérant que notre modèle maintient mieux la qualité des flux de mouvement estimés. Cela résulte probablement de l'utilisation des images intermédiaires entre l'image ES et l'image ED, ce qui réduit la quantité de mouvement présente pour chaque flux estimé.



(a) (b) (c)

FIGURE 6.8 – Score de Dice moyen (a), SSIM moyen (b) et EPE moyenne (c) des méthodes de références et de notre méthode en fonction de la distance relative à l'image de fin diastole en pourcentage du cycle cardiaque. Mieux vu en zoomant.

6.3.5 Limitations

Avoir deux réseaux distincts f_1 et f_2 entraîne une consommation élevée de mémoire GPU, ce qui limite le nombre d'images utilisées pour l'entraînement. De plus, le premier réseau f_1 est uniquement utilisé pour calculer le mouvement entre des images adjacentes. En tant que tel, c'est un algorithme de flux optique classique qui pourrait être remplacé par un réseau pré-entraîné pour libérer de la mémoire GPU. Cela simplifierait également l'algorithme en réduisant le nombre de fonctions de coûts. Au lieu d'utiliser un réseau pré-entraîné, la taille de f_1 pourrait également être réduite en supprimant son décodeur de sorte que la fusion des flux soit effectuée à une résolution inférieure, entre les cartes de caractéristiques obtenues en sortie de l'encodeur de f_1 et de celui de f_2. Par ailleurs, ce modèle tient compte uniquement du mouvement allant de la première image à l'image précédente pour effectuer l'agrégation des flux de mouvements. Ce faisant, le réseau n'a pas connaissance des mouvements intermédiaires et n'exploite donc pas pleinement l'information temporelle. Enfin notre architecture ne calcule pas de volume de coûts, ce que l'on trouve pourtant souvent dans la littérature et qui doit permettre d'améliorer la qualité du flux estimé.

6.4 Conclusion

Ce travail a présenté une nouvelle méthode d'estimation de mouvement semi-supervisée pour suivre les pixels dans les séquences IRM. Comme généralement seules les images ED et ES ont des annotations de segmentation de vérité terrain, la plupart des travaux utilisent des méthodes non supervisées apprises sur toutes les images des séquences ou des méthodes entièrement supervisées apprises uniquement sur ED et ES. Dans les deux cas, les résultats ne sont pas optimaux. Nous proposons donc un algorithme semi-supervisé qui utilise toutes les images de la séquence ainsi que les annotations de segmentation de vérité terrain pour ED et ES. Un processus itératif d'agrégation du mouvement a été présenté pour estimer le flux de mouvement entre la première image de la séquence vidéo et toutes les autres images. Pour effectuer cette tâche, un premier réseau a été mis au point pour prédire le mouvement entre des images adjacentes, tandis qu'un second réseau est chargé de fusionner les flux sur la séquence vidéo. L'approche proposée surpasse Voxelmorph, RAFT, Bioinformed, et les méthodes de "warping" itératif présentées dans le cadre de cette étude, en particulier lorsque le mouvement entre deux images est important.

Chapitre 7

Réseau à mémoire et carte de distances

7.1 Motivation

L'architecture présentée au chapitre 6 présente certaines limitations. La principale repose sur l'utilisation de 5 composantes dans la fonction de coût. Les changements architecturaux et l'utilisation de nouveaux types de données (comme c'est le cas dans cette partie avec les cartes de distance) nécessitent de modifier les poids de chacune de ces composantes, rendant l'algorithme peu adaptable du fait de la difficulté de fixer ces 5 poids de façon optimale. De plus, l'utilisation de deux réseaux séparés nécessite d'importantes ressources mémoires et de nombreux paramètres qui pourraient être alloués à d'autre modules (par exemple un convGRU pour exploiter l'information temporelle, comme présenté en section 7.2.4). Par ailleurs, le modèle de fusion du chapitre 6 s'appuie uniquement sur l'information temporelle entre la première image de la séquence et l'image précédente et ne tient donc pas compte des mouvements entre images intermédiaires. Enfin, ce dernier modèle ne calcule pas de volume de coût, alors qu'il a pourtant été montré que cela aide à prédire des mouvements plus précis (Fischer et al. 2015 ; Sun, Yang et al. 2018 ; Hui, Tang et Loy 2018 ; Teed et Deng 2020).

Par conséquent, ce travail décrit une nouvelle méthode d'estimation de mouvement semi-supervisée pour suivre les pixels dans une séquence vidéo. L'architecture populaire de "Video Object Segmentation" (VOS) présentée par Oh et al. 2019 est adaptée pour effectuer cette estimation. Plus précisément, à chaque itération, un "query encoder" extrait des caractéristiques relatives à l'image actuelle, tandis que les informations sur le mouvement entre la première image et l'image actuelle sont compressées par un "memory encoder". L'algorithme fonctionne de manière itérative pour agréger progressivement le mouvement afin d'estimer un mouvement plus large entre la première et toutes les autres images de la séquence. Comme pour la méthode précédente, les vidéos sont séparées en deux parties au niveau de l'image de fin systole en se basant sur la nature cyclique du signal cardiaque et chaque partie est traitée indépendamment par le réseau. L'algorithme peut être appliqué à des séquences de longueurs variables sans nécessiter de mécanisme de fenêtre glissante.

En outre, ce travail introduit des cartes de distance créées à partir des annota-

tions de segmentation. Elles sont utilisées pour pondérer les pixels de la fonction de coût, incitant le réseau à se concentrer sur les contours des structures cardiaques. Les performances de la version entièrement non supervisée de la méthode sont comparées à celles de la version semi-supervisée (annotations de segmentation en ED et ES) et entièrement supervisées (annotations de segmentations de toutes les images). L'algorithme est entraîné et testé sur le jeu de données Quorum contenant des vidéos cardiaques (c.f. 2.5.5). Comme ce jeu de données a été traité par le logiciel CardioTrack, nous disposons de l'information de déplacement des points de contour ainsi que des annotations de segmentation pour toutes les images et pour toutes les structures cardiaques, ce qui nous permet de mesurer les performances de la méthode pour suivre les pixels dans le temps. Les déformations du ventricule gauche et droit sont également calculées et comparées aux valeurs de références générées par CardioTrack.

7.2 Méthode

7.2.1 Architecture proposée

Avant de revenir en détails sur l'architecture utilisée dans ce chapitre, il est nécessaire de rappeler le fonctionnement des réseaux à mémoire. Ces derniers ont principalement été utilisés pour la segmentation d'image mais de récents travaux ont porté sur l'adaptation de cette architecture à l'estimation du flux optique.

Le travail fondateur présenté par Oh et al. 2019 a introduit une nouvelle architecture de réseau pour effectuer la segmentation vidéo d'objets qui a été largement adoptée dans des études ultérieures. L'algorithme traite séquentiellement les images d'une vidéo. A chaque itération, l'image actuelle est encodée par un "query encoder" tandis que les images précédentes et leurs masques de segmentation prédits ou de vérité terrain associés sont encodés par un "memory encoder". À la résolution la plus basse, les caractéristiques courantes interagissent avec les caractéristiques de la mémoire pour récupérer des informations pertinentes via une opération de "space-time memory read". Les études suivantes ont essayé d'améliorer l'opération de lecture de la mémoire en concentrant le processus de correspondance dense sur les pixels les plus pertinents, avec une fenêtre gaussienne (Seong, Hyun et Kim 2020), un processus de sélection topK (Cheng, Tai et Tang 2021a), ou en définissant une région d'intérêt autour du masque des images passées (Xie, Yao et al. 2021). Seong, Oh et al. 2021 réutilisent les pixels de correspondance topK à basse résolution pour guider le processus d'attention à des résolutions plus élevées, bénéficiant ainsi de l'information multi-échelle de la mémoire. Les "positional encoding" sinusoïdaux ont également été utilisés pour permettre aux caractéristiques actuelles de suivre des pixels similaires situés aux mêmes positions dans la mémoire (Hu, Zhang et al. 2021). Cheng, Tai et Tang 2021b ont utilisé une mesure de similarité L2 plutôt que le produit scalaire habituel dans le processus de lecture de la mémoire. En effet, ils ont montré que cette distance donne plus de poids aux valeurs non dominantes, évitant que les pixels à forte réponse ne suppriment les autres réponses. Plutôt que d'utiliser l'opération de "space-time memory read", certaines études ont utilisé des couches transformer soit pour interagir avec la mémoire en utilisant l'attention croisée (Yang, Wei et Yang 2021), soit en effectuant une self-attention entre un sous-ensemble de

tokens dans la mémoire et les caractéristiques actuelles (Duke et al. 2021). Liu, Yu et al. 2022 s'intéressent à la qualité des cartes de caractéristiques stockées dans la mémoire plutôt qu'à l'opération de lecture de mémoire. Plus précisément, ils réutilisent l'encodeur de mémoire pour décider si un masque de segmentation prédit et son image correspondante doivent être encodés dans la mémoire. Au lieu d'utiliser une seule mémoire pour stocker toutes les images, Cheng et Schwing 2022 ont utilisé trois mémoires de tailles fixes différentes qui sont mises à jour à des fréquences différentes. Dong et Fu 2024 appliquent l'architecture VOS à l'estimation du flux optique et rééchelonnent les poids d'attention dans l'opération de lecture de la mémoire de sorte que le réseau puisse s'adapter plus facilement à une séquence contenant un nombre d'images différent en inférence.

Nous reprenons ici les mêmes notations que dans le chapitre 6 en ayant pour objectif d'estimer le mouvement de chaque pixel $F_{1,t}$ entre la première image I_1 et chacune des autres images $I_t \; \forall t \in [2; T]$ d'une séquence S. Comme démontré au chapitre précédent avec l'introduction des deux méthodes « naïves » IterWarpImg et IterWarpFlow, l'application répétée d'algorithmes d'estimation du flux optique ou de recalage d'images avec un mécanisme de fenêtre glissante ou par composition successive des mouvements peut conduire à des performances sous-optimales pour deux raisons principales. Premièrement, cette méthode ne tient pas compte de la relation temporelle entre les images non adjacentes, ce qui peut potentiellement conduire à une estimation de mouvements irréguliers. Deuxièmement, cette approche aurait probablement du mal à estimer ces mouvements à mesure que la distance par rapport à la première image augmente. Sur la base de cette hypothèse, nous introduisons un algorithme, inspiré de l'architecture présentée dans Oh et al. 2019, qui fonctionne de manière itérative et agrège progressivement les informations de mouvement entre les images adjacentes I_{t-1} et I_t en s'appuyant sur le mouvement estimé $F_{1,t-1}$. Ce faisant, la quantité de mouvement à estimer est faible à chaque itération, rendant la tâche plus facile pour le réseau. De plus, l'utilisation de $F_{1,t-1}$ permet de prédire des déplacements temporellement cohérents. Par ailleurs, comme détaillé en section 7.2.7, du fait de l'usage d'un seul décodeur, cet algorithme repose sur une fonction de coût comprenant 3 composantes, contre 5 pour l'approche du chapitre 6. Cela rend plus aisée l'utilisation des cartes de distances décrites en section 7.2.6.

7.2.2 Jeu de données et modalités d'entrainement et d'inférence

Nous utilisons le même jeu de données que celui décrit dans le chapitre 6 avec la même séparation pour les données d'entrainement, de validation, et de test. Les séquences sont traitées de la même façon en entrainement et durant l'inférence et les modèles sont également entrainés pendant 180 epochs. Tout comme au chapitre 6, les séquences contiennent $N = 12$ images en entrainement.

7.2.3 Agrégation itérative du mouvement

L'architecture du réseau, similaire à Oh et al. 2019, est illustrée Figure 7.1. De même qu'au chapitre 4, l'architecture est similaire à celle de U-Net avec des enco-

deurs et un décodeur introduisant plusieurs résolutions reliées, à chaque étage, par des connexions. Plus précisément, à chaque itération sur une image de la séquence, un "query encoder" enc_Q est utilisé pour extraire les caractéristiques Q_t de l'image I_t, tandis qu'un "memory encoder" enc_M extrait les caractéristiques M_{t-1} liées aux prédictions passées. Contrairement à Oh et al. 2019, nous ne stockons pas les cartes de caractéristiques des prédictions passées dans une mémoire, mais nous nous appuyons uniquement sur la dernière carte de caractéristiques encodée par enc_M. En effet, comme expliqué dans Dong et Fu 2024, et comme confirmé par nos expériences, nous n'avons constaté aucune amélioration en utilisant plus d'une carte de caractéristiques en mémoire. À la résolution la plus basse, Q_t porte attention à Q_{t-1} et M_{t-1} avant d'être sur échantillonné par un décodeur convolutionnel.

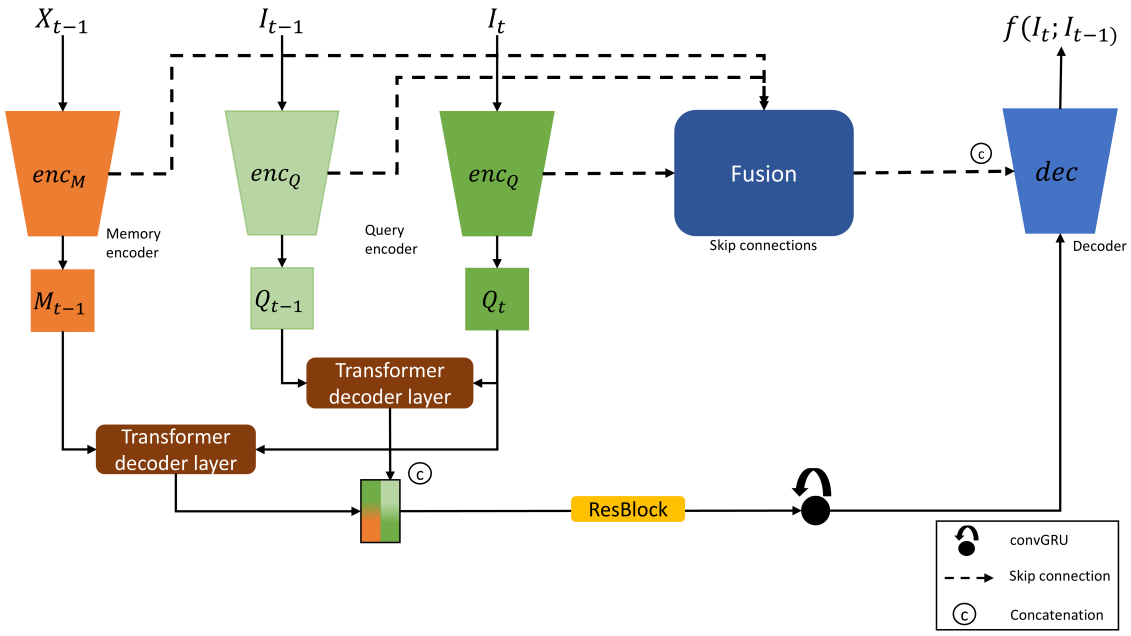


FIGURE 7.1 – Architecture du réseau

Afin d'estimer le flux de mouvement de I_1 vers toutes les autres images, l'algorithme proposé traite séquentiellement les images et agrège les mouvements inter-images. L'information de mouvement entre I_1 et I_{t-1} est encodée par enc_M et notée $M_{t-1} \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times D}$ avec $\mathcal{H} = \frac{H}{8}$, $\mathcal{W} = \frac{W}{8}$, et D le nombre de caractéristiques. Plus spécifiquement, à chaque itération t, enc_M traite un tenseur à plusieurs canaux X_{t-1} défini comme suit :

$$X_{t-1} = [I_1, I_{t-1}, F_{1,t-1}, R_{t-1,1}, E_{t-1,1}] \quad \forall\, t \in [2, T] \tag{7.1}$$

où [...] fait référence à l'opération de concaténation le long de la dimension des canaux. $R_{t-1,1}$ et $E_{t-1,1}$ sont respectivement l'image I_{t-1} déformée avec le flux $F_{1,t-1}$ et l'erreur de cette déformation par rapport à I_1. Ils sont donc définis de la façon suivante :

$$R_{t-1,1} = I_{t-1} \circ F_{1,t-1} \quad E_{t-1,1} = R_{t-1,1} - I_1 \;\forall\, t \in [3, T] \tag{7.2}$$

où $\circ$ est la fonction de "warping". L'algorithme est initialisé avec $F_{1,1} = 0$, $R_{1,1} = I_1$, and $E_{1,1} = 0$. $R_{t-1,1}$ et $E_{t-1,1}$ sont transmis à enc_M pour atténuer la propagation des erreurs et permettre au réseau d'ajuster les prédictions de mouvement.

De plus, à chaque itération, enc_Q prend en entrée I_t et extrait, à la résolution la plus basse, les caractéristiques $Q_t \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times D}$. Des couches transformers permettent alors d'effectuer de la cross-attention avec M_{t-1} et Q_{t-1}, avant que le tenseur ne soit décodé en intégrant les "skip connections" dans un module de fusion. À chaque itération, le réseau f produit un flux résiduel qui est ajouté au mouvement courant par rapport à la première image :

$$F_{1,t} = F_{1,t-1} + f(I_t, I_{t-1}) \quad \forall t \ \in [2, T] \tag{7.3}$$

7.2.4 Intégration des mouvements passés

À la résolution la plus basse, à chaque itération, Q_t prête attention à la fois à Q_{t-1} et à M_{t-1} en utilisant une couche transformer $\mathcal{T}$ similaire à celle introduite dans Vaswani et al. 2017. Plus précisément, à chaque itération, Q_t utilise $\mathcal{T}$ pour produire B_1 et B_2 qui intègrent respectivement les informations de Q_{t-1} et M_{t-1} :

$$B_1 = \mathcal{T}(Q_t, Q_{t-1}, Q_{t-1}) \tag{7.4}$$

$$B_2 = \mathcal{T}(Q_t, Q_1, M_{t-1}) \tag{7.5}$$

où $\mathcal{T}$ prend en entrée Q, K et V et peut être décrite comme la succession des opérations suivantes :

$$\begin{aligned} Z_1 &= \text{LayerNorm}(Q + A(Q, Q, Q)) \\ Z_2 &= \text{LayerNorm}(Z_1 + A(Z_1, K, V)) \\ \mathcal{T}(Q, K, V) &= \text{LayerNorm}(Z_2 + \text{MLP}(Z_2)) \end{aligned} \tag{7.6}$$

où A est la fonction d'attention décrite par Vaswani et al. 2017. Notez que, contrairement à ce que l'on trouve habituellement dans la littérature pour la cross-attention, une key et une value différentes sont utilisées dans l'équation 7.5. Plus précisément, nous effectuons une correspondance avec la key Q_1 plutôt que M_{t-1}. Cela est dû au fait que Q_1 est également généré par enc_Q, garantissant ainsi une correspondance plus précise. De plus, M_{t-1} est aligné spatialement avec Q_1 puisque enc_M prend en entrée $F_{1,t-1}$ qui provient de I_1. Par conséquent, nous supposons qu'il n'y a pas de désalignement spatial entre la key Q_1 et la value M_{t-1}.

$B_1 \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times D}$ et $B_2 \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times D}$ sont ensuite concaténés le long de la dimension des caractéristiques pour former le tenseur $\mathbf{x}_t \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times 2D}$, puis passés à un bloc résiduel (resBlock) avec une taille de filtre de 3×3 pixels pour réduire le nombre de caractéristiques à D. Ensuite, $\mathbf{x}_t$ est passé à un convGRU (Ballas et al. 2016) définit ci-dessous d'une façon similaire à Teed et Deng 2020 :

$$\begin{aligned} z_t &= \sigma(\text{Conv3x3}([\mathbf{h}_{t-1}, \mathbf{x}_t], \mathbf{W}_z)) \\ r_t &= \sigma(\text{Conv3x3}([\mathbf{h}_{t-1}, \mathbf{x}_t], \mathbf{W}_r)) \\ \tilde{\mathbf{h}_t} &= \tanh\left(\text{Conv3x3}([\mathbf{r}_t \odot \mathbf{h}_{t-1}, \mathbf{x}_t], \mathbf{W}_h)\right) \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}_t} \end{aligned} \tag{7.7}$$

où $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_h \in \mathbb{R}^{D\times 2D\times 3\times 3}$ sont des matrices de poids apprenables, $\mathbf{h}_t \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}\times D}$ est le "hidden state" produit à l'itération t. σ est la fonction sigmoïde, [...] désigne l'opération de concaténation le long de la dimension des canaux et $\odot$ est le produit de Hadamard. Nous initialisons $\mathbf{h}_1 = 0$. Le convGRU permet au réseau d'accéder aux informations de mouvement des images précédentes afin de produire des résultats temporellement cohérents. La sortie de la couche convGRU $\mathbf{h}_t$ est ensuite sur-échantillonnée par un décodeur convolutionnel.

7.2.5 Décodage et skip connections

Notre décodeur est similaire au décodeur convolutionnel présent dans l'architecture U-net (Ronneberger, Fischer et Brox 2015), mais utilise des connexions résiduelles. De plus, au lieu de simplement concaténer les cartes de caractéristiques intermédiaires de enc_Q avec les cartes de caractéristiques de même résolution du décodeur, les skip connections intègrent également à chaque résolution les caractéristiques Q_{t-1} produites par enc_Q à l'itération précédente et les caractéristiques M_{t-1} générées par enc_M. Ce processus est décrit Figure 7.2.
Pour chaque résolution $r \in \{1, 2, 4\}$, nous calculons d'abord le volume de coût $C_s^r \in \mathbb{R}^{\frac{W}{rs} \times \frac{H}{rs} \times (2d+1)^2}$, pour tous les patchs de Q_t^r séparés par une stride s, et un nombre limité de patchs de Q_{t-1}^r situés dans un voisinage de rayon d. Plus précisément, pour un patch de coordonnées $\mathbf{x_1}$ dans Q_t^r et $\mathbf{x_2}$ dans Q_{t-1}^r, le coût c est calculé comme suit :

$$c(\mathbf{x_1}, \mathbf{x_2}) = \frac{Q_t^r(\mathbf{x_1}) \cdot Q_{t-1}^r(\mathbf{x_2})}{\|Q_t^r(\mathbf{x_1})\| \|Q_{t-1}^r(\mathbf{x_2})\|} \tag{7.8}$$

avec $|\mathbf{x_1} - \mathbf{x_2}| < 2d + 1$. En raison de la contrainte de mémoire GPU, nous utilisons $s = \frac{4}{r}$. C_s^r est ensuite sur-échantillonné à l'aide d'une convolution transposée pour récupérer la résolution originale et passé à travers un bloc résiduel pour extraire les caractéristiques $Q_{t;t-1}^r \in \mathbb{R}^{\frac{W}{r} \times \frac{H}{r} \times D}$. Ensuite, nous concaténons $Q_{t;t-1}^r$ avec M_{t-1}^r le long de la dimension des caractéristiques avant de réduire le nombre de caractéristiques avec un bloc résiduel pour obtenir le tenseur $\mathrm{QM}_{t;t-1}^r \in \mathbb{R}^{\frac{W}{r} \times \frac{H}{r} \times D}$:

$$\mathrm{QM}_{t;t-1}^r = \mathrm{Resblock}([Q_{t;t-1}^r, M_{t-1}^r]) \tag{7.9}$$

$\mathrm{QM}_{t;t-1}^r$ est ensuite concaténé avec la carte de caractéristiques de même résolution correspondante du décodeur.

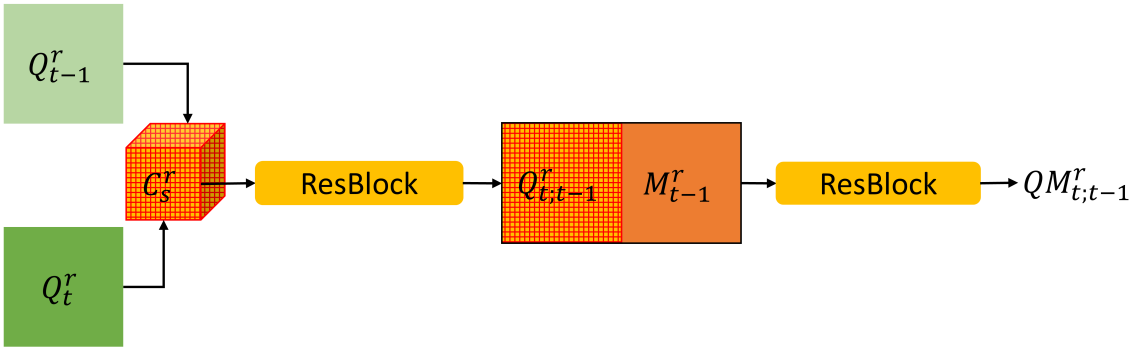


FIGURE 7.2 – Fusion des cartes de caractéristiques intermédiaires avant concaténation avec les cartes du décodeur pour une seule résolution.

7.2.6 Carte de distance pour la pondération de la fonction de coût

Pour chaque image du jeu de données, nous calculons une carte de distance à la cavité ventriculaire droite (VD), l'endocarde (ENDO) et l'épicarde (EPI) à partir des annotations de segmentation de vérité terrain. Cette carte est utilisée pour pondérer les fonctions de coût en fonction de la distance à ces structures anatomiques. L'idée est que le réseau puisse se concentrer sur les régions importantes pour le calcul de la

déformation. Les segmentations de vérité terrain contiennent des annotations pour le VD, la cavité ventriculaire gauche (VG) et le myocarde (MYO), de façon similaire à ce qui est couramment trouvé dans la segmentation d'images médicales cardiaques (Bernard et al. 2018 ; Campello et al. 2021 ; Radau et al. 2009). Le processus de calcul des cartes de distance à partir des annotations de segmentation de vérité terrain est effectué avant l'entraînement. Les contours du myocarde et du VD sont extraits en utilisant un gradient morphologique et la distance x de chaque point de contour à la structure anatomique la plus proche est calculée. Enfin, la dérivée de la fonction sigmoïde est utilisée pour rééchelonner les distances de façon à obtenir la carte de distance $\delta \in \mathbb{R}^{H \times W}$:

$$\delta = \frac{4e^{-x}}{(1 + e^{-x})^2} \tag{7.10}$$

Dans l'équation ci-dessus, la multiplication par 4 permet d'avoir une valeur maximum de 1 sur les points de contour. Dans nos expériences, nous étudions l'impact de l'exponentiation des cartes de distance à la puissance k. Notons δ^k la carte obtenue en élevant chaque pixel de δ à la puissance k. Avec $k = 0$, toutes les valeurs de la carte de distance sont égales à 1 et par conséquent, aucune pondération spatiale n'est appliquée aux fonctions de coût. Augmenter k entraîne une différence plus importante entre les distances élevées et les distances faibles, accordant ainsi une importance accrue aux erreurs près des contours de l'endocarde, de l'épicarde et du VD relativement aux autres pixels. La Figure 3.5 montre des exemples de carte de distance. Dans la suite, nous notons δ_t la carte de distance correspondant à la t ème image de la séquence.

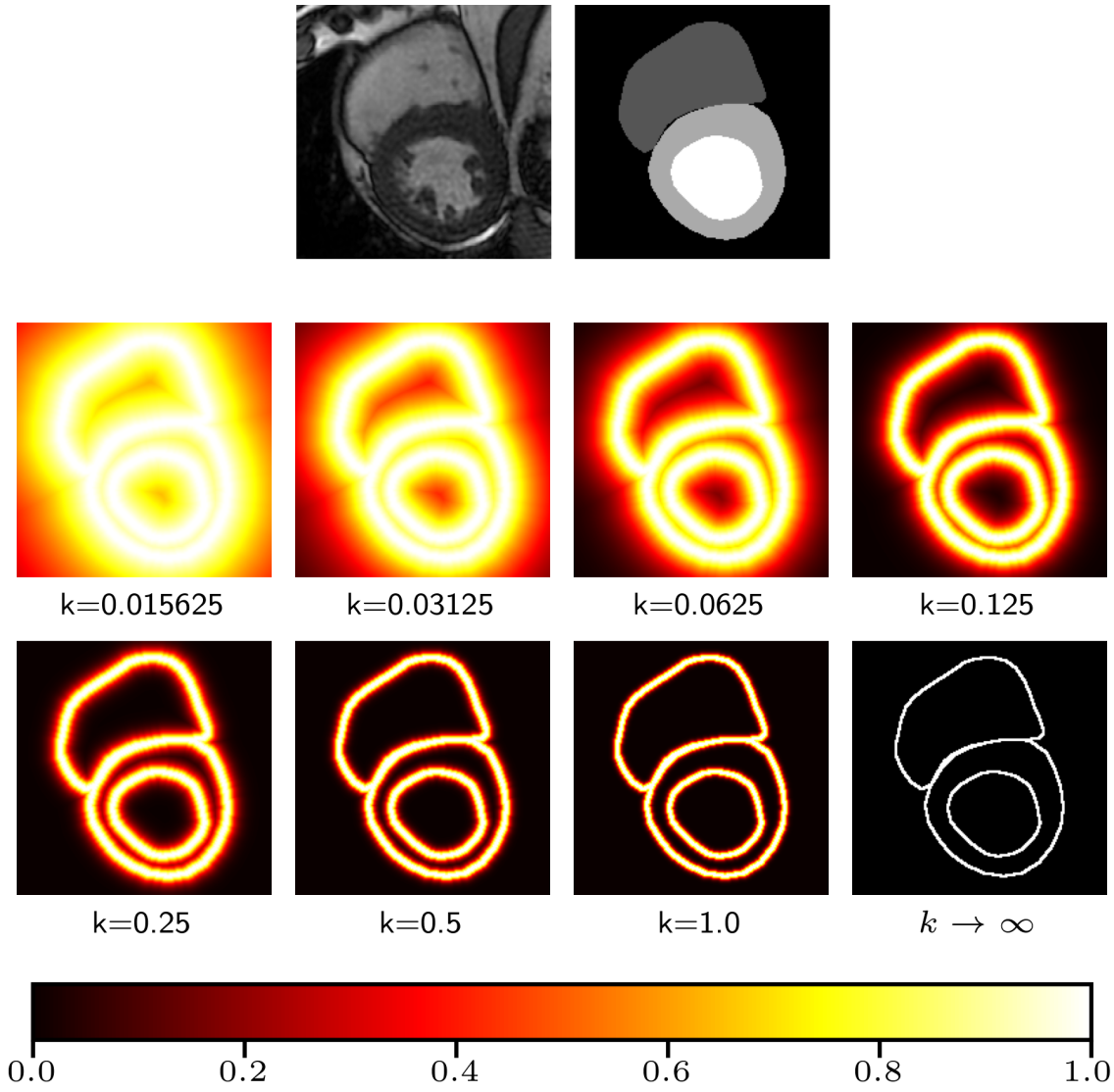


FIGURE 7.3 – Première ligne : image et sa segmentation de vérité terrain. Lignes suivantes : Cartes de distance δ^k avec différents exposants k.

7.2.7 Fonctions de coût

La fonction de coût est composée de plusieurs parties.

La première évalue la similarité entre I_{ED} et les images du cycle cardiaque I_t recalées sur I_{ED} à partir du flux estimé $F_{1,t}$:

$$\mathcal{L}^k_{sim}(I_1, R_{t,1}) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \delta^k_1(p) \times (1 - \mathrm{NCC}(I_1, R_{t,1})(p)) \tag{7.11}$$

où Ω est l'ensemble des pixels p d'une image et $\delta^k_1(p)$ est la valeur de la carte de distance de l'image ED (t=1) à la position p, élevée à la puissance k. NCC fait référence à la fonction de corrélation croisée normalisée (équation 6.6).

Afin de garantir que des flux de mouvement spatialement lisses soient générés, le gradient des flux dans les directions x et y est minimisé. La fonction utilisée

pour mesurer la régularité spatiale des flux de mouvement spécifique $F_{1,t}$ est définie comme suit :

$$\mathcal{L}^k_{smooth}(F_{1,t}) = \frac{1}{|\Omega|} \sum_{p \in \Omega} \delta^k_1(p) \times \|\nabla F_{1,t}(p)\|^2_2 \tag{7.12}$$

avec $\nabla F_{1,t}(p) = (\frac{\partial F_{1,t}(p)}{\partial x}, \frac{\partial F_{1,t}(p)}{\partial y})$.

De plus, pour un indice temporel t, une erreur de segmentation $\mathcal{L}_{seg}$ entre la segmentation de vérité terrain de l'image I_1, notée Y_1 et la segmentation à l'instant t obtenue en déformant le label de segmentation de l'image I_t, noté Y_t, à l'aide du flux $F_{1,t}$ peut se définir de la façon suivante :

$$\mathcal{L}_{seg}(Y_1, Y_t, F_{1,t}) = f_{seg}(Y_1, Y_t \circ F_{1,t}) \tag{7.13}$$

f_{seg} est une combinaison de l'erreur de Dice et de l'entropie croisée, comme proposé dans Isensee et al. 2018 (même fonction de coût de segmentation qu'au chapitre 4 et 6). Notez que nous n'utilisons pas la carte de distance δ pour pondérer $\mathcal{L}_{seg}$.
Enfin, la fonction de coût finale $\mathcal{L}$ est définie comme suit :

$$\mathcal{L}(I, R, F, Y) = \left(\frac{1}{N-1} \sum_{t=2}^{N} \lambda_1 \mathcal{L}^k_{sim}(I_1, R_{t,1}) + \lambda_2 \mathcal{L}^k_{smooth}(F_{1,t}) \right) + \lambda_3 \mathcal{L}_{seg}(Y_1, Y_N, F_{1,N}) \tag{7.14}$$

où Y_N est le label de segmentation de l'image de fin-systole qui correspond à la dernière image de la séquence d'entrainement. Seules les segmentations à ED et ES sont utilisées lors de l'apprentissage. Dans nos expériences, les meilleurs résultats ont été obtenus avec $\lambda_1 = 0.5, \lambda_2 = 1.0, \lambda_3 = 0.01$. Dans $\mathcal{L}_{sim}$ et $\mathcal{L}_{smooth}$, tous les flux de mouvement intermédiaires sont utilisés, et pas seulement le dernier flux estimé (rappelons que toutes les séquences d'apprentissage sont composées de N images, comme présenté au chapitre 6).

Que ce soit pour $\mathcal{L}^k_{sim}$, $\mathcal{L}^k_{smooth}$, ou $\mathcal{L}_{seg}$, le recalage est fait en utilisant les mouvements qui partent de la première image. Seules les segmentations en ED et ES et la carte de distance en ED sont donc utilisées. Par conséquent, notre travail prend place dans un cadre semi-supervisé.

7.2.8 Étude ablative

Nous étudions l'impact de la variation de k dans les équations 7.11 et 7.12 et mesurons également la performance de notre modèle entraîné de manière entièrement **non supervisée** avec $k = 0$ et $\lambda_3 = 0$. Une comparaison avec la version entièrement **supervisée** de l'algorithme est également présentée. Cette dernière version utilise la composante $\mathcal{L}_{seg}$ pour tous les mouvements et non uniquement pour le mouvement allant de fin systole à fin diastole. Plus précisément, la fonction de coût du modèle entièrement supervisé devient :

$$\mathcal{L}_{sup}(I, R, F, Y) = \frac{1}{N-1} \sum_{t=2}^{N} \lambda_1 \mathcal{L}^k_{sim}(I_1, R_{t,1}) + \lambda_2 \mathcal{L}^k_{smooth}(F_{1,t}) + \lambda_3 \mathcal{L}_{seg}(Y_1, Y_t, F_{1,t}) \tag{7.15}$$

Pour le modèle entièrement supervisé, les meilleures performances sont obtenues avec $\lambda_1 = 0.5$, $\lambda_2 = 1.0$ et $\lambda_3 = 0.1$.

Les résultats du modèle présenté au chapitre précédent avec et sans carte de distance sont également exposés.

Afin d'utiliser la carte de distance de la première image δ_1^k dans le modèle semi-supervisé du chapitre précédent, l'equation 6.10 peut se réecrire de la façon suivante :

$$\begin{aligned}\mathcal{L}(I,R,F,Y) = &\frac{1}{N-1}\sum_{t=2}^{N}\lambda_1\mathcal{L}^0_{sim}(I_{t-1},R_{t,t-1}) + \lambda_2\mathcal{L}^0_{smooth}(F_{t-1,t}) \\ &+ \frac{1}{N-1}\sum_{t=2}^{N}\lambda_3\mathcal{L}^k_{sim}(I_1,R_{t,1}) + \lambda_4\mathcal{L}^k_{smooth}(F_{1,t}) \\ &+ \lambda_5\mathcal{L}_{seg}(Y_1,Y_N,F_{1,N})\end{aligned} \tag{7.16}$$

Les meilleurs résultats sont obtenus avec $\lambda_1 = 0.5$, $\lambda_2 = 1.0$, $\lambda_3 = 1.5$, $\lambda_4 = 3.0$ et $\lambda_5 = 1.0$

Enfin, nous présentons également les résultats de l'estimation de la déformation globale sans suivi, à l'aide de l'algorithme de segmentation présenté au chapitre 4. En inférence, les points de contours sont extraits à partir des segmentations prédites et passés au logiciel CardioTrack pour calculer les déformations.

7.2.9 Détails d'implémentation

Comme pour le chapitre précédent, nous utilisons $N = 12$ où N est le nombre d'images pour chaque séquence durant l'entrainement. Le modèle est également implémenté avec Pytorch et un GPU NVIDIA V100 de 16G. Nous utilisons le même optimiseur AdamW avec un pas d'apprentissage et "weight decay" de 10^-4. Comme dans le chapitre précédent, le modèle est entrainé pour 180 epochs et 250 itérations par epoch. De même, le nombre de dimension est doublé à chaque couche de sous-échantillonnage de l'encodeur et divisé par deux à chaque sur-échantillonnage dans le décodeur. Nous utilisons également une taille de batch de 1 et des couches de "group-normalisation" (avec 8 groupes) sont utilisées dans tout le réseau. Notre modèle contient environ $25M$ de paramètres, a peu près autant que le modèle du chapitre 6. Cela permet de faciliter les comparaisons entre les deux modèles. Le détail de l'architecture est disponible en annexe (section .3).
Contrairement au modèle du chapitre précédent, l'architecture présentée ici utilise $D = 256$ dimensions à la plus basse résolution. Le MLP des transformers projette la carte de caractéristiques à la dimension 2048.

7.3 Résultats et discussion

7.3.1 Exponentiation des cartes de distance

La Figure 7.4 présente l'EPE, le pourcentage moyen de pixels du coeur avec un Jacobien négatif et le SSIM au sein du coeur en fonction de l'exposant k de la carte de distance. Il est apparent que plus k augmente, et plus l'EPE diminue, indiquant une plus grande précision pour l'estimation des points de contours. Cependant, cela s'accompagne également d'une réduction du SSIM, suggérant une plus faible similarité avec l'image ED. Cela montre qu'à mesure que k augmente, le réseau se concentre

davantage sur les contours des structures cardiaques, au détriment des autres zones de l'image. On observe également une réduction progressive du % $det(J_F) \leq 0$ pour k allant de 0.015625 à 0.25 puis, une augmentation progressive à partir de $k = 0.25$, indiquant une plus faible régularité du flux optique. Nous choisissons $k = \infty$ pour la suite car cela permet d'obtenir la plus faible EPE, ce qui est le principal objectif de ce travail puisque nous souhaitons estimer la déformation cardiaque de la façon la plus précise possible. Cela signifie que les fonctions de coûts ne sont estimées que sur les points de contour (le coefficient devient 0 pour les autres points). Pour obtenir de meilleurs résultats de recalage, on aurait sélectionné une carte de distance avec une plus faible valeur de k.

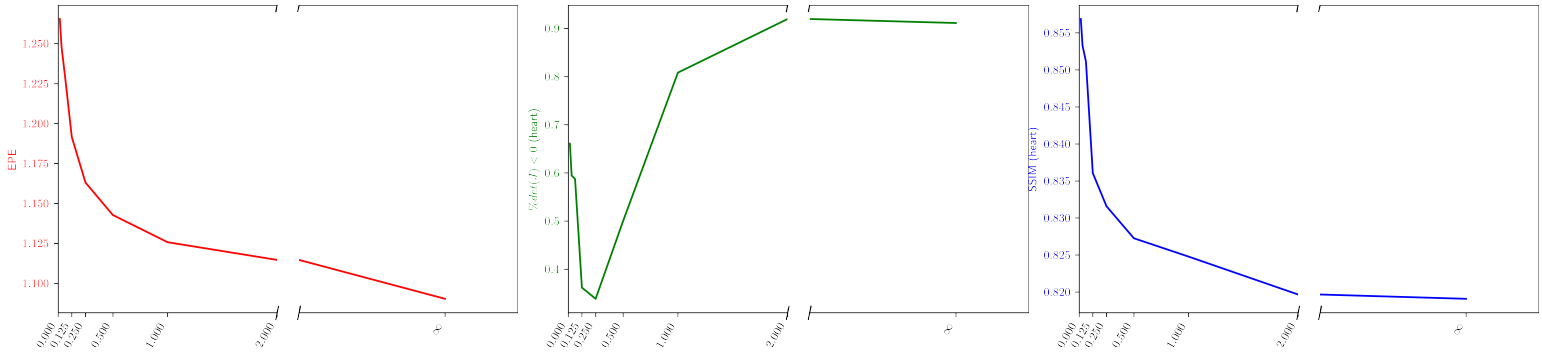


FIGURE 7.4 – EPE (a), pourcentage moyen de pixels du coeur avec un Jacobien négatif (b) et SSIM (c) en fonction de l'exposant k de la carte de distance. Mieux vu en zoomant.

7.3.2 Comparaison avec les méthodes supervisées et non supervisées

Les tableaux 7.1 et 7.2 présentent les résultats de suivi de points moyennés sur toutes les phases des séquences ou uniquement pour la phase ES respectivement pour le modèle de ce chapitre dans sa version non supervisée, semi-supervisée et entièrement supervisée. Le tableau 7.3 présente les résultats d'estimation de la déformation.

La version entièrement supervisée du modèle à mémoire de ce chapitre présente des résultats très légèrement supérieurs à la version semi-supervisée, indiquant que l'utilisation des annotations de segmentation de toutes les images de la séquence apporte peu par rapport à la seule utilisation des labels de segmentation de ED et ES, et ce, que ce soit pour le suivi de points ou pour l'estimation de la déformation. On constate par contre un gain notable par rapport à la version non supervisée, justifiant donc le contexte non supervisé dans lequel se placent ces travaux.

TABLE 7.1 – Résultats de suivi de points pour toutes les phases.
EPE ± écart-type, F1-all ± écart-type. Les p-values sont calculées entre la méthode supervisée et semi-supervisée. EPI = Epicarde, ENDO = Endocarde. Le symbole ∗ indique l'emploi des cartes de distance.

	Méthode	Moyenne	VD	EPI	ENDO
EPE (pixels)	Non supervisée	1.30 ± 0.51	1.40 ± 0.63	1.10 ± 0.43	1.39 ± 0.67
	Semi-supervisée ∗	1.09 ± 0.38	1.19 ± 0.46	1.02 ± 0.38	1.07 ± 0.42
	Supervisée ∗	$\mathbf{1.05 \pm 0.38}$	$\mathbf{1.14 \pm 0.46}$	$\mathbf{0.99 \pm 0.37}$	$\mathbf{1.01 \pm 0.40}$
	p-value	< 0.0001	< 0.0001	< 0.0001	< 0.0001
F1-all (%)	Non supervisée	8.47 ± 7.79	9.41 ± 9.29	5.52 ± 6.37	10.48 ± 11.23
	Semi-supervisée ∗	5.04 ± 5.38	6.49 ± 7.39	4.19 ± 4.96	4.45 ± 6.39
	Supervisée ∗	$\mathbf{4.45 \pm 5.01}$	$\mathbf{5.87 \pm 7.15}$	$\mathbf{3.85 \pm 4.70}$	$\mathbf{3.62 \pm 5.76}$
	p-value	< 0.0001	< 0.0001	0.0001	< 0.0001

TABLE 7.2 – Résultats de suivi de points uniquement pour la phase ES.
EPE ± écart-type, F1-all ± écart-type. Les p-values sont calculées entre la méthode supervisée et semi-supervisée. EPI = Epicarde, ENDO = Endocarde. Le symbole ∗ indique l'emploi des cartes de distance.

	Méthode	Moyenne	VD	EPI	ENDO
EPE (pixels)	Non supervisée	2.21 ± 1.05	2.39 ± 1.49	1.67 ± 0.75	2.56 ± 1.55
	Semi-supervisée ∗	1.62 ± 0.62	1.77 ± 0.80	1.49 ± 0.61	1.61 ± 0.71
	Supervisée ∗	$\mathbf{1.54 \pm 0.59}$	$\mathbf{1.67 \pm 0.79}$	$\mathbf{1.45 \pm 0.57}$	$\mathbf{1.50 \pm 0.69}$
	p-value	< 0.0001	< 0.0001	0.0011	< 0.0001
F1-all (%)	Non supervisée	22.13 ± 17.57	21.82 ± 18.76	14.22 ± 16.23	30.37 ± 28.81
	Semi-supervisée ∗	11.91 ± 12.31	14.27 ± 15.43	9.78 ± 12.06	11.69 ± 16.13
	Supervisée ∗	$\mathbf{10.22 \pm 11.31}$	$\mathbf{12.23 \pm 14.57}$	$\mathbf{8.94 \pm 10.98}$	$\mathbf{9.49 \pm 15.07}$
	p-value	< 0.0001	< 0.0001	0.1656	0.0002

TABLE 7.3 – Résultats de déformation.
Segmentation fait référence au modèle présenté au chapitre 4 et entrainé sur toutes les phases du cycle cardiaque. $r_v(r_i)$:corrélation sur les valeurs de la déformation (corrélation avec la valeur du pic de référence en fin de systole (r_v), corrélation avec les indices de référence (r_i), pentes des droites de régression correspondantes ($slope_v$, $slope_i$) et distance Euclidienne entre les courbes de déformation (dist)). Le symbole ∗ indique l'emploi des cartes de distance.

	Méthode	r_v	$slope_v$	r_i	$slope_i$	dist
Radiale	Non supervisée	0.62	0.35	0.79	0.88	2.19 ± 1.47
	Semi-supervisée ∗	0.85	0.65	0.92	0.98	1.10 ± 0.77
	Supervisée ∗	**0.86**	0.73	**0.94**	0.97	$\mathbf{1.02 \pm 0.73}$
circonférentielle	Non supervisée	0.77	0.71	0.84	0.97	0.33 ± 0.17
	Semi-supervisée ∗	0.90	0.84	**0.95**	1.00	0.18 ± 0.09
	Supervisée ∗	**0.93**	0.86	0.93	0.95	$\mathbf{0.17 \pm 0.08}$
VD	Non supervisée	0.76	0.69	0.82	0.86	0.45 ± 0.31
	Semi-supervisée ∗	0.91	0.98	0.84	0.76	0.28 ± 0.16
	Supervisée ∗	**0.92**	0.95	**0.89**	0.82	$\mathbf{0.27 \pm 0.16}$

7.3.3 Comparaison avec les modèles de référence

Une étude comparative par rapport aux modèles de référence présentés section 6.3.3 et par rapport au modèle du chapitre précédant, avec ou sans l'utilisation des cartes de distance a également été réalisée. Les résultats de suivi de points sont présentés tableaux 7.4 et 7.5 respectivement pour tous les mouvements ou uniquement pour le mouvement entre ED et ES. Le tableau 7.6 présente les résultats d'estimation de la déformation pour ces mêmes modèles .

On remarque que l'utilisation du modèle à mémoire ainsi que de la carte de distance δ_1^k permet d'obtenir de meilleurs résultats de suivi de points et d'estimation de la déformation que le modèle présenté au chapitre précédent et que les autres méthodes de références et ce de façon statistiquement significative. En revanche, pour le modèle semi-supervisé du chapitre précédent, l'utilisation de la carte de distance n'apporte pas d'amélioration importante, avec même une dégradation de l'EPE et du F1-all pour le mouvement allant de ED vers ES. Cela suggère que l'architecture du modèle à mémoire est plus adaptée pour l'utilisation de ces cartes. Cela peut s'expliquer par le plus faible nombre de composantes dans la fonction de coût ce qui rend plus aisé de fixer les poids pour atteindre le bon équilibre. En outre, le modèle du chapitre précédent étant composé de deux réseaux distincts, le second réseau prend en entrée des prédictions obtenus de façon non supervisées. En effet, les cartes de distance et annotations de segmentation ne sont pas utilisées pour superviser le premier réseau qui travaille avec des images adjacentes. Par conséquent, il se peut que le second réseau soit pénalisé par l'absence de supervision du premier réseau.

Nous avons également comparé notre approche avec l'estimation des déformations obtenues en utilisant directement les résultats de la segmentation du chapitre 4 (Tableau 7.6). On peut conclure que la segmentation ne permet pas d'estimer les déformations avec précision.

TABLE 7.4 – Résultats de suivi de points pour toutes les phases.
Comparaison avec les méthodes de base (EPE ± écart-type, F1-all ± écart-type). Les p-values sont calculées entre les modèles "Semi-supervisés de ce chapitre et du chapitre 6. EPI = Epicarde, ENDO = Endocarde. Le symbole ∗ indique l'emploi des cartes de distance.

	Methode	Moyenne	VD	EPI	ENDO
EPE (pixels)	VM-MSE	1.44 ± 0.54	1.57 ± 0.68	1.31 ± 0.51	1.44 ± 0.58
	VM-NCC	1.51 ± 0.60	1.65 ± 0.75	1.30 ± 0.53	1.57 ± 0.71
	VM-Dif	1.42 ± 0.54	1.55 ± 0.67	1.31 ± 0.51	1.40 ± 0.58
	Bioinformed	1.53 ± 0.56	1.68 ± 0.67	1.43 ± 0.54	1.48 ± 0.57
	RAFT	1.37 ± 0.52	1.53 ± 0.66	1.29 ± 0.52	1.29 ± 0.49
	SyN	1.82 ± 0.73	1.88 ± 0.85	1.50 ± 0.61	2.09 ± 0.94
	IterWarpImg	1.76 ± 0.69	1.59 ± 0.67	1.40 ± 0.56	2.29 ± 1.02
	IterWarpFlow	1.79 ± 0.68	1.62 ± 0.66	1.43 ± 0.56	2.32 ± 1.02
	Semi-sup 6	1.15 ± 0.43	1.26 ± 0.53	1.06 ± 0.40	1.13 ± 0.48
	Semi-sup 6 ∗	1.13 ± 0.42	1.24 ± 0.51	1.06 ± 0.41	1.09 ± 0.44
	Semi-sup ∗	$\mathbf{1.09 \pm 0.38}$	$\mathbf{1.19 \pm 0.46}$	$\mathbf{1.02 \pm 0.38}$	$\mathbf{1.07 \pm 0.42}$
	p-value	< 0.0001	< 0.0001	< 0.0001	0.0166
F1-all (%)	VM-MSE	10.92 ± 9.08	12.65 ± 11.08	9.01 ± 8.67	11.11 ± 10.57
	VM-NCC	12.08 ± 0.09	13.65 ± 11.00	9.02 ± 8.90	13.58 ± 11.62
	VM-Dif	10.69 ± 9.11	12.56 ± 11.13	9.12 ± 8.92	10.39 ± 10.36
	Bioinformed	11.60 ± 9.98	14.12 ± 11.86	9.94 ± 9.58	10.73 ± 10.72
	RAFT	8.88 ± 8.55	11.50 ± 11.13	7.91 ± 8.30	7.22 ± 8.40
	SyN	16.66 ± 10.29	16.53 ± 11.41	12.52 ± 10.71	20.94 ± 12.02
	IterWarpImg	16.59 ± 10.70	13.56 ± 11.37	10.74 ± 10.00	25.48 ± 14.31
	IterWarpFlow	16.71 ± 10.73	13.65 ± 11.40	10.87 ± 10.08	25.61 ± 14.32
	Semi-sup 6	5.96 ± 6.55	7.55 ± 8.56	4.73 ± 5.63	5.60 ± 8.06
	Semi-sup 6 ∗	5.68 ± 6.15	7.38 ± 8.30	4.85 ± 5.72	4.82 ± 6.68
	Semi-sup ∗	$\mathbf{5.04 \pm 5.38}$	$\mathbf{6.49 \pm 7.39}$	$\mathbf{4.19 \pm 4.96}$	$\mathbf{4.45 \pm 6.39}$
	p-value	< 0.0001	< 0.0001	0.0004	0.2398

TABLE 7.5 – Résultats de suivi de points uniquement pour la phase ES.
Comparaison avec les méthodes de base (EPE ± écart-type, F1-all ± écart-type). Les p-values sont calculées entre les modèles "Semi-supervisés de ce chapitre et du chapitre 6. EPI = Epicarde, ENDO = Endocarde. Le symbole ∗ indique l'emploi des cartes de distance.

	Methode	Moyenne	VD	EPI	ENDO
EPE (pixels)	VM-MSE	2.30 ± 0.98	2.46 ± 1.33	1.99 ± 0.91	2.45 ± 1.23
	VM-NCC	2.70 ± 1.29	2.87 ± 1.66	2.04 ± 0.99	3.21 ± 1.85
	VM-Dif	2.25 ± 0.96	2.44 ± 1.31	2.00 ± 0.88	2.31 ± 1.23
	Bioinformed	2.17 ± 0.87	2.38 ± 1.11	2.05 ± 0.89	2.08 ± 0.94
	RAFT	1.98 ± 0.84	2.22 ± 1.15	1.83 ± 0.83	1.90 ± 0.89
	SyN	3.74 ± 1.86	3.69 ± 2.21	2.64 ± 1.24	4.89 ± 2.74
	IterWarpImg	3.21 ± 1.57	2.50 ± 1.18	2.30 ± 1.15	4.81 ± 2.77
	IterWarpFlow	3.19 ± 1.34	2.60 ± 1.28	2.24 ± 1.00	4.71 ± 2.27
	Semi-sup 6	1.71 ± 0.74	1.87 ± 1.01	1.56 ± 0.69	1.69 ± 0.81
	Semi-sup 6 $*$	1.77 ± 0.75	1.90 ± 0.94	1.66 ± 0.71	1.75 ± 0.85
	Semi-sup $*$	$\mathbf{1.62 \pm 0.62}$	$\mathbf{1.77 \pm 0.80}$	$\mathbf{1.49 \pm 0.61}$	$\mathbf{1.61 \pm 0.71}$
	p-value	< 0.0001	< 0.0001	< 0.0001	0.0001
F1-all (%)	VM-MSE	24.78 ± 19.20	25.98 ± 20.49	19.56 ± 19.89	28.80 ± 27.08
	VM-NCC	31.18 ± 20.94	29.81 ± 21.29	21.66 ± 21.15	42.08 ± 31.11
	VM-Dif	24.03 ± 19.09	25.87 ± 20.71	19.99 ± 19.96	26.21 ± 26.19
	Bioinformed	22.44 ± 18.23	25.45 ± 19.95	20.15 ± 19.82	21.73 ± 22.49
	RAFT	18.49 ± 16.53	22.02 ± 19.77	15.97 ± 17.23	17.48 ± 20.73
	SyN	45.40 ± 24.17	40.87 ± 25.35	32.45 ± 27.13	62.89 ± 30.55
	IterWarpImg	38.22 ± 23.21	29.42 ± 21.91	25.40 ± 23.83	59.85 ± 32.77
	IterWarpFlow	39.12 ± 20.77	29.74 ± 21.80	23.79 ± 21.15	63.84 ± 29.40
	Semi-sup 6	13.53 ± 14.50	15.49 ± 16.37	11.08 ± 13.65	14.04 ± 19.59
	Semi-sup 6 $*$	14.97 ± 15.25	17.12 ± 17.84	12.86 ± 14.87	14.93 ± 18.80
	Semi-sup $*$	$\mathbf{11.91 \pm 12.31}$	$\mathbf{14.27 \pm 15.43}$	$\mathbf{9.78 \pm 12.06}$	$\mathbf{11.69 \pm 16.13}$
	p-value	< 0.0001	< 0.0001	< 0.0001	0.0090

TABLE 7.6 – Résultats de déformation.
Comparaison avec les méthodes de références (corrélation avec la valeur du pic de référence en fin de systole (r_v), corrélation avec les indices de référence (r_i), pentes des droites de régression correspondantes ($slope_v$, $slope_i$) et distance Euclidienne entre les courbes de déformation (dist)). Le symbole $*$ indique l'emploi des cartes de distance.

	Méthode	r_v	$slope_v$	r_i	$slope_i$	dist
Radiale	Segmentation	0.64	0.63	0.87	1.01	1.52 ± 1.15
	VM-MSE	0.52	0.46	0.79	0.96	1.97 ± 1.42
	VM-NCC	0.53	0.28	0.61	0.94	2.27 ± 1.61
	VM-Dif	0.56	0.48	0.78	0.92	1.88 ± 1.36
	Bioinformed	0.63	0.55	0.87	0.96	1.56 ± 1.37
	RAFT	0.77	0.40	0.88	0.98	1.67 ± 1.31
	SyN	0.43	0.29	0.47	0.92	2.91 ± 1.97
	IterWarpImg	0.49	0.34	0.53	1.04	3.84 ± 1.90
	IterWarpFlow	0.55	0.45	0.60	1.13	3.66 ± 1.72
	Semi-sup 6	0.81	0.54	0.91	0.98	1.31 ± 1.03
	Semi-sup 6 $*$	0.82	0.56	**0.92**	0.99	1.21 ± 1.00
	Semi-sup $*$	**0.85**	0.65	0.92	0.98	$\mathbf{1.10 \pm 0.77}$
circonférentiel	Segmentation	0.78	0.78	0.89	0.96	0.23 ± 0.11
	VM-MSE	0.77	0.88	0.83	1.00	0.28 ± 0.12
	VM-NCC	0.71	0.60	0.75	1.13	0.40 ± 0.19
	VM-Dif	0.78	0.95	0.87	1.04	0.27 ± 0.11
	Bioinformed	0.83	0.78	0.90	0.98	0.23 ± 0.10
	RAFT	0.88	0.80	0.87	0.97	0.19 ± 0.10
	SyN	0.54	0.49	0.63	1.08	0.72 ± 0.30
	IterWarpImg	0.63	0.54	0.79	1.04	0.74 ± 0.24
	IterWarpFlow	0.63	0.54	0.78	1.02	0.69 ± 0.23
	Semi-sup 6	**0.90**	0.88	0.95	0.96	0.21 ± 0.09
	Semi-sup 6 $*$	0.90	0.86	0.92	0.97	0.19 ± 0.09
	Semi-sup $*$	0.90	0.84	**0.95**	1.00	$\mathbf{0.18 \pm 0.09}$
VD	Segmentation	0.34	0.49	0.79	0.78	0.55 ± 0.57
	VM-MSE	0.63	0.63	0.72	0.71	0.49 ± 0.34
	VM-NCC	0.55	0.46	0.73	0.90	0.64 ± 0.42
	VM-Dif	0.67	0.69	0.72	0.69	0.47 ± 0.33
	Bioinformed	0.72	0.75	0.83	0.82	0.44 ± 0.28
	RAFT	0.77	0.70	0.80	0.75	0.39 ± 0.25
	SyN	0.28	0.30	0.62	0.85	0.88 ± 0.60
	IterWarpImg	0.69	0.62	0.82	0.78	0.47 ± 0.33
	IterWarpFlow	0.73	0.68	0.83	0.80	0.48 ± 0.32
	Semi-sup 6	0.89	0.89	**0.85**	0.76	0.30 ± 0.19
	Semi-sup 6 $*$	0.90	0.93	0.84	0.76	0.29 ± 0.17
	Semi-sup $*$	**0.91**	0.98	0.84	0.76	$\mathbf{0.28 \pm 0.16}$

Le tableau 7.7 présente les résultats de déformation régionale du myocarde pour le modèle semi-supervisé présenté dans ce chapitre. On constate que les performances sont moins bonnes que pour la déformation globale, et ce, pour les corrélations, les coefficients directeurs des droites de régressions, ainsi que les distances aux courbes de références. En particulier, on observe que, contrairement à la déformation globale, la déformation régionale est estimée avec une plus grande précision dans la direction radiale que circonférentielle (sauf pour le segment antérieur). Tout comme pour la déformation globale, le modèle estime mieux le moment de la survenu du pic ES que la valeur de déformation en ce pic.

TABLE 7.7 – Résultats de déformation régionale.
Chaque segment correspond à une région du myocarde comme définie par l'AHA (Cerqueira et al. 2002).

	Segment	r_v	$slope_v$	r_i	$slope_i$	dist
Radiale	Antérieur	0.69	0.49	0.85	0.86	1.92 ± 1.37
	Antéro-latéral	0.72	0.45	0.87	0.94	2.11 ± 1.91
	Inféro-latéral	0.66	0.60	0.82	0.95	2.31 ± 1.94
	Inférieur	0.80	0.61	0.78	0.85	1.96 ± 1.59
	Inféro-septal	0.80	0.81	0.79	0.91	1.15 ± 0.92
	Antéro-septal	0.77	0.77	0.81	0.79	1.22 ± 0.84
circonférentielle	Antérieur	0.76	0.68	0.74	0.79	0.51 ± 0.25
	Antéro-latéral	0.61	0.54	0.59	0.63	0.54 ± 0.29
	Inféro-latéral	0.63	0.61	0.79	0.86	0.56 ± 0.27
	Inférieur	0.77	0.64	0.74	0.76	0.51 ± 0.25
	Inféro-septal	0.66	0.53	0.80	0.88	0.55 ± 0.28
	Antéro-septal	0.74	0.62	0.76	0.77	0.46 ± 0.22

7.4 Conclusion

Cette partie a présenté une nouvelle architecture pour estimer le mouvement et suivre les pixels dans des IRM cardiaques. Contrairement à la grand majorité des travaux qui se placent soit dans un contexte non supervisé, soit dans un contexte entièrement supervisé mais en utilisant seulement deux phases de la séquence, nous avons opté pour un cadre semi-supervisé, profitant des segmentations généralement disponible en fin-diastole et fin-systole ainsi que de l'ensemble des images de la séquence. Nous avons également proposé d'utiliser un réseau à mémoire qui intègre les informations tout au long de la séquence plutôt que d'estimer le flux en intégrant les flux d'images successives. L'architecture proposée tire profit des architectures VOS et inclue également le calcul d'un volume de coût, les transformers et un convGRU. Nous avons également introduit dans les fonctions coût des cartes de distance qui permettent de pondérer l'importance des pixels. Les résultats obtenus par cette nouvelle architecture dépassent à la fois ceux des méthodes de référence utilisées pour comparaison et ceux du modèle présenté au chapitre précédant. Ils suggèrent qu'il est possible d'estimer avec précision la déformation cardiaque globale sans avoir besoin d'un jeu de données disposant d'annotations de segmentation pour toutes les images. La déformation régionale est estimée avec une plus faible précision, ce qui ouvre de nouvelles perspectives de recherche.

Quatrième partie

Conclusion et Perspectives

Chapitre 8

Conclusion et perspectives

Pour conclure, nous exposons les principales contributions de cette thèse ainsi que les perspectives pour la suite.

8.1 Conclusion

Nous avons proposé dans un premier temps un réseau de segmentation 2D capable d'extraire des biomarqueurs à partir d'images IRM petit-axe de façon à faciliter le diagnostic des cardiomyopathies ainsi que des infarctus du myocarde. L'architecture du réseau tire parti des mécanismes d'attentions afin de concentrer l'apprentissage sur les zones les plus importantes de l'image. Cela s'est traduit par l'utilisation d'une couche transformer à la plus basse résolution ainsi que par l'introduction des blocs SFB pour les plus hautes résolutions. Ces derniers utilisent la cross-attention entre les cartes de caractéristiques de l'encodeur et du décodeur de manière à réduire l'écart sémantique. Cette cross-attention a été implémentée au sein de zones réduites de l'image par l'intermédiaire de blocs Swin afin de réduire la consommation mémoire. L'utilisation de la couche transformer pour la plus faible résolution a permis d'obtenir un gain de performance statistiquement significatif par rapport à une convolution. Cette architecture a permis d'obtenir des performances de segmentation comparables à l'état de l'art. Nous avons calculé des indices volumétriques et identifié une difficulté de segmentation pour les coupes les plus basales de ventricule droit. La capacité de généralisation de ce réseau a été évaluée en entrainant et testant à la fois sur un jeu de données maison, ainsi que sur le jeu de données ACDC. Nous avons pu montrer que la baisse de performance lorsque le réseau était testé sur un autre jeu de données que le jeu de données d'entrainement était principalement dû à la segmentation des coupes les plus proches de l'apex du cœur, les autres coupes restant bien segmentées. Enfin, nous avons montré que l'entrainement avec toutes les phases du cycle cardiaque, bien qu'ayant peu d'influence sur les résultats de segmentation, conduit à un gain important de performance pour l'estimation de la déformation.

Dans un second temps, nous nous sommes tournés vers l'estimation du flux optique sur des séquences IRM ciné afin de prédire la déformation cardiaque globale et régionale en tenant compte de l'aspect temporel. Ces travaux ont abouti à la mise au point de deux architectures.

La première repose sur l'utilisation de deux réseaux de neurones. Le premier estime le déplacement des pixels entre deux images voisines tandis que le second a pour but d'agréger ces mouvements intermédiaires de façon à prédire le flux optique entre des images distantes de la séquence. L'apprentissage est réalisé de façon semi-supervisée en utilisant uniquement les annotations de segmentation des images de fin-diastole et de fin-systole et en échantillonnant de façon aléatoire les images entre ces deux phases. Cette approche a permis d'estimer la déformation entre la phase de télé-diastolique et toutes les autres phases de la séquence sur un jeu de données maison contenant 271 patients et 912 séquences cardiaques. Nous avons montré que cette approche donne des résultats supérieurs aux méthodes qui estiment le mouvement entre images voisines et composent ensuite les champs de déformation manuellement. Notre méthode donne également de meilleures performances de segmentation, similarité et suivi de points que Voxelmorph et Bioinformed, deux modèles d'estimation du mouvement sur images médicales par apprentissage profond, et RAFT un modèle populaire pour l'estimation du flux optique. De plus, nous avons obtenu des résultats de régularisation similaires à la version de Voxelmorph sans post traitement difféomorphique.

La seconde architecture tire également parti des segmentations de fin-diastole et de fin-systole habituellement disponibles, dans un cadre semi-supervisé. Nous avons proposé l'utilisation d'un réseau à mémoire qui intègre l'information temporelle contenu dans les images passées au lieu d'estimer le flux à partir des flux d'images successives. L'architecture tire parti des architectures VOS proposées pour la segmentation de séquences vidéo mais intègre aussi un volume de coût, les transformers et un convGRU. Cette architecture permet de réduire le nombre de composantes dans la fonction de coût par rapport au modèle de fusion de la partie précédente. Cela a été rendu possible par l'utilisation d'un seul décodeur ainsi que d'un encodeur de mémoire inspiré des architectures des réseaux à mémoire utilisées pour la segmentation vidéo. Nous avons aussi montré l'intérêt d'extraire des cartes de distance à partir des annotations de segmentation de manière à pondérer les fonctions de coûts durant l'entrainement. Plusieurs tests ont été conduits avec des cartes de distance plus ou moins discriminatives. Les meilleurs résultats de déformation et de suivi de points ont été obtenus avec des cartes de distance binaire. Dans ce dernier cas, les résultats obtenus sont meilleurs que ceux du réseau de fusion de la partie précédente. Cette nouvelle architecture obtient des résultats d'estimation de la déformation globale nettement meilleurs qu'une approche par segmentation 2D. Cependant, les déformations estimées pour chaque segment du myocarde sont moins précises que la déformation globale du muscle cardiaque.

8.2 Perspectives

Durant cette thèse, l'apprentissage multi-tâche du flux optique et de la segmentation a été testé à la fois pour le modèle de fusion de la partie 6 et le réseau à mémoire de la partie 7 sans permettre d'améliorer les performances d'estimation du mouvement. Certains travaux ont pourtant montré l'intérêt d'une telle approche. Il se peut que l'absence de gain de performance dans l'estimation du flux optique soit liée à une pondération sous-optimale des composantes de la fonction de coût ou

à de mauvais choix architecturaux. Par conséquent, les architectures présentées et les poids des différentes composantes de la fonction de coût pourraient être modifiés pour prédire à la fois la segmentation des images de la séquence et le mouvement des pixels entre ces images, de manière à obtenir une approche unifiée. Cela permettrait également au calcul du flux optique d'être entièrement supervisé par la segmentation.

Il serait également intéressant d'utiliser une architecture qui exploite l'information temporelle des images suivantes de la séquence et non uniquement des images passées. Pour cela, un convGRU (ou convLSTM) bidirectionnel pourrait être utilisé. Un seul encodeur serait alors nécessaire, ce qui permet de se passer de l'encodeur de mémoire du chapitre 7. L'inconvénient résiderait alors dans l'impossibilité d'utiliser le flux de mouvement accumulé, ce qui pourrait poser problème pour le processus d'agrégation des mouvements (équation 7.3). Toujours dans cette optique d'utiliser les mouvements futurs et dans un contexte multi-tâches il pourrait être pertinent, à chaque instant, de faire prédire au réseau le mouvement vers l'image suivante. Cela inciterait le réseau à anticiper les mouvements futurs, ce qui est un principe fondamental en neuroscience (c.f littérature sur le "predictive coding" souvent adapté à l'apprentissage profond (Lotter, Kreiman et Cox 2017 ; Han, Wen et al. 2018 ; Oord, Li et Vinyals 2019)).

Il pourrait également être envisagé de passer les points de contour directement au réseau sous la forme de coordonnées 2D. Le réseau reposerait alors sur deux encodeurs séparés, l'un pour les points de contours, l'autre pour les images de la séquence. Une fusion entre les caractéristiques générées par les deux encodeurs pourrait alors être effectuée lorsque la résolution de l'image est la plus faible et que le nombre de caractéristiques est le plus élevé. Cette approche est particulièrement adaptée à l'usage des transformers puisque le traitement de données 1D comme les points de contours nécessite moins de ressources mémoire que le traitement des images. De plus, il a été montré que la cross-attention fonctionne bien sur des données multimodales comme les images et le texte. Dans notre cas, la cross-attention pourrait être utilisée pour fusionner l'information provenant des images et des points de contours.

Ces travaux se sont concentrés sur l'usage des convolutions et couches transformers. Or, ces deux méthodes de traitement des images ont des limitations importantes. La première dispose d'un champ réceptif limité et ne permet pas de prêter attention à d'autres caractéristiques (pas de cross-attention). La seconde consomme beaucoup de ressources mémoire (ce qui les rend difficilement adaptable au traitement vidéo) et l'apprentissage est lent. Par conséquent, il serait intéressant d'utiliser les dernières avancés sur les "State Space Models" (SSM). Ces derniers ont une formulation similaire au filtre de Kalman et ont montré des résultats supérieurs aux transformers pour le texte et la vision tout en nécessitant moins de ressources mémoire et de paramètres. Le traitement récursif des données par les SSM les rend particulièrement adapté, entre autres, au traitement vidéo. Par conséquent, les modèles reposant sur les SSM (MAMBA (Gu et Dao 2024), Vision Mamba (Zhu, Liao et al. 2024), convSSM (Smith, De Mello et al. 2023), S4 (Gu, Goel et Ré 2022), etc...) pourraient permettre d'obtenir de meilleures performances sur les séquences IRM dynamiques.

Cinquième partie

Annexe

.1 Chapitre 1

.1.1 Augmentation des données

Les paramètres d'augmentation sont les valeurs par défaut de nnU-net (Isensee et al. 2018). Les transformations sont appliquées à l'aide du framework batchgenerator, qui est utilisé par défaut par nnU-net. Les paramètres d'augmentation sont les mêmes pour le jeu de données ACDC et le jeu de données maison.

Les augmentations sont effectuées à la volée pendant l'entraînement. L'image x est transformée pour obtenir x' avec une probabilité α :

— Rotations aléatoires entre $-180°$ et $180°$, $\alpha = 0.2$
— Correction gamma avec préservation de la moyenne et de l'écart-type :

$$x' = \left(\frac{x - \min(x)}{\max(x) - \min(x)}\right)^{\gamma} \times (\max(x) - \min(x)) + \min(x)$$

où γ est échantillonné aléatoirement dans $[0.7; 1.5]$, $\alpha = 0.3$
— zoom : facteur de zoom échantillonné dans $[0.7; 1.4]$, $\alpha = 0.2$
— Inversion (flipping) le long de l'axe vertical ou horizontal, chacun avec $\alpha = 0.5$
— Augmentation du contraste :

$$x' = (x - \min(x)) \times \beta + \min(x)$$

où β est échantillonné aléatoirement dans $[0.75; 1.25]$, $\alpha = 0.15$
— Simulation de basse résolution : sous-échantillonne chaque image (linéairement) par un facteur aléatoire et sur-échantillonne à la résolution d'origine. Sous-échantillonnage avec interpolation par plus proche voisin, sur-échantillonnage avec interpolation par spline de 3ème ordre. L'image est sous-échantillonnée par un facteur β échantillonné aléatoirement dans $[0.5; 1.0]$. Aucun anti-aliasing n'est utilisé. $\alpha = 0.25$
— Bruit gaussien : ajoute du bruit gaussien additif. La variance est échantillonnée dans $[0.0; 0.1]$, $\alpha = 0.1$
— Flou gaussien : échantillonne uniformément $\sigma \in [1; 5]$ pour le noyau gaussien, $\alpha = 0.1$
— Ajustement de la luminosité :

$$x' = x \times \beta$$

avec $\beta \in [0.75; 1.25]$, $\alpha = 0.15$

.1.2 Architecture du réseau de segmentation

TABLE 1 – Architecture du réseau.
La fonction d'activation Gelu et la batch-normalisation sont utilisées après chaque convolution. Une couche convolutionnelle est composée de 2 et 1 paires de convolutions pour l'encodeur et le décodeur respectivement. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de batch-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 2$	(1, H, W)	(64, H, W)	65,376
Strided_conv	[3×3, 2]×1	(64, H, W)	(128, H/2, W/2)	74,112
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 2$	(128, H/2, W/2)	(128, H/2, W/2)	591,360
Strided_conv	[3×3, 2]×1	(128, H/2, W/2)	(256, H/4, W/4)	295,680
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 2$	(256, H/4, W/4)	(256, H/4, W/4)	2,362,368
Strided_conv	[3×3, 2]×1	(256, H/4, W/4)	(512, H/8, W/8)	1,181,184
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(512, H/8, W/8)	(512, H/8, W/8)	4,721,664
Transformer Layer	×1	(512, H/8, W/8)	(512, H/8, W/8)	3,152,384
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(512, H/8, W/8)	(512, H/8, W/8)	4,721,664
Transposed_conv	[2×2, 2]×1	(512, H/8, W/8)	(256, H/4, W/4)	525,056
Swin Filtering Block	×1	(256, H/4, W/4)	(256, H/4, W/4)	1,126,672
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(512, H/4, W/4)	(256, H/4, W/4)	1,771,008
Transposed_conv	[2×2, 2]×1	(256, H/4, W/4)	(128, H/2, W/2)	131,456
Swin Filtering Block	×1	(128, H/2, W/2)	(128, H/2, W/2)	284,808
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(256, H/2, W/2)	(128, H/2, W/2)	443,136
Transposed_conv	[2×2, 2]×1	(128, H/2, W/2)	(64, H, W)	32,960
Swin Filtering Block	×1	(64, H, W)	(64, H, W)	72,772
Conv_layer	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(128, H, W)	(4, H, W)	4,776

.2 Chapitre 2

.2.1 Architecture du réseau d'agrégation de flux

TABLE 2 – Architecture du réseau.
La fonction d'activation Gelu et la group-normalisation sont utilisées après chaque convolution. Une couche convolutionnelle est composée de 2 convolutions. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de group-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(1, H, W)	(64, H, W)	37,824
Strided_conv	[3×3, 2]×1	(64, H, W)	(128, H/2, W/2)	74,112
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(128, H/2, W/2)	(128, H/2, W/2)	295,680
Strided_conv	[3×3, 2]×1	(128, H/2, W/2)	(256, H/4, W/4)	295,680
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(256, H/4, W/4)	(256, H/4, W/4)	1,181,184
Strided_conv	[3×3, 2]×1	(256, H/4, W/4)	(512, H/8, W/8)	1,181,184
Transformer decoder layer	×1	(512, H/8, W/8)	(512, H/8, W/8)	4,204,032
Transposed_conv	[2×2, 2]×1	(512, H/8, W/8)	(256, H/4, W/4)	525,056
Skip_connection_merging	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(512, H/4, W/4)	(256, H/4, W/4)	1,771,008
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(512, H/4, W/4)	(256, H/4, W/4)	1,771,008
Transposed_conv	[2×2, 2]×1	(256, H/4, W/4)	(128, H/2, W/2)	131,456
Skip_connection_merging	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(256, H/2, W/2)	(128, H/2, W/2)	443,136
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(256, H/2, W/2)	(128, H/2, W/2)	443,136
Transposed_conv	[2×2, 2]×1	(128, H/2, W/2)	(64, H, W)	32,960
Skip_connection_merging	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(128, H, W)	(64, H, W)	110,976
Conv_layer	$\begin{bmatrix} 3\times3, 1 \\ 3\times3, 1 \end{bmatrix}\times 1$	(128, H, W)	(64, H, W)	110,976
Final conv	[3×3, 1]×1	(64, H, W)	(2, H, W)	1,154

.3 Chapitre 3

.3.1 Architecture du réseau à mémoire

TABLE 3 – Architecture de l'encodeur du réseau.
La fonction d'activation Gelu et la group-normalisation sont utilisées après chaque convolution d'un Resblock. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de group-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
ResBlock_memory_1	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times 1$	(6, H, W)	(64, H, W)	41,280
ResBlock_memory_2	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(64, H, W)	(128, H/2, W/2)	230,528
ResBlock_memory_3	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(128, H/2, W/2)	(128, H/2, W/2)	295,680
ResBlock_memory_4	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(128, H/2, W/2)	(256, H/4, W/4)	919,808
ResBlock_memory_5	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(256, H/4, W/4)	(256, H/4, W/4)	1,181,184
ResBlock_memory_6	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(256, H/4, W/4)	(256, H/8, W/8)	1,247,488
ResBlock_memory_7	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(256, H/8, W/8)	(256, H/8, W/8)	1,181,184
ResBlock_query_1	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times 1$	(1, H, W)	(64, H, W)	38,080
ResBlock_query_2	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(64, H, W)	(128, H/2, W/2)	230,528
ResBlock_query_3	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(128, H/2, W/2)	(128, H/2, W/2)	295,680
ResBlock_query_4	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(128, H/2, W/2)	(256, H/4, W/4)	919,808
ResBlock_query_5	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(256, H/4, W/4)	(256, H/4, W/4)	1,181,184
ResBlock_query_6	$\begin{bmatrix} 3\times3,2 \\ 3\times3,1 \\ 1\times1,2 \end{bmatrix}\times 1$	(256, H/4, W/4)	(256, H/8, W/8)	1,247,488
ResBlock_query_7	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(256, H/8, W/8)	(256, H/8, W/8)	1,181,184

TABLE 4 – Architecture des skip connections du réseau.

La fonction d'activation Gelu et la group-normalisation sont utilisées après chaque convolution d'un Resblock. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de group-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
Cost_volume_upsample_1	$\begin{bmatrix} 2\times2,2 \end{bmatrix}\times1$	(81, H/4, W/4)	(81, H, W)	105,057
Cost_volume_expand_1	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(81, H, W)	(64, H, W)	89,280
Skip_co_reduction_1	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(128, H, W)	(64, H, W)	119,360
Cost_volume_upsample_2	$\begin{bmatrix} 2\times2,2 \end{bmatrix}\times1$	(81, H/4, W/4)	(81, H/2, W/2)	26,325
Cost_volume_expand_2	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(81, H/2, W/2)	(128, H/2, W/2)	252,288
Skip_co_reduction_2	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(256, H/2, W/2)	(128, H/2, W/2)	476,288
Cost_volume_expand_3	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(81, H/4, W/4)	(256, H/4, W/4)	799,488
Skip_co_reduction_3	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(512, H/4, W/4)	(256, H/4, W/4)	1,902,848

TABLE 5 – Architecture du décodeur du réseau.

La fonction d'activation Gelu et la group-normalisation sont utilisées après chaque convolution d'un Resblock. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de group-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
Transposed_conv_1	$\begin{bmatrix} 2\times2,2 \end{bmatrix}\times1$	(256, H/8, W/8)	(256, H/4, W/4)	262,912
ResBlock_decoder_1	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(512, H/4, W/4)	(256, H/4, W/4)	1,902,848
Transposed_conv_2	$\begin{bmatrix} 2\times2,2 \end{bmatrix}\times1$	(256, H/4, W/4)	(128, H/2, W/2)	131,840
ResBlock_decoder_2	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(256, H/2, W/2)	(128, H/2, W/2)	476,288
Transposed_conv_3	$\begin{bmatrix} 2\times2,2 \end{bmatrix}\times1$	(128, H/2, W/2)	(64, H, W)	32,960
ResBlock_decoder_3	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times1$	(256, H, W)	(128, H, W)	119,360
Final_conv	$\begin{bmatrix} 3\times3,1 \end{bmatrix}\times1$	(64, H, W)	(2, H, W)	1,154

TABLE 6 – Architecture du bottleneck du réseau.
La fonction d'activation Gelu et la group-normalisation sont utilisées après chaque convolution d'un Resblock. Pour une convolution, les paramètres sont donnés sous la forme [taille du noyau, stride]. Le nombre de paramètres apprenables inclut les paramètres de group-normalisation.

Nom de la couche	**[taille du noyau, stride]**	**Taille de la carte de caractéristiques d'entrée (#caractéristiques, hauteur, largeur)**	**Taille de la carte de caractéristiques de sortie (#caractéristiques, hauteur, largeur)**	**# paramètres apprenables**
GRU_cell	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \end{bmatrix}\times 1$	(512, H/8, W/8)	(256, H/8, W/8)	3,539,712
Transformer_memory	$\times 1$	(256, H/8, W/8)	(256, H/8, W/8)	1,578,752
Transformer_query	$\times 1$	(256, H/8, W/8)	(256, H/8, W/8)	1,578,752
ResBlock_reduction	$\begin{bmatrix} 3\times3,1 \\ 3\times3,1 \\ 1\times1,1 \end{bmatrix}\times 1$	(512, H/8, W/8)	(256, H/8, W/8)	1,902,848

Bibliographie

A, Shamla Beevi et al. (2023). *Swin-EchoNet : Deep Learning-based Two-Chamber Segmentation of 2D Echocardiography using Swin Transformer*. ISSN : 2693-5015.

Aletras, Anthony H. et al. (1999). "DENSE : Displacement Encoding with Stimulated Echoes in Cardiac Functional MRI". In : *Journal of magnetic resonance (San Diego, Calif. : 1997)* 137.1, p. 247-252. ISSN : 1090-7807.

Alom, Md Zahangir et al. (2018). *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*. arXiv :1802.06955 [cs].

Alvarez-Florez, Laura et al. (2023). *Deep Learning for Automatic Strain Quantification in Arrhythmogenic Right Ventricular Cardiomyopathy*. arXiv :2311.14448 [cs, eess].

Alzubaidi, Laith et al. (2021). *MedNet : Pre-trained Convolutional Neural Network Model for the Medical Imaging Tasks*. arXiv :2110.06512 [cs].

Amano, Yasuo et al. (2018). "Cardiac MR Imaging of Hypertrophic Cardiomyopathy : Techniques, Findings, and Clinical Relevance". In : *Magnetic Resonance in Medical Sciences* 17.2, p. 120-131. ISSN : 1347-3182.

Amin, Elena K., Adrienne Campbell-Washburn et Kanishka Ratnayaka (2022). "MRI-Guided Cardiac Catheterization in Congenital Heart Disease : How to Get Started". In : *Current Cardiology Reports* 24.4, p. 419-429. ISSN : 1523-3782.

Arega, Tewodros Weldebirhan et Stéphanie Bricq (2020). "Automatic Myocardial Scar Segmentation from Multi-sequence Cardiac MRI Using Fully Convolutional Densenet with Inception and Squeeze-Excitation Module". en. In : *Myocardial Pathology Segmentation Combining Multi-Sequence Cardiac Magnetic Resonance Images*. Sous la dir. de Xiahai Zhuang et Lei Li. Cham : Springer International Publishing, p. 102-117. ISBN : 978-3-030-65651-5.

Arnekvist, Isac et al. (2020). *The effect of Target Normalization and Momentum on Dying ReLU*. arXiv :2005.06195 [cs, stat].

Arsigny, Vincent et al. (2006). "A log-Euclidean framework for statistics on diffeomorphisms". eng. In : *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention* 9.Pt 1, p. 924-931.

Atito, Sara, Muhammad Awais et Josef Kittler (2022). *SiT : Self-supervised vIsion Transformer*. arXiv :2104.03602 [cs].

Augustine, Daniel et al. (2013). "Global and regional left ventricular myocardial deformation measures by magnetic resonance feature tracking in healthy volunteers : comparison with tagging and relevance of gender". In : *Journal of Cardiovascular Magnetic Resonance* 15.1, p. 8. ISSN : 1532-429X.

Authors/Task Force Members et al. (2008). "Management of acute myocardial infarction in patients presenting with persistent ST-segment elevation : The Task

Force on the management of ST-segment elevation acute myocardial infarction of the European Society of Cardiology :" in : *European Heart Journal* 29.23, p. 2909-2945. ISSN : 0195-668X.

Avants, B. B. et al. (2008). "Symmetric diffeomorphic image registration with cross-correlation : Evaluating automated labeling of elderly and neurodegenerative brain". In : *Medical Image Analysis*. Special Issue on The Third International Workshop on Biomedical Image Registration – WBIR 2006 12.1, p. 26-41. ISSN : 1361-8415.

Azad, Reza et al. (2022). *DAE-Former : Dual Attention-guided Efficient Transformer for Medical Image Segmentation*. arXiv :2212.13504 [cs] version : 1.

Ba, Jimmy Lei, Jamie Ryan Kiros et Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv :1607.06450 [cs, stat].

Bai, Wenjia et al. (2018). "Automated cardiovascular magnetic resonance image analysis with fully convolutional networks". In : *Journal of Cardiovascular Magnetic Resonance* 20.1, p. 65. ISSN : 1532-429X.

Balakrishnan, Guha et al. (2019). "VoxelMorph : A Learning Framework for Deformable Medical Image Registration". In : *IEEE Transactions on Medical Imaging* 38.8. arXiv :1809.05231 [cs], p. 1788-1800. ISSN : 0278-0062, 1558-254X.

Balas, Valentina E., Raghvendra Kumar et Rajshree Srivastava (2019). *Recent Trends and Advances in Artificial Intelligence and Internet of Things*. en. Google-Books-ID : XRS_DwAAQBAJ. Springer Nature. ISBN : 978-3-030-32644-9.

Ballas, Nicolas et al. (2016). *Delving Deeper into Convolutional Networks for Learning Video Representations*. arXiv :1511.06432 [cs].

Basit, Hajira, Daniel Brito et Saurabh Sharma (2024). "Hypertrophic Cardiomyopathy". eng. In : *StatPearls*. Treasure Island (FL) : StatPearls Publishing.

Bello, Irwan et al. (2020). *Attention Augmented Convolutional Networks*. arXiv :1904.09925 [cs].

Bernard, Olivier et al. (2018). "Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis : Is the Problem Solved ?" en. In : *IEEE Transactions on Medical Imaging* 37.11, p. 2514-2525. ISSN : 0278-0062, 1558-254X.

Bianconi, Francesco et al. (2021). "Comparative evaluation of conventional and deep learning methods for semi-automated segmentation of pulmonary nodules on CT". In : *Quantitative Imaging in Medicine and Surgery* 11.7, p. 3286-3305. ISSN : 2223-4292.

Blackshear, Joseph L. et John A. Odell (1996). "Appendage obliteration to reduce stroke in cardiac surgical patients with atrial fibrillation". In : *The Annals of Thoracic Surgery* 61.2, p. 755-759. ISSN : 0003-4975.

Blendowski, Max, Lasse Hansen et Mattias P. Heinrich (2021). "Weakly-supervised learning of multi-modal features for regularised iterative descent in 3D image registration". In : *Medical Image Analysis* 67, p. 101822. ISSN : 1361-8415.

Brahim, K. et al. (2021). "A 3D Network Based Shape Prior for Automatic Myocardial Disease Segmentation in Delayed-Enhancement MRI". In : *IRBM* 42.6, p. 424-434. ISSN : 1959-0318.

Buss, Sebastian J. et al. (2015). "Prediction of functional recovery by cardiac magnetic resonance feature tracking imaging in first time ST-elevation myocardial infarction. Comparison to infarct size and transmurality by late gadolinium en-

hancement". eng. In : *International Journal of Cardiology* 183, p. 162-170. ISSN : 1874-1754.

Butler, Daniel J. et al. (2012). "A Naturalistic Open Source Movie for Optical Flow Evaluation". en. In : *Computer Vision – ECCV 2012*. Sous la dir. d'Andrew Fitzgibbon et al. Berlin, Heidelberg : Springer, p. 611-625. ISBN : 978-3-642-33783-3.

Bycroft, Clare et al. (2018). "The UK Biobank resource with deep phenotyping and genomic data". en. In : *Nature* 562.7726. Publisher : Nature Publishing Group, p. 203-209. ISSN : 1476-4687.

Caicedo, Juan C. et al. (2019). "Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images". en. In : *Cytometry Part A* 95.9. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.23863, p. 952-965. ISSN : 1552-4930.

Caillol, H., W. Pieczynski et A. Hillion (1997). "Estimation of fuzzy Gaussian mixture and unsupervised statistical image segmentation". In : *IEEE Transactions on Image Processing* 6.3. Conference Name : IEEE Transactions on Image Processing, p. 425-440. ISSN : 1941-0042.

Campello, Víctor M. et al. (2021). "Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Segmentation : The M&Ms Challenge". In : *IEEE Transactions on Medical Imaging* 40.12. Conference Name : IEEE Transactions on Medical Imaging, p. 3543-3554. ISSN : 1558-254X.

Cao, Hu, Yueyue Wang et al. (2021). *Swin-Unet : Unet-like Pure Transformer for Medical Image Segmentation*. arXiv :2105.05537 [cs, eess].

Cao, Zhen, Chuanfeng Ma et al. (2022). "Relay-UNet : Reduce Semantic Gap for Glomerular Image Segmentation". en. In : *Intelligence Science IV*. Sous la dir. de Zhongzhi Shi, Yaochu Jin et Xiangrong Zhang. Cham : Springer International Publishing, p. 378-385. ISBN : 978-3-031-14903-0.

Carion, Nicolas et al. (2020). *End-to-End Object Detection with Transformers*. arXiv :2005.12872 [cs].

Carr, H. Y. (1958). "Steady-State Free Precession in Nuclear Magnetic Resonance". In : *Physical Review* 112.5. Publisher : American Physical Society, p. 1693-1701.

Caruana, Rich, Steve Lawrence et C. Giles (2000). "Overfitting in Neural Nets : Backpropagation, Conjugate Gradient, and Early Stopping". In : *Advances in Neural Information Processing Systems*. T. 13. MIT Press.

Cawley, Peter J., Jeffrey H. Maki et Catherine M. Otto (2009). "Cardiovascular Magnetic Resonance Imaging for Valvular Heart Disease". In : *Circulation* 119.3. Publisher : American Heart Association, p. 468-478.

Cerqueira, Manuel D. et al. (2002). "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart. A statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association". eng. In : *Circulation* 105.4, p. 539-542. ISSN : 1524-4539.

Chan, Tony et Luminita Vese (1999). "An Active Contour Model without Edges". en. In : *Scale-Space Theories in Computer Vision*. Sous la dir. de Mads Nielsen et al. Berlin, Heidelberg : Springer, p. 141-151. ISBN : 978-3-540-48236-9.

Chang, Qi et al. (2022). *DeepRecon : Joint 2D Cardiac Segmentation and 3D Volume Reconstruction via A Structure-Specific Generative Method*. en. arXiv :2206.07163 [cs, eess].

Chartsias, Agisilaos et al. (2020). “Multimodal Cardiac Segmentation Using Disentangled Representation Learning”. en. In : *Statistical Atlases and Computational Models of the Heart. Multi-Sequence CMR Segmentation, CRT-EPiggy and LV Full Quantification Challenges*. Sous la dir. de Mihaela Pop et al. T. 12009. Series Title : Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 128-137. ISBN : 978-3-030-39073-0 978-3-030-39074-7.

Chaurasia, Abhishek et Eugenio Culurciello (2017). “LinkNet : Exploiting Encoder Representations for Efficient Semantic Segmentation”. In : *2017 IEEE Visual Communications and Image Processing (VCIP)*. arXiv :1707.03718 [cs], p. 1-4.

Chen, Chen, Wenjia Bai et Daniel Rueckert (2019). “Multi-Task Learning for Left Atrial Segmentation on GE-MRI”. en. In : t. 11395. arXiv :1810.13205 [cs], p. 292-301.

Chen, Chen, Carlo Biffi et al. (2019). “Learning Shape Priors for Robust Cardiac MR Segmentation from Multi-view Images”. en. In : t. 11765. arXiv :1907.09983 [cs, eess], p. 523-531.

Chen, Jieneng, Yongyi Lu et al. (2021). *TransUNet : Transformers Make Strong Encoders for Medical Image Segmentation*. arXiv :2102.04306 [cs].

Chen, Liang-Chieh, George Papandreou et al. (2016). *Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs*. arXiv :1412.7062 [cs].

— (2018). “DeepLab : Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 834-848. ISSN : 1939-3539.

Chen, Liang-Chieh, Yukun Zhu et al. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. arXiv :1802.02611 [cs].

Chen, Qiuhui, Haiying Lyu et al. (2023). *Volumetric Medical Image Segmentation via Scribble Annotations and Shape Priors*. en. arXiv :2310.08084 [cs].

Chen, Sihong, Kai Ma et Yefeng Zheng (2019). *Med3D : Transfer Learning for 3D Medical Image Analysis*. arXiv :1904.00625 [cs].

Chen, Siqi et Richard J. Radke (2009). “Level set segmentation with both shape and intensity priors”. In : *2009 IEEE 12th International Conference on Computer Vision*. ISSN : 2380-7504, p. 763-770.

Chen, Xiongchao, Bo Zhou et al. (2022). “CT-free attenuation correction for dedicated cardiac SPECT using a 3D dual squeeze-and-excitation residual dense network”. In : *Journal of Nuclear Cardiology* 29.5, p. 2235-2250. ISSN : 1071-3581.

Chen, Yong-Sheng, Yi-Ping Hung et Chiou-Shann Fuh (2001). “Fast block matching algorithm based on the winner-update strategy”. In : *IEEE Transactions on Image Processing* 10.8. Conference Name : IEEE Transactions on Image Processing, p. 1212-1222. ISSN : 1941-0042.

Cheng, Ho Kei et Alexander G. Schwing (2022). “XMem : Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model”. en. In : *Computer Vision – ECCV 2022*. Sous la dir. de Shai Avidan et al. Lecture Notes in Computer Science. Cham : Springer Nature Switzerland, p. 640-658. ISBN : 978-3-031-19815-1.

Cheng, Ho Kei, Yu-Wing Tai et Chi-Keung Tang (2021a). “Modular Interactive Video Object Segmentation : Interaction-to-Mask, Propagation and Difference-Aware Fusion”. en. In : p. 5559-5568.

Cheng, Ho Kei, Yu-Wing Tai et Chi-Keung Tang (2021b). “Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation”. In : *Advances in Neural Information Processing Systems*. T. 34. Curran Associates, Inc., p. 11781-11794.

Cheng, Yizong (1995). “Mean shift, mode seeking, and clustering”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 790-799. ISSN : 1939-3539.

Cherti, Mehdi et Jenia Jitsev (2022). “Effect of pre-training scale on intra- and inter-domain, full and few-shot transfer learning for natural and X-Ray chest images”. In : *2022 International Joint Conference on Neural Networks (IJCNN)*. ISSN : 2161-4407, p. 1-9.

Chu, Xiangxiang et al. (2023). *Conditional Positional Encodings for Vision Transformers*. arXiv :2102.10882 [cs].

Chung, Ginmo et Luminita A. Vese (2005). “Energy Minimization Based Segmentation and Denoising Using a Multilayer Level Set Approach”. en. In : *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Sous la dir. d’Anand Rangarajan, Baba Vemuri et Alan L. Yuille. Berlin, Heidelberg : Springer, p. 439-455. ISBN : 978-3-540-32098-2.

Chung, Mina K., Lee L. Eckhardt et al. (2020). “Lifestyle and Risk Factor Modification for Reduction of Atrial Fibrillation : A Scientific Statement From the American Heart Association”. en. In : *Circulation* 141.16. ISSN : 0009-7322, 1524-4539.

Çiçek, Özgün et al. (2016). *3D U-Net : Learning Dense Volumetric Segmentation from Sparse Annotation*. arXiv :1606.06650 [cs].

Cirino, Allison L. et Carolyn Ho (1993). “Hypertrophic Cardiomyopathy Overview”. eng. In : *GeneReviews®*. Sous la dir. de Margaret P. Adam et al. Seattle (WA) : University of Washington, Seattle.

Clevert, Djork-Arné, Thomas Unterthiner et Sepp Hochreiter (2016). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. arXiv :1511.07289 [cs].

Dabov, Kostadin et al. (2006). “Image denoising with block-matching and 3D filtering”. In : *Image Processing : Algorithms and Systems, Neural Networks, and Machine Learning*. T. 6064. SPIE, p. 354-365.

Dai, Zihang, Hanxiao Liu et al. (2021). “CoAtNet : Marrying Convolution and Attention for All Data Sizes”. In : *arXiv :2106.04803 [cs]*. arXiv : 2106.04803 version : 2.

Dai, Zihang, Zhilin Yang et al. (2019). *Transformer-XL : Attentive Language Models Beyond a Fixed-Length Context*. arXiv :1901.02860 [cs, stat].

Dalca, Adrian V. et al. (2018). “Unsupervised Learning for Fast Probabilistic Diffeomorphic Registration”. en. In : t. 11070. arXiv :1805.04605 [cs], p. 729-738.

— (2019). “Unsupervised Learning of Probabilistic Diffeomorphic Registration for Images and Surfaces”. In : *Medical Image Analysis* 57. arXiv :1903.03545 [cs], p. 226-236. ISSN : 13618415.

Damiano, Ralph J. et al. (2011). “The Cox maze IV procedure : Predictors of late recurrence”. In : *The Journal of Thoracic and Cardiovascular Surgery* 141.1, p. 113-121. ISSN : 0022-5223.

Dangi, Shusil, Ziv Yaniv et Cristian A. Linte (2019). "Left Ventricle Segmentation and Quantification from Cardiac Cine MR Images via Multi-task Learning". In : *Statistical atlases and computational models of the heart. STACOM (Workshop)* 11395, p. 21-31.

Davidson's principles and practice of medicine. - NLM Catalog - NCBI (2024).

Dent, Susan F. (2019). *Practical Cardio-Oncology*. en. Google-Books-ID : kSysDwAAQBAJ. CRC Press. ISBN : 978-1-351-58356-5.

Ding, Mingyu et al. (2020). "Every Frame Counts : Joint Learning of Video Segmentation and Optical Flow". en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07. Number : 07, p. 10713-10720. ISSN : 2374-3468.

Dong, Qiaole et Yanwei Fu (2024). *MemFlow : Optical Flow Estimation and Prediction with Memory*. arXiv :2404.04808 [cs].

Dong, Suyu, Gongning Luo et al. (2018). "A Combined Fully Convolutional Networks and Deformable Model for Automatic Left Ventricle Segmentation Based on 3D Echocardiography". In : *BioMed Research International* 2018, p. 5682365. ISSN : 2314-6133.

Dosovitskiy, Alexey et al. (2021). *An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale*. arXiv :2010.11929 [cs].

Duke, Brendan et al. (2021). "SSTVOS : Sparse Spatiotemporal Transformers for Video Object Segmentation". en. In : *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA : IEEE, p. 5908-5917. ISBN : 978-1-66544-509-2.

Elliott, Perry et al. (2008). "Classification of the cardiomyopathies : a position statement from the european society of cardiology working group on myocardial and pericardial diseases". In : *European Heart Journal* 29.2, p. 270-276. ISSN : 0195-668X.

Erturk, S. (2003). "Digital image stabilization with sub-image phase correlation based global motion estimation". In : *IEEE Transactions on Consumer Electronics* 49.4. Conference Name : IEEE Transactions on Consumer Electronics, p. 1320-1325. ISSN : 1558-4127.

Fan, Chunyu et al. (2023). "ViT-FRD : A Vision Transformer Model for Cardiac MRI Image Segmentation Based on Feature Recombination Distillation". en. In : *IEEE Access* 11, p. 129763-129772. ISSN : 2169-3536.

Ferdian, Edward et al. (2020). "Fully Automated Myocardial Strain Estimation from Cardiovascular MRI–tagged Images Using a Deep Learning Framework in the UK Biobank". en. In : *Radiology : Cardiothoracic Imaging* 2.1, e190032. ISSN : 2638-6135.

Ferede, Fisseha Admasu et Madhusudhanan Balasubramanian (2023). "SSTM : Spatiotemporal recurrent transformers for multi-frame optical flow estimation". In : *Neurocomputing* 558, p. 126705. ISSN : 0925-2312.

Fischer, Philipp et al. (2015). *FlowNet : Learning Optical Flow with Convolutional Networks*. en. arXiv :1504.06852 [cs].

Fitzgerald, Benjamin T., John Bashford et Gregory M. Scalia (2017). "Regression of the Anatomic Cardiac Features of Amyloid Light Chain Cardiac Amyloidosis Accompanied by Normalization of Global Longitudinal Strain". In : *CASE* 1.2, p. 46-48. ISSN : 2468-6441.

Ford, L. R. et D. R. Fulkerson (1957). “A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem”. en. In : *Canadian Journal of Mathematics* 9, p. 210-218. ISSN : 0008-414X, 1496-4279.

Fornasier-Santos, Charly (2018). “Entraînement, Préparation physique & Physiologie cardiovasculaire appliqués au rugby à XV”. Thèse de doct.

Foroosh, H., J.B. Zerubia et M. Berthod (2002). “Extension of phase correlation to subpixel registration”. In : *IEEE Transactions on Image Processing* 11.3. Conference Name : IEEE Transactions on Image Processing, p. 188-200. ISSN : 1941-0042.

Fu, Zhenyin et al. (2022). “TF-Unet :An automatic cardiac MRI image segmentation method”. en. In : *Mathematical Biosciences and Engineering* 19.5, p. 5207-5222. ISSN : 1551-0018.

Fukushima, Kunihiko (1975). “Cognitron : A self-organizing multilayered neural network”. en. In : *Biological Cybernetics* 20.3, p. 121-136. ISSN : 1432-0770.

Galazis, Christoforos et al. (2022). “Tempera : Spatial Transformer Feature Pyramid Network for Cardiac MRI Segmentation”. en. In : t. 13131. arXiv :2203.00355 [cs, eess], p. 268-276.

Gao, Yunhe, Mu Zhou et Dimitris N. Metaxas (2021). “UTNet : A Hybrid Transformer Architecture for Medical Image Segmentation”. en. In : *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*. Sous la dir. de Marleen de Bruijne et al. Cham : Springer International Publishing, p. 61-71. ISBN : 978-3-030-87199-4.

Gastaud, M., M. Barlaud et G. Aubert (2004). “Combining shape prior and statistical features for active contour segmentation”. In : *IEEE Transactions on Circuits and Systems for Video Technology* 14.5. Conference Name : IEEE Transactions on Circuits and Systems for Video Technology, p. 726-734. ISSN : 1558-2205.

Gatehouse, Peter D. et al. (2010). “Flow measurement by cardiovascular magnetic resonance : a multi-centre multi-vendor study of background phase offset errors that can compromise the accuracy of derived regurgitant or shunt flow measurements”. In : *Journal of Cardiovascular Magnetic Resonance* 12.1, p. 5. ISSN : 1532-429X.

Geiger, A et al. (2013). “Vision meets robotics : The KITTI dataset”. en. In : *The International Journal of Robotics Research* 32.11. Publisher : SAGE Publications Ltd STM, p. 1231-1237. ISSN : 0278-3649.

Germans, Tjeerd et al. (2006). “Structural abnormalities of the inferoseptal left ventricular wall detected by cardiac magnetic resonance imaging in carriers of hypertrophic cardiomyopathy mutations”. eng. In : *Journal of the American College of Cardiology* 48.12, p. 2518-2523. ISSN : 1558-3597.

Gersh, Bernard J. et al. (2011). “2011 ACCF/AHA Guideline for the Diagnosis and Treatment of Hypertrophic Cardiomyopathy : a report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines. Developed in collaboration with the American Association for Thoracic Surgery, American Society of Echocardiography, American Society of Nuclear Cardiology, Heart Failure Society of America, Heart Rhythm Society, Society for Cardiovascular Angiography and Interventions, and Society of Thoracic Surgeons”. eng. In : *Journal of the American College of Cardiology* 58.25, e212-260. ISSN : 1558-3597.

Ghadim, Yalda Zafari et Hamed Azarnoush (2023). *Learning Fast Diffeomorphic Registration for Cardiac Motion Estimation in 3D Echocardiography*. en.

Gibou, Frederic, Ronald Fedkiw et Stanley Osher (2018). "A review of level-set methods and some recent applications". In : *Journal of Computational Physics* 353, p. 82-109. ISSN : 0021-9991.

Goldberg, Andrew V. et Robert E. Tarjan (1988). "A new approach to the maximum-flow problem". In : *Journal of the ACM* 35.4, p. 921-940. ISSN : 0004-5411.

Gong, Hao et al. (2022). "Improving coronary artery imaging in single source CT with cardiac motion correction using attention and spatial transformer based neural networks". In : *Proceedings of SPIE–the International Society for Optical Engineering* 12031, 120311E. ISSN : 0277-786X.

Grosgeorge, D. et al. (2013). "Graph cut segmentation with a statistical shape model in cardiac MRI". In : *Computer Vision and Image Understanding* 117.9, p. 1027-1035. ISSN : 1077-3142.

Grothues, Frank et al. (2002). "Comparison of interstudy reproducibility of cardiovascular magnetic resonance with two-dimensional echocardiography in normal subjects and in patients with heart failure or left ventricular hypertrophy". eng. In : *The American Journal of Cardiology* 90.1, p. 29-34. ISSN : 0002-9149.

Grzeszczyk, Michal K., Szymon Płotka et Arkadiusz Sitek (2022). "Multi-task Swin Transformer for Motion Artifacts Classification and Cardiac Magnetic Resonance Image Segmentation". en. In : *Statistical Atlases and Computational Models of the Heart. Regular and CMRxMotion Challenge Papers*. Sous la dir. d'Oscar Camara et al. Cham : Springer Nature Switzerland, p. 409-417. ISBN : 978-3-031-23443-9.

Gu, Albert et Tri Dao (2024). *Mamba : Linear-Time Sequence Modeling with Selective State Spaces*. arXiv :2312.00752 [cs].

Gu, Albert, Karan Goel et Christopher Ré (2022). *Efficiently Modeling Long Sequences with Structured State Spaces*. arXiv :2111.00396 [cs].

Gupta, Lalit et Thotsapon Sortrakul (1998). "A gaussian-mixture-based image segmentation algorithm". In : *Pattern Recognition* 31.3, p. 315-325. ISSN : 0031-3203.

Habijan, Marija et al. (2021). "Whole Heart Segmentation Using 3D FM-Pre-ResNet Encoder–Decoder Based Architecture with Variational Autoencoder Regularization". en. In : *Applied Sciences* 11.9. Number : 9 Publisher : Multidisciplinary Digital Publishing Institute, p. 3912. ISSN : 2076-3417.

Hammouda, K. et al. (2020). "A New Framework for Performing Cardiac Strain Analysis from Cine MRI Imaging in Mice". en. In : *Scientific Reports* 10.1, p. 7725. ISSN : 2045-2322.

Han, Dongchen, Xuran Pan et al. (2023). "FLatten Transformer : Vision Transformer using Focused Linear Attention". en. In : *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France : IEEE, p. 5938-5948. ISBN : 9798350307184.

Han, Kuan, Haiguang Wen et al. (2018). *Deep Predictive Coding Network with Local Recurrent Processing for Object Recognition*. arXiv :1805.07526 [cs].

Harley, Adam W., Zhaoyuan Fang et Katerina Fragkiadaki (2022). *Particle Video Revisited : Tracking Through Occlusions Using Point Trajectories*. en. arXiv :2204.04153 [cs].

Hatamizadeh, Ali et al. (2021). *UNETR : Transformers for 3D Medical Image Segmentation*. arXiv :2103.10504 [cs, eess].

He, Kaiming, Xiangyu Zhang et al. (2015). *Deep Residual Learning for Image Recognition*. arXiv :1512.03385 [cs].

He, Xuming, R.S. Zemel et M.A. Carreira-Perpinan (2004). "Multiscale conditional random fields for image labeling". In : *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* T. 2. ISSN : 1063-6919, p. II-II.

Held, K. et al. (1997). "Markov random field segmentation of brain MR images". In : *IEEE Transactions on Medical Imaging* 16.6. Conference Name : IEEE Transactions on Medical Imaging, p. 878-886. ISSN : 1558-254X.

Hendrycks, Dan et Kevin Gimpel (2023). *Gaussian Error Linear Units (GELUs)*. arXiv :1606.08415 [cs].

Hering, Alessa et al. (2018). *Enhancing Label-Driven Deep Deformable Image Registration with Local Distance Metrics for State-of-the-Art Cardiac Motion Tracking*. arXiv :1812.01859 [cs].

Hinton, Geoffrey E. et al. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. en. arXiv :1207.0580 [cs].

Hochreiter, Sepp et Jürgen Schmidhuber (1994). "SIMPLIFYING NEURAL NETS BY DISCOVERING FLAT MINIMA". In : *Advances in Neural Information Processing Systems*. T. 7. MIT Press.

— (1997). "Flat Minima". In : *Neural Computation* 9.1, p. 1-42. ISSN : 0899-7667.

Hoge, W.S. (2003). "A subspace identification extension to the phase correlation method [MRI application]". In : *IEEE Transactions on Medical Imaging* 22.2. Conference Name : IEEE Transactions on Medical Imaging, p. 277-280. ISSN : 1558-254X.

Horn, Berthold K. P. et Brian G. Schunck (1981). "Determining optical flow". In : *Artificial Intelligence* 17.1, p. 185-203. ISSN : 0004-3702.

Hu, Jie, Li Shen et al. (2019). *Squeeze-and-Excitation Networks*. en. arXiv :1709.01507 [cs].

Hu, Li, Peng Zhang et al. (2021). "Learning Position and Target Consistency for Memory-based Video Object Segmentation". en. In : *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA : IEEE, p. 4142-4152. ISBN : 978-1-66544-509-2.

Hu, Yipeng, Marc Modat, Eli Gibson, Nooshin Ghavami et al. (2018). "Label-driven weakly-supervised learning for multimodal deformable image registration". In : *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. ISSN : 1945-8452, p. 1070-1074.

Hu, Yipeng, Marc Modat, Eli Gibson, Wenqi Li et al. (2018). "Weakly-supervised convolutional neural networks for multimodal image registration". In : *Medical Image Analysis* 49, p. 1-13. ISSN : 1361-8415.

Huang, Yu-Len et Dar-Ren Chen (2004). "Watershed segmentation for breast tumor in 2-D sonography". In : *Ultrasound in Medicine & Biology* 30.5, p. 625-632. ISSN : 0301-5629.

Huang, Xiaohong, Zhifang Deng et al. (2021). *MISSFormer : An Effective Medical Image Segmentation Transformer*. en. arXiv :2109.07162 [cs].

Huang, Zhaoyang, Xiaoyu Shi et al. (2022). *FlowFormer : A Transformer Architecture for Optical Flow*. arXiv :2203.16194 [cs] version : 4.

Huang, Zhiheng, Davis Liang et al. (2020). *Improve Transformer Models with Better Relative Position Embeddings*. arXiv :2009.13658 [cs].

Hui, Tak-Wai, Xiaoou Tang et Chen Change Loy (2018). *LiteFlowNet : A Lightweight Convolutional Neural Network for Optical Flow Estimation*. arXiv :1805.07036 [cs].

Hundley, W. G. et al. (1995). "Magnetic resonance imaging assessment of the severity of mitral regurgitation. Comparison with invasive techniques". eng. In : *Circulation* 92.5, p. 1151-1158. ISSN : 0009-7322.

Hur, Junhwa et Stefan Roth (2019). "Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation". In : *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA : IEEE, p. 5747-5756. ISBN : 978-1-72813-293-8.

Ibtehaz, Nabil et M. Sohel Rahman (2020). "MultiResUNet : Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation". en. In : *Neural Networks* 121. arXiv :1902.04049 [cs], p. 74-87. ISSN : 08936080.

Ilg, Eddy et al. (2016). *FlowNet 2.0 : Evolution of Optical Flow Estimation with Deep Networks*. arXiv :1612.01925 [cs].

Immanuel, Sp, Dr Bala et Alma George (2011). "A Study on Block Matching Algorithms for Motion Estimation". In : *International Journal on Computer Science and Engineering* 3.

Ioffe, Sergey et Christian Szegedy (2015). "Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift". en. In : *Proceedings of the 32nd International Conference on Machine Learning*. ISSN : 1938-7228. PMLR, p. 448-456.

Isensee, Fabian et al. (2018). *nnU-Net : Self-adapting Framework for U-Net-Based Medical Image Segmentation*. arXiv :1809.10486 [cs].

Islam, Md Amirul et al. (2017). "Gated Feedback Refinement Network for Dense Image Labeling". en. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI : IEEE, p. 4877-4885. ISBN : 978-1-5386-0457-1.

Jaderberg, Max et al. (2015). "Spatial Transformer Networks". In : *Advances in Neural Information Processing Systems*. T. 28. Curran Associates, Inc.

Janai, Joel et al. (2017). "Slow Flow : Exploiting High-Speed Cameras for Accurate and Diverse Optical Flow Reference Data". en. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI : IEEE, p. 1406-1416. ISBN : 978-1-5386-0457-1.

Jiang, Shihao et al. (2021). "Learning to Estimate Hidden Motions with Global Motion Aggregation". en. In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada : IEEE, p. 9752-9761. ISBN : 978-1-66542-812-5.

Jin, Felix Q. et al. (2019). "Comparison of Deep Learning and Classical Image Processing for Skin Segmentation". In : *2019 IEEE International Ultrasonics Symposium (IUS)*. ISSN : 1948-5727, p. 1152-1155.

Kalchbrenner, Nal et al. (2017). *Neural Machine Translation in Linear Time*. arXiv :1610.10099 [cs].

Kamnitsas, Konstantinos et al. (2016). "DeepMedic for Brain Tumor Segmentation". en. In : *Brainlesion : Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Sous la dir. d'Alessandro Crimi et al. Cham : Springer International Publishing, p. 138-149. ISBN : 978-3-319-55524-9.

Kass, Michael, Andrew Witkin et Demetri Terzopoulos (1988). “Snakes : Active contour models”. en. In : *International Journal of Computer Vision* 1.4, p. 321-331. ISSN : 1573-1405.

Katharopoulos, Angelos et al. (2020). “Transformers are RNNs : Fast Autoregressive Transformers with Linear Attention”. en. In : *Proceedings of the 37th International Conference on Machine Learning*. ISSN : 2640-3498. PMLR, p. 5156-5165.

Kawel-Boehm, Nadine et al. (2020). “Reference ranges (“normal values”) for cardiovascular magnetic resonance (CMR) in adults and children : 2020 update”. In : *Journal of Cardiovascular Magnetic Resonance* 22.1, p. 87. ISSN : 1532-429X.

Keskar, Nitish Shirish, Dheevatsa Mudigere et al. (2017). *On Large-Batch Training for Deep Learning : Generalization Gap and Sharp Minima*. arXiv :1609.04836 [cs, math].

Keskar, Nitish Shirish et Richard Socher (2017). *Improving Generalization Performance by Switching from Adam to SGD*. arXiv :1712.07628 [cs, math].

Khan, Salman et al. (2022). “Transformers in Vision : A Survey”. en. In : *ACM Computing Surveys* 54.10s, p. 1-41. ISSN : 0360-0300, 1557-7341.

Kim, Boah, Dong Hwan Kim, Seong Ho Park et al. (2021). “CycleMorph : Cycle consistent unsupervised deformable image registration”. In : *Medical Image Analysis* 71, p. 102036. ISSN : 1361-8415.

Kim, Boah, Jieun Kim, June-Goo Lee et al. (2019). *Unsupervised Deformable Image Registration Using Cycle-Consistent CNN*. en. arXiv :1907.01319 [cs, eess, stat].

Kingma, Diederik P. et Jimmy Ba (2017). *Adam : A Method for Stochastic Optimization*. arXiv :1412.6980 [cs].

Krebs, Julian, Hervé Delingette et al. (2019). “Learning a Probabilistic Model for Diffeomorphic Registration”. en. In : *IEEE Transactions on Medical Imaging* 38.9. arXiv :1812.07460 [cs], p. 2165-2176. ISSN : 0278-0062, 1558-254X.

Krebs, Julian, Tommaso Mansi et al. (2018). *Unsupervised Probabilistic Deformation Modeling for Robust Diffeomorphic Registration*. en. arXiv :1804.07172 [cs].

Krishnaswamy, Deepa et al. (2023). “A Novel 3D-to-3D Diffeomorphic Registration Algorithm With Applications to Left Ventricle Segmentation in MR and Ultrasound Sequences”. en. In : *IEEE Access* 11, p. 3144-3159. ISSN : 2169-3536.

Kruk, Dominika et al. (2017). “Segmentation Integrating Texture and Shape A Priori Applied to Cardiac MR Images”. In.

Kuang, Dongyang (2019). *On Reducing Negative Jacobian Determinant of the Deformation Predicted by Deep Registration Networks*. arXiv :1907.00068 [cs, eess].

Kumar, Vinay et al. (2007). *Robbins Basic Pathology*. en. Elsevier Health Sciences. ISBN : 978-1-4377-0066-4.

Kuo, C. -C. Jay (2016). “Understanding convolutional neural networks with a mathematical model”. In : *Journal of Visual Communication and Image Representation* 41, p. 406-413. ISSN : 1047-3203.

Kuznetsova, Alina et al. (2020). “The Open Images Dataset V4”. en. In : *International Journal of Computer Vision* 128.7, p. 1956-1981. ISSN : 1573-1405.

Kwan, Alan C. et al. (2024). “Deep Learning-Derived Myocardial Strain”. In : *JACC : Cardiovascular Imaging*. ISSN : 1936-878X.

Lamy, Jérôme et al. (2018). “Scan-rescan reproducibility of ventricular and atrial MRI feature tracking strain”. In : *Computers in Biology and Medicine* 92, p. 197-203. ISSN : 0010-4825.

Lee, Lik Chuan et Martin Genet (2019). "Validation of Equilibrated Warping—Image Registration with Mechanical Regularization—On 3D Ultrasound Images". en. In : *Functional Imaging and Modeling of the Heart*. Sous la dir. d'Yves Coudière et al. Cham : Springer International Publishing, p. 334-341. ISBN : 978-3-030-21949-9.

Levner, Ilya et Hong Zhang (2007). "Classification-Driven Watershed Segmentation". In : *IEEE Transactions on Image Processing* 16.5. Conference Name : IEEE Transactions on Image Processing, p. 1437-1445. ISSN : 1941-0042.

Li, Chunming, Chiu-Yen Kao et al. (2008). "Minimization of Region-Scalable Fitting Energy for Image Segmentation". In : *IEEE Transactions on Image Processing* 17.10. Conference Name : IEEE Transactions on Image Processing, p. 1940-1949. ISSN : 1941-0042.

Li, Junlong, Yiheng Xu et al. (2022). "DiT : Self-supervised Pre-training for Document Image Transformer". en. In : *Proceedings of the 30th ACM International Conference on Multimedia*. Lisboa Portugal : ACM, p. 3530-3539. ISBN : 978-1-4503-9203-7.

Li, Ming, Chengjia Wang, Heye Zhang et al. (2020). "MV-RAN : Multiview recurrent aggregation network for echocardiographic sequences segmentation and full cardiac cycle analysis". In : *Computers in Biology and Medicine* 120, p. 103728. ISSN : 0010-4825.

Li, Tianyang, Benzheng Wei et al. (2020). "Direct estimation of left ventricular ejection fraction via a cardiac cycle feature learning architecture". In : *Computers in Biology and Medicine* 118, p. 103659. ISSN : 0010-4825.

Li, Wen, Limin Wang, Wei Li et al. (2017). *WebVision Database : Visual Learning and Understanding from Web Data*. arXiv :1708.02862 [cs].

Li, Xiangtai, Houlong Zhao et al. (2020). "Gated Fully Fusion for Semantic Segmentation". en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 34.07. Number : 07, p. 11418-11425. ISSN : 2374-3468.

Li, Zhaowen, Zhiyang Chen et al. (2021). "MST : Masked Self-Supervised Transformer for Visual Representation". In : *Advances in Neural Information Processing Systems*. T. 34. Curran Associates, Inc., p. 13165-13176.

Lilly, Leonard S. (2012). *Pathophysiology of Heart Disease : A Collaborative Project of Medical Students and Faculty*. en. Google-Books-ID : 0lxSGJYeXikC. Lippincott Williams & Wilkins. ISBN : 978-1-4698-1668-5.

Liu, Pengpeng, Michael Lyu et al. (2019). "SelFlow : Self-Supervised Learning of Optical Flow". In : p. 4571-4580.

Liu, Shan, Bo Yang et al. (2022). "2D/3D Multimode Medical Image Registration Based on Normalized Cross-Correlation". en. In : *Applied Sciences* 12.6. Number : 6 Publisher : Multidisciplinary Digital Publishing Institute, p. 2828. ISSN : 2076-3417.

Liu, Yong, Ran Yu et al. (2022). *Learning Quality-aware Dynamic Memory for Video Object Segmentation*. arXiv :2207.07922 [cs].

Liu, Ze, Yutong Lin et al. (2021). "Swin Transformer : Hierarchical Vision Transformer Using Shifted Windows". en. In : p. 10012-10022.

Liu, Ze, Jia Ning et al. (2022). "Video Swin Transformer". en. In : p. 3202-3211.

Liu, Zexiong et Xuan Yang (2019). "A Squeeze Convolutional Network For MRI Right Ventricle Segmentation". In : *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, p. 697-700.

Liu, Zhe, Yu-Qing Song et al. (2019). “Liver CT sequence segmentation based with improved U-Net and graph cut”. In : *Expert Systems with Applications* 126, p. 54-63. ISSN : 0957-4174.

Liu, Ziwei, Xiaoxiao Li et al. (2018). “Deep Learning Markov Random Field for Semantic Segmentation”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.8. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 1814-1828. ISSN : 1939-3539.

Lloyd, S. (1982). “Least squares quantization in PCM”. In : *IEEE Transactions on Information Theory* 28.2. Conference Name : IEEE Transactions on Information Theory, p. 129-137. ISSN : 1557-9654.

Long, Jonathan, Evan Shelhamer et Trevor Darrell (2015). “Fully Convolutional Networks for Semantic Segmentation”. In : p. 3431-3440.

Loshchilov, Ilya et Frank Hutter (2017). *SGDR : Stochastic Gradient Descent with Warm Restarts*. arXiv :1608.03983 [cs, math].

— (2019). *Decoupled Weight Decay Regularization*. arXiv :1711.05101 [cs, math].

Lotter, William, Gabriel Kreiman et David Cox (2017). *Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning*. arXiv :1605.08104 [cs, q-bio].

Lotz, Joachim et al. (2002). “Cardiovascular Flow Measurement with Phase-Contrast MR Imaging : Basic Facts and Implementation”. In : *RadioGraphics* 22.3. Publisher : Radiological Society of North America, p. 651-671. ISSN : 0271-5333.

Lu, Allen, Shawn S. Ahn et al. (2021). “Learning-Based Regularization for Cardiac Strain Analysis via Domain Adaptation”. en. In : *IEEE Transactions on Medical Imaging* 40.9, p. 2233-2245. ISSN : 0278-0062, 1558-254X.

Lu, Guo, Chunlei Cai et al. (2020). *Content Adaptive and Error Propagation Aware Deep Video Compression*. arXiv :2003.11282 [cs, eess].

Lu, Jiayi, Renchao Jin et al. (2023). “A discontinuity-preserving regularization for deep learning-based cardiac image registration”. en. In : *Physics in Medicine & Biology* 68.9, p. 095024. ISSN : 0031-9155, 1361-6560.

Lu, Lu, Yeonjong Shin et al. (2020). “Dying ReLU and Initialization : Theory and Numerical Examples”. In : *Communications in Computational Physics* 28, p. 1671-1706.

Lu, Yawen, Qifan Wang et al. (2023). “TransFlow : Transformer as Flow Learner”. en. In : *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada : IEEE, p. 18063-18073. ISBN : 9798350301298.

Lu, Ziwei, Guanyu Yang et al. (2019). “Unsupervised Three-Dimensional Image Registration Using a Cycle Convolutional Neural Network”. In : *2019 IEEE International Conference on Image Processing (ICIP)*. ISSN : 2381-8549, p. 2174-2178.

Lucas, Bruce D et Takeo Kanade (1981). “An Iterative Image Registration Technique with an Application to Stereo Vision”. In : *IJCAI’81 : 7th international joint conference on Artificial intelligence*. T. 2. Vancouver, Canada, p. 674-679.

Luo, Hao, Pichao Wang et al. (2021). *Self-Supervised Pre-Training for Transformer-Based Person Re-Identification*. arXiv :2111.12084 [cs].

Luo, Wenjie, Yujia Li et al. (2016). “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks”. In : *Advances in Neural Information Processing Systems*. T. 29. Curran Associates, Inc.

Maas, Andrew L. (2013). “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. In.

MacQueen, James et al. (1967). “Some methods for classification and analysis of multivariate observations”. In : *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. T. 1. 14. Oakland, CA, USA, p. 281-297.

Maes, F. et al. (1997). “Multimodality image registration by maximization of mutual information”. In : *IEEE Transactions on Medical Imaging* 16.2. Conference Name : IEEE Transactions on Medical Imaging, p. 187-198. ISSN : 1558-254X.

Maron, Barry J. (2002). “Hypertrophic cardiomyopathy : a systematic review”. eng. In : *JAMA* 287.10, p. 1308-1320. ISSN : 0098-7484.

Martín-Isla, Carlos et al. (2023). “Deep Learning Segmentation of the Right Ventricle in Cardiac MRI : The M&Ms Challenge”. en. In : *IEEE Journal of Biomedical and Health Informatics* 27.7, p. 3302-3313. ISSN : 2168-2194, 2168-2208.

Martini, Chiara et al. (2010). *Left and Right Ventricle Assessment with Cardiac CT : Validation Study versus Cardiac MRI*.

Marwick, Thomas H. (2006). “Measurement of Strain and Strain Rate by Echocardiography : Ready for Prime Time ?” In : *Journal of the American College of Cardiology* 47.7, p. 1313-1327. ISSN : 0735-1097.

Masutani, Evan M. et al. (2023). “Deep Learning Synthetic Strain : Quantitative Assessment of Regional Myocardial Wall Motion at MRI”. In : *Radiology : Cardiothoracic Imaging* 5.3, e220202. ISSN : 2638-6135.

Maurya, Akansh et al. (2022). *PARSE challenge 2022 : Pulmonary Arteries Segmentation using Swin U-Net Transformer(Swin UNETR) and U-Net*. arXiv :2208.09636 [cs, eess].

Mayer, Nikolaus et al. (2016). “A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation”. en. In : *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv :1512.02134 [cs, stat], p. 4040-4048.

McRobbie, Donald W. et al. (2017). *MRI from Picture to Proton*. en. Google-Books-ID : yj6wDgAAQBAJ. Cambridge University Press. ISBN : 978-1-316-68825-0.

Moore, C. C. et al. (2000). “Three-dimensional systolic strain patterns in the normal human left ventricle : characterization with tagged MR imaging”. eng. In : *Radiology* 214.2, p. 453-466. ISSN : 0033-8419.

Morales, Manuel A., Maaike van den Boomen et al. (2021). “DeepStrain : A Deep Learning Workflow for the Automated Characterization of Cardiac Mechanics”. English. In : *Frontiers in Cardiovascular Medicine* 8. Publisher : Frontiers. ISSN : 2297-055X.

Morales, Manuel A., Julia Cirillo et al. (2023). “Comparison of DeepStrain and Feature Tracking for Cardiac MRI Strain Analysis”. en. In : *Journal of Magnetic Resonance Imaging* 57.5. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/jmri.28374, p. 1507-1515. ISSN : 1522-2586.

Munkhdalai, Tsendsuren, Manaal Faruqui et Siddharth Gopal (2024). *Leave No Context Behind : Efficient Infinite Context Transformers with Infini-attention*. arXiv :2404.07143 [cs].

Myronenko, Andriy (2019). “3D MRI Brain Tumor Segmentation Using Autoencoder Regularization”. In : *Brainlesion : Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Sous la dir. d’Alessandro Crimi et al. Cham : Springer International Publishing, p. 311-320. ISBN : 978-3-030-11726-9.

Nguyen, Thanh Minh et Q. M. Jonathan Wu (2013). “Fast and Robust Spatially Constrained Gaussian Mixture Model for Image Segmentation”. In : *IEEE Tran-*

sactions on Circuits and Systems for Video Technology 23.4. Conference Name : IEEE Transactions on Circuits and Systems for Video Technology, p. 621-635. ISSN : 1558-2205.

Ni, Yangfan et al. (2023). *A Multi-Scale Spatial Transformer U-Net for Simultaneously Automatic Reorientation and Segmentation of 3D Nuclear Cardiac Images*. en. arXiv :2310.10095 [cs, eess].

Niklaus, Simon et Feng Liu (2020). "Softmax Splatting for Video Frame Interpolation". In : p. 5437-5446.

Nikolic, G. et H. J. Marriott (1985). "Left bundle branch block with right axis deviation : a marker of congestive cardiomyopathy". eng. In : *Journal of Electrocardiology* 18.4, p. 395-404. ISSN : 0022-0736.

Noh, Hyeonwoo, Seunghoon Hong et Bohyung Han (2015). "Learning Deconvolution Network for Semantic Segmentation". In : p. 1520-1528.

Oh, Seoung Wug et al. (2019). "Video Object Segmentation Using Space-Time Memory Networks". en. In : *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South) : IEEE, p. 9225-9234. ISBN : 978-1-72814-803-8.

Oktay, Ozan, Enzo Ferrante et al. (2018). "Anatomically Constrained Neural Networks (ACNNs) : Application to Cardiac Image Enhancement and Segmentation". In : *IEEE Transactions on Medical Imaging* 37.2. Conference Name : IEEE Transactions on Medical Imaging, p. 384-395. ISSN : 1558-254X.

Oktay, Ozan, Jo Schlemper et al. (2018). *Attention U-Net : Learning Where to Look for the Pancreas*. arXiv :1804.03999 [cs].

Oord, Aaron van den, Yazhe Li et Oriol Vinyals (2019). *Representation Learning with Contrastive Predictive Coding*. arXiv :1807.03748 [cs, stat].

Osman, N. F. et al. (1999). "Cardiac motion tracking using CINE harmonic phase (HARP) magnetic resonance imaging". eng. In : *Magnetic Resonance in Medicine* 42.6, p. 1048-1060. ISSN : 0740-3194.

Pang, Yanwei et al. (2019). "Towards Bridging Semantic Gap to Improve Semantic Segmentation". en. In : *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South) : IEEE, p. 4229-4238. ISBN : 978-1-72814-803-8.

Panjwani, D.K. et G. Healey (1995). "Markov random field models for unsupervised segmentation of textured color images". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.10. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 939-954. ISSN : 1939-3539.

Papandrianos, Nikolaos I. et al. (2022). "Deep Learning-Based Automated Diagnosis for Coronary Artery Disease Using SPECT-MPI Images". In : *Journal of Clinical Medicine* 11.13, p. 3918. ISSN : 2077-0383.

Parato, Vito Maurizio et al. (2016). "Echocardiographic diagnosis of the different phenotypes of hypertrophic cardiomyopathy". eng. In : *Cardiovascular Ultrasound* 14.1, p. 30. ISSN : 1476-7120.

Peiris, Himashi et al. (2021). "A Volumetric Transformer for Accurate 3D Tumor Segmentation". In : *arXiv :2111.13300 [cs, eess]*. arXiv : 2111.13300.

Peng, Jing et al. (2021). "A Multi-Task Network for Cardiac Magnetic Resonance Image Segmentation and Classification". en. In : *Intelligent Automation & Soft Computing* 29.3, p. 259-272. ISSN : 1079-8587.

Pennell, Dudley J. et al. (2004). “Clinical indications for cardiovascular magnetic resonance (CMR) : Consensus Panel report”. eng. In : *European Heart Journal* 25.21, p. 1940-1965. ISSN : 0195-668X.

Perk, Joep et al. (2012). “European Guidelines on cardiovascular disease prevention in clinical practice (version 2012). The Fifth Joint Task Force of the European Society of Cardiology and Other Societies on Cardiovascular Disease Prevention in Clinical Practice (constituted by representatives of nine societies and by invited experts)”. eng. In : *European Heart Journal* 33.13, p. 1635-1701. ISSN : 1522-9645.

Petersen, Steffen E., Nay Aung et al. (2016). “Reference ranges for cardiac structure and function using cardiovascular magnetic resonance (CMR) in Caucasians from the UK Biobank population cohort”. In : *Journal of Cardiovascular Magnetic Resonance* 19.1, p. 18. ISSN : 1097-6647.

Petersen, Steffen E., Paul M. Matthews et al. (2016). “UK Biobank’s cardiovascular magnetic resonance protocol”. In : *Journal of Cardiovascular Magnetic Resonance* 18, p. 8. ISSN : 1097-6647.

Petit, Olivier et al. (2021). “U-Net Transformer : Self and Cross Attention for Medical Image Segmentation”. In : *arXiv :2103.06104 [cs, eess]*. arXiv : 2103.06104.

Plaksyvyi, Arsen, Maria Skublewska-Paszkowska et Paweł Powroźnik (2023). “A Comparative Analysis of Image Segmentation Using Classical and Deep Learning Approach”. EN. In : *Advances in Science and Technology. Research Journal* Vol. 17.no 6. ISSN : 2299-8624.

Pohle, Regina et Klaus D. Toennies (2001). “Segmentation of medical images using adaptive region growing”. In : *Medical Imaging 2001 : Image Processing*. T. 4322. SPIE, p. 1337-1346.

Poudel, Rudra P. K., Pablo Lamata et Giovanni Montana (2016). *Recurrent Fully Convolutional Neural Networks for Multi-slice MRI Cardiac Segmentation*. arXiv :1608.03974 [cs, stat].

Prasad, Sunil M et al. (2003). “The Cox maze III procedure for atrial fibrillation : long-term efficacy in patients undergoing lone versus concomitant procedures”. In : *The Journal of Thoracic and Cardiovascular Surgery* 126.6, p. 1822-1827. ISSN : 0022-5223.

Puiseux, Thomas et al. (2021). “Numerical simulation of time-resolved 3D phase-contrast magnetic resonance imaging”. en. In : *PLOS ONE* 16.3. Publisher : Public Library of Science, e0248816. ISSN : 1932-6203.

Q. Wang et al. (2020). “ECA-Net : Efficient Channel Attention for Deep Convolutional Neural Networks”. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Journal Abbreviation : 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), p. 11531-11539.

Qayyum, Abdul et al. (2020). *Segmentation of the Myocardium on Late-Gadolinium Enhanced MRI based on 2.5 D Residual Squeeze and Excitation Deep Learning Model*. arXiv :2005.13643 [cs, eess].

Qin, Chen, Wenjia Bai et al. (2018). “Joint Motion Estimation and Segmentation from Undersampled Cardiac MR Image”. In : t. 11074. arXiv :1908.07623 [cs, eess], p. 55-63.

Qin, Chen, Shuo Wang et al. (2020). *Biomechanics-informed Neural Networks for Myocardial Motion Tracking in MRI*. arXiv :2006.04725 [cs, eess].

Qiu, Huaqi et al. (2021). “Learning Diffeomorphic and Modality-invariant Registration using B-splines”. en. In : *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*. ISSN : 2640-3498. PMLR, p. 645-664.

Radau, Perry et al. (2009). “Evaluation Framework for Algorithms Segmenting Short Axis Cardiac MRI.” In : *The MIDAS Journal*.

Raffel, Colin et al. (2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In : *Journal of Machine Learning Research* 21.140, p. 1-67. ISSN : 1533-7928.

Raghu, Maithra et al. (2021). “Do Vision Transformers See Like Convolutional Neural Networks ?” In : *Advances in Neural Information Processing Systems*. Sous la dir. de M. Ranzato et al. T. 34. Curran Associates, Inc., p. 12116-12128.

Raisi-Estabragh, Zahra et al. (2021). “Cardiovascular magnetic resonance imaging in the UK Biobank : a major international health research resource”. In : *European Heart Journal - Cardiovascular Imaging* 22.3, p. 251-258. ISSN : 2047-2404.

Ramachandran, Prajit et al. (2019). “Stand-Alone Self-Attention in Vision Models”. In : *arXiv :1906.05909 [cs]*. arXiv : 1906.05909.

Ranjan, Anurag et Michael J. Black (2016). *Optical Flow Estimation using a Spatial Pyramid Network*. arXiv :1611.00850 [cs].

Razavi, Reza et al. (2003). “Cardiac catheterisation guided by MRI in children and adults with congenital heart disease”. eng. In : *Lancet (London, England)* 362.9399, p. 1877-1882. ISSN : 1474-547X.

Reed, Grant W, Jeffrey E Rossi et Christopher P Cannon (2017). “Acute myocardial infarction”. In : *The Lancet* 389.10065, p. 197-210. ISSN : 0140-6736.

Reinke, Annika et al. (2022). *Common Limitations of Image Processing Metrics : A Picture Story*.

Ren, Zhile et al. (2018). *A Fusion Approach for Multi-Frame Optical Flow Estimation*. arXiv :1810.10066 [cs].

El-Rewaidy, Hossam et Ahmed S. Fahmy (2016). “Improved estimation of the cardiac global function using combined long and short axis MRI images of the heart”. en. In : *BioMedical Engineering OnLine* 15.1. Number : 1 Publisher : BioMed Central, p. 1-14. ISSN : 1475-925X.

Ridnik, Tal et al. (2021). *ImageNet-21K Pretraining for the Masses*. arXiv :2104.10972 [cs].

Ronneberger, Olaf, Philipp Fischer et Thomas Brox (2015). *U-Net : Convolutional Networks for Biomedical Image Segmentation*. en.

Rother, Carsten, Vladimir Kolmogorov et Andrew Blake (2004). “"GrabCut" : interactive foreground extraction using iterated graph cuts”. In : *ACM Transactions on Graphics* 23.3, p. 309-314. ISSN : 0730-0301.

Russakovsky, Olga et al. (2015). “ImageNet Large Scale Visual Recognition Challenge”. en. In : *International Journal of Computer Vision* 115.3, p. 211-252. ISSN : 1573-1405.

Salman, Nader (2010). “From 3D point clouds to feature preserving meshes”. In.

Sandkühler, Robin et al. (2019). “Recurrent Registration Neural Networks for Deformable Image Registration”. In : *Advances in Neural Information Processing Systems*. T. 32. Curran Associates, Inc.

Santurkar, Shibani et al. (2018). “How Does Batch Normalization Help Optimization ?” In : *Advances in Neural Information Processing Systems*. T. 31. Curran Associates, Inc.

Savioli, Nicoló et al. (2018). "Automated segmentation on the entire cardiac cycle using a deep learning work-flow". In : *arXiv :1809.01015 [cs, stat]*. arXiv : 1809.01015.

Scharstein, Daniel et Richard Szeliski (2002). "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". en. In : *International Journal of Computer Vision* 47.1, p. 7-42. ISSN : 1573-1405.

Schulz-Menger, Jeanette et al. (2013). "Standardized image interpretation and post processing in cardiovascular magnetic resonance : Society for Cardiovascular Magnetic Resonance (SCMR) board of trustees task force on standardized post processing". eng. In : *Journal of Cardiovascular Magnetic Resonance : Official Journal of the Society for Cardiovascular Magnetic Resonance* 15.1, p. 35. ISSN : 1532-429X.

Schuster, Andreas et al. (2013). "Cardiovascular magnetic resonance myocardial feature tracking for quantitative viability assessment in ischemic cardiomyopathy". eng. In : *International Journal of Cardiology* 166.2, p. 413-420. ISSN : 1874-1754.

Sehar, Uroosa et Muhammad Luqman Naseem (2022). "How deep learning is empowering semantic segmentation". en. In : *Multimedia Tools and Applications* 81.21, p. 30519-30544. ISSN : 1573-7721.

Seong, Hongje, Junhyuk Hyun et Euntai Kim (2020). "Kernelized Memory Network for Video Object Segmentation". en. In : *Computer Vision – ECCV 2020*. Sous la dir. d'Andrea Vedaldi et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 629-645. ISBN : 978-3-030-58542-6.

Seong, Hongje, Seoung Wug Oh et al. (2021). "Hierarchical Memory Matching Network for Video Object Segmentation". en. In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada : IEEE, p. 12869-12878. ISBN : 978-1-66542-812-5.

Shafarenko, L., M. Petrou et J. Kittler (1997). "Automatic watershed segmentation of randomly textured color images". In : *IEEE Transactions on Image Processing* 6.11. Conference Name : IEEE Transactions on Image Processing, p. 1530-1544. ISSN : 1941-0042.

Shaw, Peter, Jakob Uszkoreit et Ashish Vaswani (2018). *Self-Attention with Relative Position Representations*. arXiv :1803.02155 [cs].

Sheikhjafari, Ameneh et al. (2022). *Unsupervised diffeomorphic cardiac image registration using parameterization of the deformation field*. en. arXiv :2208.13275 [cs, eess].

Shi, Xiaoyu, Zhaoyang Huang, Weikang Bian et al. (2023). *VideoFlow : Exploiting Temporal Cues for Multi-frame Optical Flow Estimation*. arXiv :2303.08340 [cs].

Shi, Xiaoyu, Zhaoyang Huang, Dasong Li et al. (2023). "FlowFormer++ : Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation". en. In : *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada : IEEE, p. 1599-1610. ISBN : 9798350301298.

Shi, Xingjian, Zhourong Chen et al. (2015). "Convolutional LSTM Network : A Machine Learning Approach for Precipitation Nowcasting". In : *Advances in Neural Information Processing Systems*. T. 28. Curran Associates, Inc.

Smistad, Erik et al. (2021). "Real-time temporal coherent left ventricle segmentation using convolutional LSTMs". In : *2021 IEEE International Ultrasonics Symposium (IUS)*. ISSN : 1948-5727, p. 1-4.

Smith, Jimmy, Shalini De Mello et al. (2023). "Convolutional State Space Models for Long-Range Spatiotemporal Modeling". en. In : *Advances in Neural Information Processing Systems* 36, p. 80690-80729.

Smith, Leslie N. (2017). *Cyclical Learning Rates for Training Neural Networks*. arXiv :1506.01186 [cs].

Smith, Leslie N. et Nicholay Topin (2018). *Super-Convergence : Very Fast Training of Neural Networks Using Large Learning Rates*. arXiv :1708.07120 [cs, stat].

Sui, Xiuchao et al. (2022). "CRAFT : Cross-Attentional Flow Transformer for Robust Optical Flow". en. In : *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA : IEEE, p. 17581-17590. ISBN : 978-1-66546-946-3.

Sun, Chen, Abhinav Shrivastava et al. (2017). "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". In : p. 843-852.

Sun, Deqing, Xiaodong Yang et al. (2018). *PWC-Net : CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume*. arXiv :1709.02371 [cs].

Sun, Jing Ping, Zoran B. Popović et al. (2004). "Noninvasive quantification of regional myocardial function using Doppler-derived velocity, displacement, strain rate, and strain in healthy volunteers : effects of aging". In : *Journal of the American Society of Echocardiography* 17.2, p. 132-138. ISSN : 0894-7317.

Tang, C., D. D. Blatter et D. L. Parker (1993). "Accuracy of phase-contrast flow measurements in the presence of partial-volume effects". eng. In : *Journal of magnetic resonance imaging : JMRI* 3.2, p. 377-385. ISSN : 1053-1807.

Tang, Jun (2010). "A color image segmentation algorithm based on region growing". In : *2010 2nd International Conference on Computer Engineering and Technology*. T. 6, p. V6–634–V6-637.

Tang, Yucheng, Dong Yang et al. (2022). "Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis". en. In : p. 20730-20740.

Teed, Zachary et Jia Deng (2020). *RAFT : Recurrent All-Pairs Field Transforms for Optical Flow*. arXiv :2003.12039 [cs].

Tremeau, Alain et Nathalie Borel (1997). "A region growing and merging algorithm to color segmentation". In : *Pattern Recognition* 30.7, p. 1191-1203. ISSN : 0031-3203.

Tuli, Shikhar et al. (2021). *Are Convolutional Neural Networks or Transformers more like human vision ?* arXiv :2105.07197 [cs].

Ulyanov, Dmitry, Andrea Vedaldi et Victor Lempitsky (2017). *Instance Normalization : The Missing Ingredient for Fast Stylization*. arXiv :1607.08022 [cs].

V. Graves, Catharine et al. (2023). "Siamese pyramidal deep learning network for strain estimation in 3D cardiac cine-MR". In : *Computerized Medical Imaging and Graphics* 108, p. 102283. ISSN : 0895-6111.

Vaswani, Ashish et al. (2017). "Attention is All you Need". In : *Advances in Neural Information Processing Systems*. T. 30. Curran Associates, Inc.

Velasco Jimeno, Carlos (2019). "Advanced Imaging Techniques for Cardiovascular Research". Thèse de doct.

Venkatesan, Ragav et Baoxin Li (2017). *Convolutional Neural Networks in Visual Computing : A Concise Guide*. en. Google-Books-ID : bAM7DwAAQBAJ. CRC Press. ISBN : 978-1-351-65032-8.

Vicente, Sara, Vladimir Kolmogorov et Carsten Rother (2008). “Graph cut based image segmentation with connectivity priors”. In : *2008 IEEE Conference on Computer Vision and Pattern Recognition*. ISSN : 1063-6919, p. 1-8.

Vigneault, Davis M. et al. (2018). “omega-Net (Omega-Net) : Fully automatic, multi-view cardiac MR detection, orientation, and segmentation with deep neural networks”. en. In : *Medical Image Analysis* 48, p. 95-106. ISSN : 13618415.

Vinnakota, Kalyan C. et James B. Bassingthwaighte (2004). “Myocardial density and composition : a basis for calculating intracellular metabolite concentrations”. In : *American Journal of Physiology-Heart and Circulatory Physiology* 286.5. Publisher : American Physiological Society, H1742-H1749. ISSN : 0363-6135.

Vos, Bob D. de, Floris F. Berendsen, Max A. Viergever, Hessam Sokooti et al. (2019). “A deep learning framework for unsupervised affine and deformable image registration”. In : *Medical Image Analysis* 52, p. 128-143. ISSN : 1361-8415.

Vos, Bob D. de, Floris F. Berendsen, Max A. Viergever, Marius Staring et al. (2017). “End-to-End Unsupervised Deformable Image Registration with a Convolutional Neural Network”. en. In : t. 10553. arXiv :1704.06065 [cs], p. 204-212.

Waks, Jonathan W et Mark E Josephson (2014). “Mechanisms of Atrial Fibrillation – Reentry, Rotors and Reality”. In : *Arrhythmia & Electrophysiology Review* 3.2, p. 90-100. ISSN : 2050-3369.

Wang, Chengjia, Guang Yang et Giorgos Papanastasiou (2022). “Unsupervised Image Registration towards Enhancing Performance and Explainability in Cardiac and Brain Image Analysis”. en. In : *Sensors* 22.6. Number : 6 Publisher : Multidisciplinary Digital Publishing Institute, p. 2125. ISSN : 1424-8220.

Wang, Haojia, Xicheng Chen et al. (2022). “E-DU : Deep neural network for multimodal medical image segmentation based on semantic gap compensation”. en. In : *Computers in Biology and Medicine* 151, p. 106206. ISSN : 0010-4825.

Wang, Haonan, Peng Cao et al. (2021). “UCTransNet : Rethinking the Skip Connections in U-Net from a Channel-wise Perspective with Transformer”. In : *arXiv :2109.04335 [cs, eess]*. arXiv : 2109.04335.

Wang, Hongyi, Shiao Xie, Lanfen Lin et al. (2022). “Mixed Transformer U-Net for Medical Image Segmentation”. In : *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ISSN : 2379-190X, p. 2390-2394.

Wang, Huiyu, Yukun Zhu et al. (2020). *Axial-DeepLab : Stand-Alone Axial-Attention for Panoptic Segmentation*. arXiv :2003.07853 [cs].

Wang, Jian et Miaomiao Zhang (2020). “DeepFLASH : An Efficient Network for Learning-Based Medical Image Registration”. en. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA : IEEE, p. 4443-4451. ISBN : 978-1-72817-168-5.

Wang, Jing, Shuyu Wang, Wei Liang et al. (2022). “The auto segmentation for cardiac structures using a dual-input deep learning network based on vision saliency and transformer”. en. In : *Journal of Applied Clinical Medical Physics* 23.5. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/acm2.13597, e13597. ISSN : 1526-9914.

Wang, Sinong, Belinda Z. Li et al. (2020). *Linformer : Self-Attention with Linear Complexity*. arXiv :2006.04768 [cs, stat].

Wang, Wenhai, Enze Xie, Xiang Li et al. (2021). *Pyramid Vision Transformer : A Versatile Backbone for Dense Prediction without Convolutions*. arXiv :2102.12122 [cs].

Wang, Xiaoyan, Luyao Wang, Xingyu Zhong et al. (2021). "PaI-Net : A modified U-Net of reducing semantic gap for surgical instrument segmentation". en. In : *IET Image Processing* 15.12. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1049/ipr2.12283, p. 2959-2969. ISSN : 1751-9667.

Wang, Yu, Changyu Sun et al. (2023). "StrainNet : Improved Myocardial Strain Analysis of Cine MRI by Deep Learning from DENSE". In : *Radiology : Cardiothoracic Imaging* 5.3, e220196. ISSN : 2638-6135.

Wang, Yu et Wanjun Zhang (2021). "A Dense RNN for Sequential Four-Chamber View Left Ventricle Wall Segmentation and Cardiac State Estimation". English. In : *Frontiers in Bioengineering and Biotechnology* 9. Publisher : Frontiers. ISSN : 2296-4185.

Wilson, Ashia C et al. (2017). "The Marginal Value of Adaptive Gradient Methods in Machine Learning". In : *Advances in Neural Information Processing Systems*. T. 30. Curran Associates, Inc.

Woo, Sanghyun et al. (2018). "CBAM : Convolutional Block Attention Module". en. In : *Computer Vision – ECCV 2018*. Sous la dir. de Vittorio Ferrari et al. T. 11211. Series Title : Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 3-19. ISBN : 978-3-030-01233-5 978-3-030-01234-2.

Wu, Guangyang, Xiaohong Liu et al. (2023). "AccFlow : Backward Accumulation for Long-Range Optical Flow". en. In : p. 12119-12128.

Wu, Haiping, Bin Xiao et al. (2021). "CvT : Introducing Convolutions to Vision Transformers". In : *arXiv :2103.15808 [cs]*. arXiv : 2103.15808.

Wu, Hao et Jonathan L. Shapiro (2006). "Does overfitting affect performance in estimation of distribution algorithms". en. In : *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. Seattle Washington USA : ACM, p. 433-434. ISBN : 978-1-59593-186-3.

Wu, Kan, Houwen Peng et al. (2021). "Rethinking and Improving Relative Position Encoding for Vision Transformer". en. In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada : IEEE, p. 10013-10021. ISBN : 978-1-66542-812-5.

Wu, Yuxin et Kaiming He (2018). *Group Normalization*. arXiv :1803.08494 [cs].

Xiao, Tete et al. (2021). "Early Convolutions Help Transformers See Better". In : *Advances in Neural Information Processing Systems*. T. 34. Curran Associates, Inc., p. 30392-30400.

Xie, Haozhe, Hongxun Yao et al. (2021). "Efficient Regional Memory Network for Video Object Segmentation". en. In : *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA : IEEE, p. 1286-1295. ISBN : 978-1-66544-509-2.

Xie, Yiting et David Richmond (2018). "Pre-training on Grayscale ImageNet Improves Medical Image Classification". In : p. 0–0.

Xu, Guoping, Xingrong Wu et al. (2021). "LeViT-UNet : Make Faster Encoders with Transformer for Medical Image Segmentation". In : *arXiv :2107.08623 [cs]*. arXiv : 2107.08623.

Xu, Haofei, Jiaolong Yang et al. (2021). "High-Resolution Optical Flow from 1D Attention and Correlation". en. In : *2021 IEEE/CVF International Conference*

on Computer Vision (ICCV). Montreal, QC, Canada : IEEE, p. 10478-10487. ISBN : 978-1-66542-812-5.

Xu, Haofei, Jing Zhang et al. (2022). "GMFlow : Learning Optical Flow via Global Matching". en. In : *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA : IEEE, p. 8111-8120. ISBN : 978-1-66546-946-3.

Xue, Wufeng et al. (2022). "Improved Segmentation of Echocardiography With Orientation-Congruency of Optical Flow and Motion-Enhanced Segmentation". In : *IEEE Journal of Biomedical and Health Informatics*, p. 1.

Yan, Wenjun et al. (2019). "Cine MRI analysis by deep learning of optical flow : Adding the temporal dimension". en. In : *Computers in Biology and Medicine* 111, p. 103356. ISSN : 0010-4825.

Yang, Gengshan et Deva Ramanan (2019). "Volumetric Correspondence Networks for Optical Flow". In : *Advances in Neural Information Processing Systems*. T. 32. Curran Associates, Inc.

Yang, Ruiping, Kun Liu et Yongquan Liang (2024). "A fusion-attention swin transformer for cardiac MRI image segmentation". en. In : *IET Image Processing* 18.1. _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1049/ipr2.12936, p. 105-115. ISSN : 1751-9667.

Yang, Zongxin, Yunchao Wei et Yi Yang (2021). "Associating Objects with Transformers for Video Object Segmentation". In : *Advances in Neural Information Processing Systems*. T. 34. Curran Associates, Inc., p. 2491-2502.

Ye, Meng, Mikael Kanski, Dong Yang, Leon Axel et al. (2024). "Unsupervised Exemplar-Based Image-to-Image Translation and Cascaded Vision Transformers for Tagged and Untagged Cardiac Cine MRI Registration". en. In : *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA : IEEE, p. 7629-7639. ISBN : 9798350318920.

Ye, Meng, Mikael Kanski, Dong Yang, Qi Chang et al. (2021). "DeepTag : An Unsupervised Deep Learning Method for Motion Tracking on Cardiac Tagging Magnetic Resonance Images". en. In : p. 7261-7271.

Ying, Xue (2019). "An Overview of Overfitting and its Solutions". en. In : *Journal of Physics : Conference Series* 1168, p. 022022. ISSN : 1742-6588, 1742-6596.

Yu, Fisher et Vladlen Koltun (2016). *Multi-Scale Context Aggregation by Dilated Convolutions*. arXiv :1511.07122 [cs].

Yu, Hanchao, Xiao Chen et al. (2020). *Motion Pyramid Networks for Accurate and Efficient Cardiac Motion Estimation*. arXiv :2006.15710 [cs, eess].

Yu, Hanchao, Shanhui Sun et al. (2020). "FOAL : Fast Online Adaptive Learning for Cardiac Motion Estimation". en. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA : IEEE, p. 4312-4322. ISBN : 978-1-72817-168-5.

Yue, Qian et al. (2019). "Cardiac Segmentation from LGE MRI Using Deep Neural Network Incorporating Shape and Spatial Priors". en. In : *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Sous la dir. de Dinggang Shen et al. T. 11765. Series Title : Lecture Notes in Computer Science. Cham : Springer International Publishing, p. 559-567. ISBN : 978-3-030-32244-1 978-3-030-32245-8.

Yushkevich, Paul A. et al. (2006). "User-guided 3D active contour segmentation of anatomical structures : Significantly improved efficiency and reliability". In : *NeuroImage* 31.3, p. 1116-1128. ISSN : 1053-8119.

Z. Liu et al. (2021). "Swin Transformer : Hierarchical Vision Transformer using Shifted Windows". In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Journal Abbreviation : 2021 IEEE/CVF International Conference on Computer Vision (ICCV), p. 9992-10002.

Zeiler, Matthew D. et Rob Fergus (2014). "Visualizing and Understanding Convolutional Networks". en. In : *Computer Vision – ECCV 2014*. Sous la dir. de David Fleet et al. Cham : Springer International Publishing, p. 818-833. ISBN : 978-3-319-10590-1.

Zhang, Dongqing, Ilknur Icke et al. (2018). "A multi-level convolutional LSTM model for the segmentation of left ventricle myocardium in infarcted porcine cine MR images". In : *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. ISSN : 1945-8452, p. 470-473.

Zhang, Feihu, Oliver J. Woodford et al. (2021). "Separable Flow : Learning Motion Cost Volumes for Optical Flow Estimation". en. In : p. 10807-10817.

Zhang, Liang, Georgios Vasileios Karanikolas et al. (2018). "FULLY AUTOMATIC SEGMENTATION OF THE RIGHT VENTRICLE VIA MULTI-TASK DEEP NEURAL NETWORKS". In : *Proceedings of the ... IEEE International Conference on Acoustics, Speech, and Signal Processing. ICASSP (Conference)* 2018, p. 6677-6681. ISSN : 1520-6149.

Zhang, Wei, Kazuyoshi Itoh et al. (1990). "Parallel distributed processing model with local space-invariant interconnections and its optical architecture". In : *Applied optics* 29.32, p. 4790-4797.

Zhang, Wei, Jun Tanida et al. (1988). "Shift-invariant pattern recognition neural network and its optical architecture". In : *Proceedings of annual conference of the Japan Society of Applied Physics*. T. 564. Montreal, CA.

Zhang, Xiaoran, Chenyu You et al. (2022). *Learning correspondences of cardiac motion from images using biomechanics-informed modeling*. arXiv :2209.00726 [cs, eess].

Zhang, Zijun, Lin Ma et al. (2018). *Normalized Direction-preserving Adam*. arXiv :1709.04546 [cs, stat].

Zhao, Chengqian, Cheng Feng et al. (2020). "OF-MSRN : Optical Flow-Auxiliary Multi-Task Regression Network for Direct Quantitative Measurement, Segmentation and Motion Estimation". en. In : *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01. Number : 01, p. 1218-1225. ISSN : 2374-3468.

Zhao, Hengshuang, Jianping Shi et al. (2017). *Pyramid Scene Parsing Network*. arXiv :1612.01105 [cs].

Zhao, Shengyu, Yue Dong et al. (2019). "Recursive Cascaded Networks for Unsupervised Medical Image Registration". en. In : *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South) : IEEE, p. 10599-10609. ISBN : 978-1-72814-803-8.

Zhao, Shiyu, Long Zhao et al. (2022). "Global Matching with Overlapping Attention for Optical Flow Estimation". en. In : *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA : IEEE, p. 17571-17580. ISBN : 978-1-66546-946-3.

Zhao, Ziyuan, Jinxuan Hu et al. (2022). “MMGL : Multi-Scale Multi-View Global-Local Contrastive learning for Semi-supervised Cardiac Image Segmentation”. en. In : *2022 IEEE International Conference on Image Processing (ICIP)*. arXiv :2207.01883 [cs, eess], p. 401-405.

Zhou, Hong-Yu, Jiansen Guo et al. (2022). “nnFormer : Interleaved Transformer for Volumetric Segmentation”. In : *arXiv :2109.03201 [cs]*. arXiv : 2109.03201.

Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee et al. (2018). “UNet++ : A Nested U-Net Architecture for Medical Image Segmentation”. In : *arXiv :1807.10165 [cs, eess, stat]*. arXiv : 1807.10165.

Zhu, Lianghui, Bencheng Liao et al. (2024). *Vision Mamba : Efficient Visual Representation Learning with Bidirectional State Space Model*. arXiv :2401.09417 [cs].

Zhu, Xizhou, Weijie Su et al. (2021). *Deformable DETR : Deformable Transformers for End-to-End Object Detection*. arXiv :2010.04159 [cs].

Zhuoran, Shen et al. (2021). “Efficient Attention : Attention with Linear Complexities”. en. In : *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA : IEEE, p. 3530-3538. ISBN : 978-1-66540-477-8.

Zoni-Berisso, Massimo et al. (2014). “Epidemiology of atrial fibrillation : European perspective”. In : *Clinical Epidemiology* 6, p. 213-220. ISSN : 1179-1349.