La vraie utilité de l'adaptation des paramètres des couches de normalisation d'un CNN en Fully Test Time Adaptation

par

Ghassen BAKLOUTI

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE
M. Sc. A.

MONTRÉAL, LE 19 SEPTEMBRE, 2023

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Ismail Ben Ayed, directeur de mémoire
Département de génie des systèmes, École de technologie supérieure

Mme. Houda Bahig, codirectrice
Département de radio-oncologie du Centre Hospitalier de l'Université de Montréal

M. Matthew Toews, président du jury
Département de génie des systèmes, École de technologie supérieure

M. José Dolz, membre du jury
Département de génie logiciel et des TI, École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 11 SEPTEMBRE 2023

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Tout d'abord je tiens à exprimer ma profonde gratitude envers mon directeur de recherche, Professeur Ismail Ben Ayed, et ma codirectrice de recherche, Docteur Houda Bahig, pour leur soutien, leur confiance et leur motivation tout au long de cette aventure intellectuelle. Vos conseils éclairés et votre expertise précieuse ont été d'une valeur inestimable pour la réussite de ce projet et m'ont aidé énormément pour me développer et acquérir des nouvelles connaissances qui me seront utiles pendant toute ma carrière.

Je tiens également à remercier tous les collègues du laboratoire LIVIA et du centre de recherche du CHUM, tous les amis de classe et tous les professeurs de l'École de technologie supérieure qui ont partagé leurs expériences, leurs idées et leur soutien au cours de cette aventure. Leur esprit collaboratif a grandement enrichi mon expérience de recherche. Merci !

Mes remerciements s'adressent aussi à mes amis notamment ceux qui tiennent une place très particulière dans ma vie et qui étaient toujours là pour m'encourager et me supporter. Merci pour tous les moments de rire et les agréables souvenirs que nous avons partagés et pour ceux à venir.

Enfin, je souhaite exprimer ma reconnaissance envers ma famille, mes parents Sami et Wissal et mes deux petites fleurs Chahd et Lina. Votre amour éternel et votre support inconditionnel ont été toujours les piliers qui m'ont permis d'atteindre cet objectif et tout autre objectif dans ma vie. Je suis extrêmement reconnaissant d'avoir une famille aussi aimante et attentionnée à mes côtés. Je vous aime infiniment !

La vraie utilité de l'adaptation des paramètres des couches de normalisation d'un CNN en Fully Test Time Adaptation

Ghassen BAKLOUTI

RÉSUMÉ

Avec les résultats prometteurs achevés par les méthodes qui se basent sur l'approche du Fully Test Time Adaptation, la communauté de la vision par ordinateur apporte un intérêt de recherche de plus en plus important pour étudier cette approche. L'élément caractéristique de cette approche est qu'elle abandonne l'adaptation d'un réseau de neurones à un changement de distribution de données lors de la phase d'apprentissage pour restreindre cet ajustement à la phase d'inférence pour chaque nouveau lot de test. Dans la littérature, la plupart des méthodes qui étudient cette approche focalisent sur l'adaptation des paramètres des couches de normalisation des CNNs, ce qui permet d'obtenir des bons résultats. Ainsi la question qui se pose, est ce que vraiment on a besoin d'adapter tous les paramètres des couches de normalisation pour achever ces résultats ? Pour répondre à cette question on propose ce travail pour étudier l'utilité et l'efficacité d'adapter les différents paramètres des couches de normalisation d'un CNN lors du FTTA. Dans ce contexte, on a proposé une étude comparative qui discute l'effet de divers éléments sur l'efficacité de l'application du FTTA à la segmentation bimodale des tumeurs de la tête et du cou en utilisant les deux modalités CT-Scan et PET-Scan. Les résultats obtenus montrent que les paramètres à adapter représentent l'élément le plus critique en appliquant l'approche FTTA et prouvent que le gain achevé en adaptant les paramètres appris par le réseau (alpha et beta) des couches de normalisation d'un CNN est négligeable, en comparaison au gain obtenu en choisissant les bons paramètres statistiques de ces couches. Ceci pose une vraie question sur l'utilité d'adapter les paramètres alpha et beta souvent utilisé en FTTA dans la littérature. Pour généraliser ces résultats, on a étendu notre étude vers un scénario de classification d'images naturelles en utilisant la base CIFAR. Les résultats obtenus pour ce deuxième scénario confirment bien nos premiers résultats et l'interrogation posée. Le code de ce projet se trouve dans le répertoire GitHub suivant : https://github.com/ghassenbaklouti/FTTA-For-HNTS.

Mots-clés: Fully Test Time Adaptation, Paramètres à adapter des couches de normalisation, Segmentation bimodale des tumeurs de la tête et du cou, CT-Scan, Pet-Scan, CIFAR

The real usefulness of adapting batch normalization parameters in Fully Test Time Adaptation

Ghassen BAKLOUTI

ABSTRACT

With the promising results of Fully Test Time Adaptation methods, the computer vision community has increased its research interest in thoroughly investigating this approach. The fundamental feature of a Fully Test Time Adaptation method is that it foregoes updating a neural network to a data shift during the learning phase, limiting the adaptation to the inference phase on each new test batch. Examining the literature, we found that most works studying this approach focus on updating batch normalization parameters. While the obtained results are encouraging, one interesting question arises : do we really need to update all the batch normalization parameters to increase the model performance in inference ? To answer this question, we propose to study the real usefulness of adapting batch normalization parameters in Fully Test Time Adaptation. In this context, we have carried out a comparative study to illustrate how FTTA performs on models built for head and neck tumor segmentation from CT-Scan and PET-Scan under various settings. Our findings show that the parameters to be adapted in FTTA represent the most critical factor to be considered. It also shows that the gain obtained by updating the batch norm parameters (alpha and beta) is negligible compared to the improvement obtained by setting the right batch norm statistical parameters that correspond to the actual batch test. This casts doubt on the usefulness of updating the batch normalization's scale and bias parameters often used in FTTA in the literature. For more generalizability, we extended our study to a second scenario related to natural image classification using the CIFAR dataset. The new results confirm our first findings, strengthening our doubt about the utility of adapting the batch normalization's parameters in FTTA. The code of this project can be found in the following GitHub repository : https://github.com/ghassenbaklouti/FTTA-For-HNTS.

Keywords: Fully Test Time Adaptation, Batch norm parameters, Head and Neck Tumors segmentation, CT-Scan, Pet-Scan, CIFAR

TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 REVUE LITTÉRAIRE 5
1.1 La segmenation des images médicales 5
1.2 Les méthodes d'apprentissage : supervisées, faiblement supervisées et non supervisées 7
1.2.1 Complètement supervisées 7
1.2.2 Non supervisées 7
1.2.3 Faiblement supervisées 8
1.2.4 Résumé des méthodes d'apprentissage 9
1.3 Le problème de généralisation dans la littérature 10
1.3.1 Data Shift 10
1.3.2 Les solutions proposées dans la littérature pour résoudre le problème de généralisation 12
1.3.2.1 Fine Tuning 13
1.3.2.2 L'adaptation du domaine 13
1.3.2.3 Source Free Domain Adaptaion 14
1.3.2.4 Test Time Training 15
1.3.2.5 Fully Test Time Adaptation 17
1.3.2.6 Comparaison des différentes procédures d'adaptation 20

CHAPITRE 2 MISE EN SITUATION 23
2.1 Motivation 23
2.1.1 Les limites liées à la modalité CT-Scan 24
2.1.2 Les limites liées à la modalité PET-Scan 25
2.1.3 Les limites liées à la vérité terrain 26
2.1.4 La co-ségmentation est la solution 27
2.2 Les défis reliés à la procédure FTTA 28
2.2.1 Qu'est-ce qu'on adapte 29
2.2.2 Quelle fonction de perte doit-on utiliser pour faire l'adaptation 31
2.3 Définition du problème 32

CHAPITRE 3 FTTA FOR HEAD AND NECK TUMORS SEGMENTATION 33
3.1 FTTA for Head and Neck Tumors Segmentation 33
3.1.1 Phase de l'apprentissage 33
3.1.2 Phase de l'adaptation, l'application du FTTA 33
3.2 Base de données 35
3.3 Le réseau de neurones 37
3.4 Expérimentations et Résultats 38
3.4.1 Prétraitement des données 38

3.4.2 Entraînement et FTTA 40
3.4.3 Résultats 41
3.4.4 Étude Comparative 42
3.4.4.1 Mode d'adaptation, continu ou discontinu 43
3.4.4.2 L'effet du nombre d'étapes 45
3.4.4.3 L'impact des fonctions de perte 47
3.4.4.4 L'impact des paramètres à adapter 50

CHAPITRE 4 FTTA FOR CIFAR10 53
4.1 Base de données 53
4.2 Le réseau de neurones 56
4.3 La démarche expérimentale 56
4.4 Résultats 57

CONCLUSION ET RECOMMANDATIONS 65

BIBLIOGRAPHIE 67

LISTE DES TABLEAUX

Page

Tableau 1.1 Comparaison entre les différentes procédures proposées dans la littérature pour résoudre le problème du changement de la distribution de données entre la base d'entraînement et la base de test 21

Tableau 3.1 La liste des centres dont été collectés les données la base de données de la deuxième édition 2021 de la compétition Hecktor et les machines scanneurs utilisées lors de la collection des données Adapté de (Andrearczyk *et al.* (2022)) 36

Tableau 3.2 La décomposition des données des cinq centres pour la phase d'apprentissage et la phase d'inférence 37

Tableau 3.3 Les résultats de la méthode Tent appliquée à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue et qui se base sur une seule étape d'adaptation pour chaque nouveau sujet 41

Tableau 3.4 Comparaison entre le mode continu et discontinu de la méthode Tent appliquée à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se base sur une seule étape d'adaptation par nouveau sujet 43

Tableau 3.5 Comparaison sur l'effet du nombre d'étapes lors de l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue 45

Tableau 3.6 Comparaison sur l'impact des fonctions de perte pour l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue et qui se base sur une seule étape d'adaptation par nouveau sujet 49

Tableau 3.7 Comparaison à propos l'effet des paramètres des couches de normalisation sur la performance finale de l'ajustement du modèle 51

Tableau 4.1 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit gaussien de sévérités différentes appliquée à la base CIFAR-10 58

Tableau 4.2 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit de photon de sévérités différentes appliquée à la base CIFAR-10 59

Tableau 4.3 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit impulsionnel de sévérités différentes appliquée à la base CIFAR-10 59

Tableau 4.4 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de mise au point de sévérités différentes appliquée à la base CIFAR-10 59

Tableau 4.5 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de vitre de sévérités différentes appliquée à la base CIFAR-10 60

Tableau 4.6 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou cinétique de sévérités différentes appliquée à la base CIFAR-10 60

Tableau 4.7 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de zoom de sévérités différentes appliquée à la base CIFAR-10 60

Tableau 4.8 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de neige de sévérités différentes appliquée à la base CIFAR-10 61

Tableau 4.9 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de givre de sévérités différentes appliquée à la base CIFAR-10 61

Tableau 4.10 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de brouillard de sévérités différentes appliquée à la base CIFAR-10 61

Tableau 4.11 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de luminosité de sévérités différentes appliquée à la base CIFAR-10 62

Tableau 4.12 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de contraste de sévérités différentes appliquée à la base CIFAR-10 62

Tableau 4.13 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature transformation élastique de sévérités différentes appliquée à la base CIFAR-10 62

Tableau 4.14 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature compression jpeg de sévérités différentes appliquée à la base CIFAR-10 63

Tableau 4.15 Moyenne du taux de classification par sévérité pour les différentes configurations du FTTA d'un CNN Resnet-26 aux corruptions citées dans la partie 4.1 appliquées à la base CIFAR-10 63

LISTE DES FIGURES

Page

Figure 1.1 Exemple de la segmentation d'une tumeur de la tête et du cou. Les pixels en rouge font partie de la tumeur alors que les autres pixels n'appartiennent pas à cette région 6

Figure 1.2 Exemple d'une faible annotation, sous la forme de traits et de points, pour une tâche de segmentation des organes abdominaux Tirée de (Luo *et al.* (2022)) 9

Figure 1.3 Exemple d'un changement de la distribution de données entre une PET-Scan et une CT-Scan de la tête et du cou d'un même patient Tirée de (Andrearczyk *et al.* (2022)) 12

Figure 2.1 Exemple d'une CT-Scan d'un patient avec une tumeur de la tête et du cou entouré par les tissus mous présents dans cette région 25

Figure 2.2 Exemple d'une PET-Scan d'un patient avec une tumeur de la tête et du cou qui se voit comme une zone de chaleur de résolution médiocre Tirée de (Andrearczyk *et al.* (2022)) 26

Figure 3.1 L'architecture UNET 37

Figure 3.2 Bloc contracteur 38

Figure 3.3 Bloc extracteur 38

Figure 3.4 Exemple de la procédure de prétraitement des données appliquée à une CT-Scan Tirée de (Andrearczyk *et al.* (2022)) 39

Figure 3.5 Exemple de la procédure de prétraitement des données appliquée à une PET-Scan Tirée de (Andrearczyk *et al.* (2022)) 40

Figure 3.6 Comparaison qualitative entre les performances de segmentation achevée par le modèle entrainé avant et après l'adaptation 42

Figure 3.7 L'effet du nombre d'étapes sur le temps nécessaire pour effectuer l'adaptation à un seul sujet, donné en seconde 47

Figure 4.1 Comparaison qualitative entre les différentes sévérités de corruptions appliquées à la base CIFAR-10 pour un exemple de bruit impulsionnel 54

Figure 4.2 Comparaison qualitative entre les différentes corruptions utilisées dans ce travail 55

Figure 4.3 Bloc résiduel 56

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CT-Scan	Tomodensitométrie
PET-Scan	Tomographie par émission de positrons
MRI-Scan	Imagerie par résonance magnétique
IA	Intelligence artificielle
CNN	Réseau de neurones convolutif
ML	Machine Learning
FSL	Apprentissage totalement supervisé
GAN	Réseau de neurones génératif
WSL	Apprentissage faiblement supervisé
UL	Apprentissage non supervisé
GPT	Generative Pre-trained transformer
FT	Fine Tuning
DA	Domain adaptation
SFDA	Source Free Domain Adaptation
KL	Kullback-Leibler
FTTA	Fully Test Time Adaptation
BN	Batch Normalization
Hecktor	HEad and neCK TumOR
MICCAI	Medical Image Computing and Computer Assisted Intervention
DSC	Indice de similarité de Sørensen-Dice

LISTE DES SYMBOLES ET UNITÉS DE MESURE

k	Classe d'une donnée
i	Pixel
Ω	Image
Ω_s	Image source
Ω_t	Image cible
$\mathcal{I}_m$	Ensemble d'images
θ	Paramètres d'un modèle ML
$f(\theta)$	Modèle ML de paramètres θ
$\mathcal{Y}_i$	Vecteur ground truth associé au pixel i
$\mathcal{L}$	Fonction de perte
$\mathcal{L}_s$	Fonction de perte supervisée
$\mathcal{L}_{us}$	Fonction de perte non supervisée
$\hat{\mathcal{Y}}(i,\theta)$	Vecteur prédiction du modèle $f(\theta)$ pour le pixel i
K	Nombre de classes
λ	paramètre de pondération
$\mathcal{X}$	Lot de données
$\mathcal{X}'$	Lot d'activations
$\mathcal{X}_s$	Données du domaine source
$\mathcal{X}_t$	Données du domaine cible
$\mathcal{Y}_s$	Étiquettes du domaine source
$\mathcal{Y}_t$	Étiquettes du domaine cible
x_t	Donnée du domaine cible
$\hat{y}_t$	Prédiction associée à la donnée x_t

D_s	Domaine source
D_t	Domaine cible
$\mathcal{P}(.)$	Distribution de données
$E(.)$	Moyenne
$Var(.)$	Deviation standard
$g(.)$	Fonction d'activation
I	Intensité
$\mathcal{H}$	Entropie
(u, v)	Les coordonnées d'une pixel i dans une image Ω

INTRODUCTION

L'intégration des réseaux de neurones à convolutions dans le domaine de l'imagerie médicale a connu une évolution rapide dans les dernières années. Récemment les modèles CNNs sont largement utilisés pour diverses tâches liées à l'analyse et l'interprétation des images médicales comme la classification de ces images en différentes catégories, la détection des anomalies à partir de ces images et la segmentation de ces images en différentes régions d'intérêts. Une bonne performance de ces modèles nécessite toujours un apprentissage extensif sur des données de la même distribution ou des distributions similaires à celles de la phase d'inférence. Ainsi une dégradation majeure de la précision du modèle ML est attendue une fois la distribution de données subite des changements significatifs lors de l'inférence.

L'adaptation des modèles déjà entrainés pour des nouvelles distributions de données a été toujours considéré comme un centre d'intérêt par la communauté scientifique de la vision par ordinateur. Dans ce cadre, plusieurs procédures ont été developpées pour s'attaquer à ce problème comme Fine Tuning (Yosinski, Clune, Bengio & Lipson (2014)), Domain Adaptation (Zhu, Park, Isola & Efros (2017)), Source Free Doamin Adaptation (Bateson, Kervadec, Dolz, Lombaert & Ayed (2022a)), Test Time Training (Sun *et al.* (2020)) et Fully Test Time Adaptation (Wang, Shelhamer, Liu, Olshausen & Darrell (2020)). Quoique chacune de ces procédures puisse aboutir à une amélioration de la performance du modèle adapté selon le type des données qu'elle utilise (données annotées, données non annotées, données du domaine source, données du domaine cible...) et les contraintes qu'elle impose pour l'appliquer, on remarque une tendance vers la procédure Fully Test Time Adaptation dans la période dernière, puisqu'elle ne pose pas beaucoup de conditions pour l'utiliser tout en permettant d'avoir de bons résultats.

Le concept du Fully Test Time Adaptation est d'adapter le modèle à chaque nouveau lot de données du domaine cible lors de la phase d'inférence avant de le traiter sans aucune adaptation préalable du modèle au domaine cible lors de la phase d'apprentissage. Cet ajustement se fait

par mettre à jour les paramètres du modèle pour être optimaux à ce nouveau lot sans aucune supervision. De cette façon tout ce dont nous avons besoin pour appliquer cette procédure est le modèle à adapter et le lot du domaine cible non annoté pour lequel on va faire l'adaptation. Avec sa simplicité, les résultats obtenus dans la littérature en relation avec le Fully Test Time Adaptation sont tous prometteurs (Wang *et al.* (2020), Niu *et al.* (2023)), Bateson, Lombaert & Ben Ayed (2022b)). Cependant cette procédure n'est pas encore bien explorée au niveau des tâches en relation avec l'imagerie médicale et pose beaucoup de questions à propos les paramètres du modèle à adapter, comment doit-on faire cette adaptation et la performance finale achevée.

Dans ce projet de recherche on vise à appliquer le Fully Test Time Adaptation pour adapter des modèles déjà entrainés pour faire la segmentation des tumeurs de la tête et du cou à des nouvelles distributions de données afin de vérifier la capacité de cette procédure à améliorer la performance de ces modèles d'une part et d'autre part étudier l'impact de plusieurs éléments comme la fonction de perte et les paramètres à adapter sur son application.

Le premier chapitre présente une revue littéraire sur notre sujet de recherche. Il commence par présenter le problème de segmentation des images médicales et les méthodes d'apprentissage les plus fréquentes pour aborder ce problème avant d'introduire le problème du changement des distributions de données entre la phase d'apprentissage et la phase d'inférence. Par la suite, il donne un aperçu sur les différentes procédures citées dans la littérature pour adapter les modèles déjà entrainés à des nouvelles distributions, soit le Fine Tuning, le Doamin Aaptation, le Source Free Domain Adaptation, le Test Time Training et le Fully Test Time Adaptation, tout en expliquant les principales différences entre ces procédures, ce qui permet de les classer selon des catégories.

Le deuxième chapitre met en situation notre problème de recherche, soit la segmentation des tumeurs de la tête et du cou en s'appuyant sur les deux modalités CT-Scan et PET-Scan. Il commence par motiver ce sujet et présenter son état d'art avant d'introduire les difficultés

souvent rencontrées dans la littérature pour apprendre la segmentation automatique des tumeurs de la tête et du cou, soit les limites liées aux deux modalités CT-Scan/PET-Scan et les limites liées à la variabilité et son impact sur la vérité terrain. Il présente ensuite la co-ségmentation comme solution proposée dans la littérature pour adresser ces limites et se termine par expliquer les défis potentiels qu'on peut rencontrer en appliquant le Fully Test Time Adaptation pour ce problème.

Dans le troisième chapitre, on présente la démarche expérimentale suivie pour appliquer le Fully Test Time Adaptation pour adaptater des modèles entrainés pour faire la segmentation automatique des tumeurs de la tête et du cou à des nouvelles distributions de données. Les résultats obtenus mettent en évidence l'amélioration des performances de ces modèles suite à cette application. Nous présentons ensuite une étude comparative qui discute l'impact des différents paramètres sur la performance finale achevée par le Fully Test Time Adaptation comme le mode d'adaptation, la fonction de perte utilisée et les paramètres à adapter. Ceci nous mène à la conclusion qui suit : trouver les bons paramètres statistiques à utiliser au niveau des couches de normalisations d'un modèle ML lors de l'inférence a un impact plus lourd qu'adapter les paramètres de ces couches appris par le réseau souvent utilisé dans la littérature.

Pour donner un aspect plus général aux résultats obtenus après l'étude comparative du troisième chapitre, nous avons étendu nos expérimentations vers un autre scénario de classification des images naturelles. Le but est d'adapter un modèle ML déjà entrainé avec la base CIFAR à des nouvelles distributions de données obtenues en appliquant différents types de corruptions à cette base. Dans le quatrième chapitre, nous présentons la démarche expérimentale suivie pour implémenter ce scénario avant de présenter les différents résultats trouvés et qui confirment les résultats obtenus précédemment.

La dernière partie de cette thèse est la conclusion qui donne un bref résumé de ce travail de recherche et présente des futurs axes de recherche sur lesquels on peut travailler.

CHAPITRE 1

REVUE LITTÉRAIRE

Ce chapitre fournit un aperçu du problème de segmentation des images médicales, avant de présenter une revue des méthodes d'apprentissage fréquemment employées pour aborder ce sujet. Il expose aussi le fameux problème de généralisation très souvent rencontré au cours de la phase d'inférence et analyse les diverses techniques offertes dans la littérature pour le résoudre.

1.1 La segmenation des images médicales

Avec la diversité des modalités d'images médicales présentes aujourd'hui comme la tomodensitométrie (CT-Scan), l'imagerie par résonance magnétique (MRI-Scan) et la tomographie par émission de positrons (PET-Scan) et leurs utilitées pour faire des analyses métaboliques et anatomiques d'une région particulière du corps, la segmentation de ces images se considère comme l'une des pratiques les plus communes dans le milieu clinique pour diagnostiquer et puis planifier et suivre l'avancement d'un traitement potentiel. Cette tâche, bien qu'elle ressemble au problème de segmentation des images naturelles, comme le but d'assigner une classe k à chaque pixel i appartenant à une image Ω afin de visualiser les différentes structures existentes dans l'image, elle nécessite obligatoirement la présence d'un spécialiste, médecin radiologue, afin de faire et/ou valider cette assignation. Cette exigence, conjuguée à la confidentialité appliquée aux données médicales, joue un rôle important dans la distinction de ce problème par la communauté scientifique lors de ces travaux de recherche. Ainsi on parle essentiellement du problème du manque de données, qui se voit à travers une simple comparaison entre les données disponibles pour faire l'apprentissage de la segmentation automatique relative aux images naturelles et celle relative aux images médicales. Dans le deuxième cas, les données sont beaucoup moins disponible et beaucoup plus coûteux à avoir.

En se référant à la littérature, on trouve que l'intérêt accordé par la communauté scientifique à la segmentation des images médicales a pris un aspect croissant avec l'intégration des réseaux de neurones à convolutions (CNN) dans cette branche pour atteindre plus de 2000 articles

scientifiques en 2021 (Niyas, Pawan, Kumar & Rajan (2022)). Cet intérêt a donné lieu à plusieurs architectures et méthodes puissantes qui ont permis d'obtenir des résultats satisfaisants sur plusieurs tâches de segmentation d'organes et de tumeurs (Litjens *et al.* (2017)) ouvrant en même temps la porte sur d'autres défis comme la capacité d'un modèle à généraliser à des nouvelles données.

La figure 1.1 montre une tâche de segmentation des images médicales. En effet, c'est une CT-Scan qui présente la partie de la tête et du cou d'un patient qui souffre d'un cancer. L'objectif est de délimiter les contours de la tumeur.

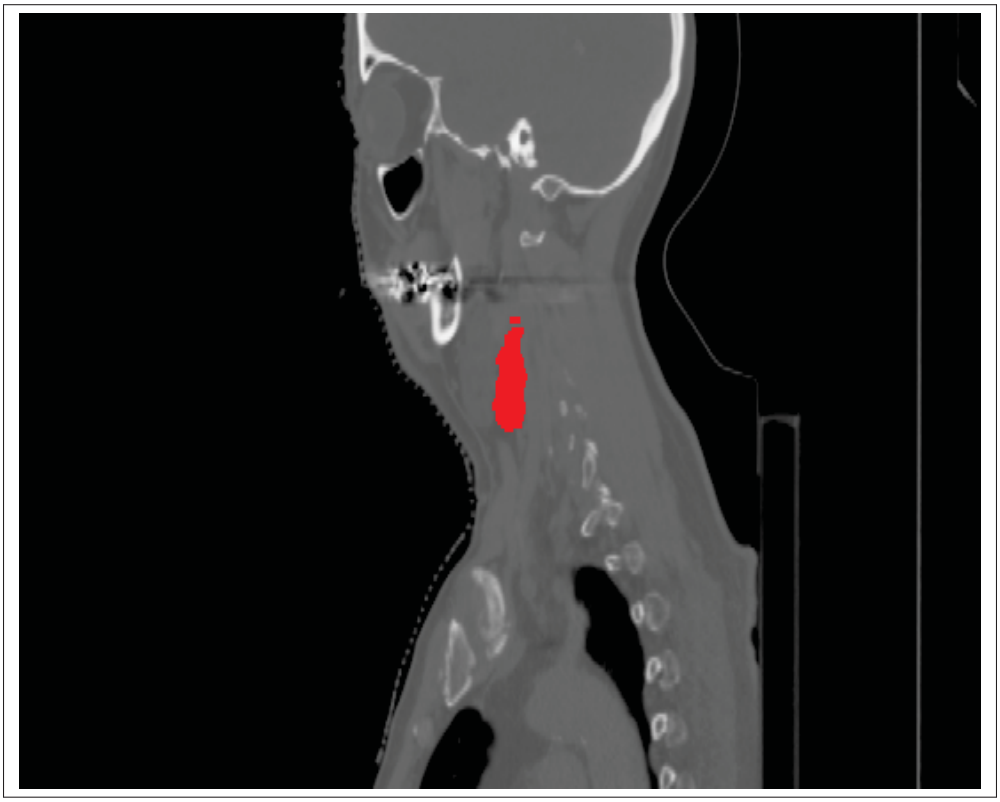


Figure 1.1 Exemple de la segmentation d'une tumeur de la tête et du cou. Les pixels en rouge font partie de la tumeur alors que les autres pixels n'appartiennent pas à cette région

1.2 Les méthodes d'apprentissage : supervisées, faiblement supervisées et non supervisées

Un des facteurs les plus importants qui permet à un modèle machine learning ML de bien performer est le bon choix de la méthode adéquate à exploiter lors de son apprentissage selon le type de données disponibles (annotées, non annotées...), ainsi, trois catégories principales d'apprentissage peuvent être distinguées dans la littérature de la segmentation des images médicales : complètement supervisée, non supervisée et faiblement supervisée.

1.2.1 Complètement supervisées

L'apprentissage totalement supervisé FSL est très populaire dans la littérature des segmentations des images médicales (Ronneberger, Fischer & Brox (2015) ; Andrearczyk *et al.* (2020b) ; Iantsen, Visvikis & Hatt (2021)). En suivant ce paradigme d'apprentissage, un modèle ML de paramètres θ est entrainé en utilisant un ensemble d'images I_m entièrement annotées. De cette façon, pour chaque pixel i appartenant à une image source Ω_s, notre modèle $f(\theta)$ connaît déjà le vrai vecteur de segmentation (ground-truth) $\mathcal{Y}_i = \{y^{(1)}(i), ..., y^{(K)}(i)\} \in \{0, 1\}^K$ de ce pixel. L'entrainment consiste ainsi à minimiser une fonction d'erreur supervisée $\mathcal{L}_s$ qui relie ses predictions pour ce pixel $\hat{\mathcal{Y}}(i, \theta) = \{\hat{y}^{(1)}(i, \theta), ..., \hat{y}^{(K)}(i, \theta)\} \in [0, 1]^K$ et le vecteur ground-truth selon l'équation suivante :

$$\min_{\theta} \sum_{i \in \Omega_s} \mathcal{L}_s(\mathcal{Y}_i, \hat{\mathcal{Y}}(i, \theta)) \tag{1.1}$$

1.2.2 Non supervisées

L'apprentissage non supervisé UL est le cas où un modèle ML apprend en s'appuyant sur des images entièrement non annotées. Les algorithmes adoptant ce paradigme d'apprentissage sont essentiellement utilisés pour analyser les données, les regrouper et extraire des motifs qui peuvent être utile à mieux les comprendre.

Dans le contexte de la segmentation des images médicales, l'apprentissage non supervisé a connu une croissance significative dans les dernières années menant à plusieurs techniques qui ont été appliquées avec succès pour extraire des structures anatomiques précises, identifier des pathologies et segmenter des régions d'intérêt dans des images médicales. Parmi ces techniques, on peut citer les réseaux adverses génératifs GAN souvent utilisés pour générer des nouvelles données comme dans le travail de (Sivanesan, Braga, Sonnadara & Dhindsa (2019)) qui ont intègrè un réseau génératif GAN capable de synthétiser des images médicales annotées. Celles-ci sont utilisées par la suite pour entrainer des réseaux de neurones supervisés pour faire la segmentation.

1.2.3 Faiblement supervisées

L'apprentissage faiblement supervisé WSL représente le cas où la supervision du modèle ML provient d'un ensemble de données annotées de façon imprécise et/ou incomplète qui aide ce dernier à construire des connaissances sur les réponses attendues mais pas les connaître vraiment. Pour la segmentation médicale, cette annotation peut être sous la forme des traits et de points comme montre la figure 1.2, des boîtes englobantes ou d'autres formes... Parmi les techniques qui utilisent l'apprentissage non supervisé on peut citer celle où l'entrainement d'un modèle ML se fait en optimisant une fonction d'erreur composée de deux parties. Une première partie $\mathcal{L}_s$ qui s'occupe de la fiable annotation disponible et une deuxième partie non supervisée $\mathcal{L}_{us}$ qui joue le rôle d'un régularisateur. Dans ce cas, les deux parties sont généralement reliées par un paramètre de pondération λ de la manière suivante :

$$\mathcal{L} = \mathcal{L}_s + \lambda \cdot \mathcal{L}_{us} \tag{1.2}$$

Figure 1.2 Exemple d'une faible annotation, sous la forme de traits et de points, pour une tâche de segmentation des organes abdominaux
Tirée de (Luo *et al.* (2022))

.

1.2.4 Résumé des méthodes d'apprentissage

En résumé, trois catégories de méthodes d'apprentissage se présentent dans la littérature de la segmentation des images médicales : soit totalement supervisé, non supervisé et faiblement supervisé. Un choix adéquat de cette méthode selon les données disponibles pour faire apprendre le modèle ML peut facilement augmenter sa performance, cependant cette performance peut

diminuer significativement pendant la phase d'inférence notamment à la présence de nouvelles distributions de données non vues avant. On parle ainsi du problème de généralisation.

1.3 Le problème de généralisation dans la littérature

Le problème de généralisation représente aujourd'hui une enquête très populaire dans le domaine des images médicales (Cheplygina, de Bruijne & Pluim (2019)) qui aborde la capacité d'un modèle ML à éteindre son savoir à des nouvelles distributions de données différentes de celles qu'il a utilisées pour apprendre, phénomène souvent connu par Data Shift dans le contexte de l'apprentissage machine. Ainsi les deux questions qui se posent : Premièrement, qu'est-ce qu'on veut dire exactement par le terme *Data Shift* ? Deuxièment, Quelles sont les techniques proposées dans la littérature pour s'attaquer à ce problème ?

1.3.1 Data Shift

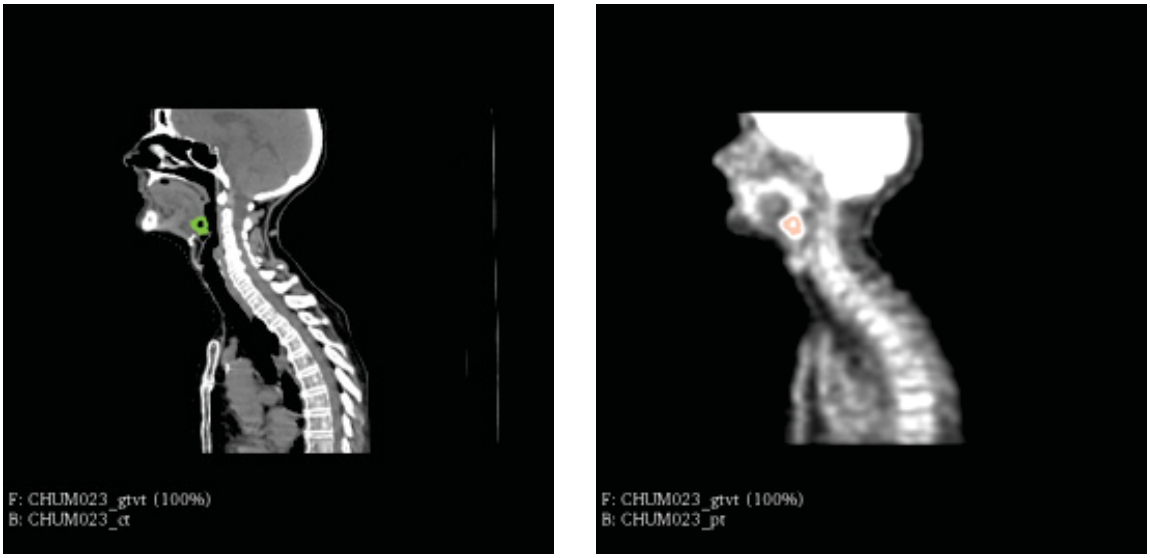
Une distribution de données fait référence à la façon dont ces données sont réparties dans un ensemble et elle peut être décrite en matière de plusieurs mesures notamment la moyenne, la médiane, la variance, l'écart-type ou d'autres termes statistiques qui aident à mieux comprendre les propriétés de ces données et les analyser. Par la suite, un changement dans la distribution de données ou Data Shift peut être défini comme étant toute différence de propriété statistique qui se manifeste entre les données d'apprentissage et celles de test et qui peut influencer la performance du modèle entrainé lors de la phase d'inférence. Dans ce cadre, (Storkey *et al.* (2009)) a distingué six différentes catégories de changements de distributions de données possibles qui sont :

- ***Covariate Shift*** ou le changement de la covariable qui se produit lorsque seulement la répartition des données d'entrainement diffère de celle des données d'inférence $\mathcal{P}(\mathcal{X}_s) \neq \mathcal{P}(\mathcal{X}_t)$ et que la relation entre ces données et leurs étiquettes reste inchangée entre les deux phases $\mathcal{P}(\mathcal{Y}_s|\mathcal{X}_s) = \mathcal{P}(\mathcal{Y}_t|\mathcal{X}_t)$.
- ***Prior Shift*** ou le changement de la distribution des étiquettes des données entre la phase d'apprentissage et la phase d'inférence $\mathcal{P}(\mathcal{Y}_s) \neq \mathcal{P}(\mathcal{Y}_t)$ qui peut se causer suite à un

changement de la méthode d'échantillonnage tout en gardant la distribution des données inchangée $\mathcal{P}(\mathcal{X}_s) = \mathcal{P}(\mathcal{X}_t)$.

- ***Imbalanced Data*** qui s'occurre suite à une répartition inégale des classes dans les données d'entrainement qui entraîne un changement potentiel entre la distribution des étiquettes de la base source et celle des étiquettes de la base cible sous la forme de $\mathcal{P}(\mathcal{Y}_s) \neq \mathcal{P}(\mathcal{Y}_t)$. Ainsi on se trouve avec des événements rares avec lesquels le modèle ML performe d'une façon mediocre.
- ***Sample Selection Biais*** qui se manifeste quand la sélection des données d'entraînement soit biaisée par certains facteurs et par la suite ne soit pas totalement représentative des échantillons de test. Cette sélection biaisée peut entraîner un changement dans les distributions de données $\mathcal{P}(\mathcal{X}_s) \neq \mathcal{P}(\mathcal{X}_t)$ ainsi que les distributions de leurs étiquettes $\mathcal{P}(\mathcal{Y}_s) \neq \mathcal{P}(\mathcal{Y}_t)$.
- ***Source Component Shift*** qui se produit lorsque les données proviennent de plusieurs sources différentes et que la participation de chaque source dans cette composition peut varier en passant de l'étape d'entraînement à l'étape de test, provoquant ainsi une variation potentielle de la distribution des données $\mathcal{P}(\mathcal{X})$ et/ou leurs étiquettes $\mathcal{P}(\mathcal{Y})$.
- ***Domain Shift*** qui se caractérise généralement par un changement au niveau des mesures utilisées pour décrire les données, ainsi au lieu d'observer $\mathcal{X}$ on observe une transformation $\mathcal{F}(\mathcal{X})$ de celle-ci. Dans ce cadre, la relation entre les données et leurs étiquettes $\mathcal{P}(\mathcal{Y}|\mathcal{F}(\mathcal{X}))$ ne changent pas en passant de la base source à la base du test, mais elle dépend de la transformation $\mathcal{F}$ appliquée aux données initiales.

En relation avec les images médicales, le changement des distributions de données a été souvent reconnu comme un phénomène très fréquent qui se produit comme résultat de l'hétérogénéité des patients, la diversité des modalités d'images, la variété des machines employées dans les sites cliniques et surtout la différence des protocoles utilisés pour prendre ces images. Un exemple de ça est illustré par la figure 1.3 où le changement de distribution de données se manifeste entre deux modalités différentes pour la même partie du corps d'un même patient.



a) CT-Scan b) PET-Scan

Figure 1.3 Exemple d'un changement de la distribution de données entre une PET-Scan et une CT-Scan de la tête et du cou d'un même patient
Tirée de (Andrearczyk *et al.* (2022))

1.3.2 Les solutions proposées dans la littérature pour résoudre le problème de généralisation

Très récemment, le domaine de l'apprentissage machine s'est distinguée par une explosion exceptionnelle avec l'apparition de plusieurs modèles énormes qui possèdent des centaines de millions de paramètres à entraîner dont le plus fameux l'agent ChatGPT lancé par OpenAI et Microsoft à la fin de 2022 et qui se base sur le modèle de langage GPT-3 (Generative Pre-trained transformer) composé de 175 milliards de paramètres (Brown *et al.* (2020)).

Selon (Boudiaf, Mueller, Ayed & Bertinetto (2022)) basé sur l'étude (Patterson *et al.* (2021)), l'entraînement d'un modèle GPT-3 à partir de zéro émet 552 tonnes de CO_2 qui représente un équivalent de 6 vols de New York vers San Francisco, et avec la migration mondiale vers l'économie verte qui se base sur l'adaptation de nouveaux procédés moins polluants, il y a une tendance de la communauté scientifique vers les sujets de recherches portant sur l'adaptation des

modèles ML déjà entrainés vers d'autres scénarios. Ainsi, l'apprentissage à grande échelle se fait une seule fois. Par la suite, on consomme moins d'énergie, sans considérer le temps gagné !

Dans ce cadre, on va présenter, dans la suite, les différentes solutions proposées dans la littérature pour adapter les modèles ML déjà entrainés pour des nouvelles distributions de données en indiquant la différence majeure entre ces solutions.

1.3.2.1 Fine Tuning

Le Fine Tuning (FT) est une procédure d'apprentissage couramment utilisée pour ajuster des modèles ML pré-entraînés sur des grandes bases de données annotées, comme ImageNet, pour d'autres tâches plus spécifiques, comme la classification des voitures, en utilisant un ensemble de données annoté plus propre à cette tâche. Ainsi, on commence par prendre un modèle ML déjà entraîné sur une banque de données source et on l'adapte à une nouvelle tâche plus particulière en le réentraînant sur une base de données spécifique à cette tâche appelée base cible. Cette adaptation peut toucher tous les paramètres du réseau comme on peut conserver les paramètres qui apprennent des caractéristiques de bas niveau inchangés. Ainsi on garde ces paramètres et on ajuste seulement ceux qui apprennent des caractéristiques de haut niveau qui sont plus sensibles à la tâche considérée (Yosinski *et al.* (2014)).

L'avantage de cette procédure est qu'elle permet d'achever de bonnes performances avec moins d'entraînement et moins de données, cependant elle nécessite une base d'entrainement qui appartient au domaine cible. Ceci limite son application, notamment dans le contexte du FSL puisque dans la plupart des cas ce n'est pas faisable d'avoir une base cible annotée pour réentraîner le modèle.

1.3.2.2 L'adaptation du domaine

Pour faire face au manque de données annotées au niveau du domaine cible qui empêche l'application du FT dans plusieurs scénarios, une autre approche appelée l'adaptation du domaine DA a été souvent appliqué pour ajuster les modèles ML déjà entrainés pour un autre

domaine cible sans besoin d'annotations. Au cours de cet ajustement, cette procédure réentraîne le modèle considéré en s'appuyant sur une combinaison de fonctions de perte qui regroupe une supervisée qui s'applique aux données sources $\mathcal{X}_s$ et leurs étiquettes $\mathcal{Y}_s$ et une non supervisée qui relie les données sources $\mathcal{X}_s$ et les données cibles $\mathcal{X}_t$ afin de minimiser l'écart entre les deux domaines. Ainsi la forme générale de cette combinaison sera comme suit :

$$\mathcal{L} = \mathcal{L}_s + \lambda \cdot \mathcal{L}_{us} = \mathcal{L}(\mathcal{X}_s, \mathcal{Y}_s) + \lambda \cdot \mathcal{L}(\mathcal{X}_s, \mathcal{X}_t) \tag{1.3}$$

Parmi les travaux qui appliquent le DA on cite celui de (Zhu *et al.* (2017)) qui utilise un réseau GAN pour transférer une image d'un domaine source $\mathcal{D}_s$ à un domaine cible $\mathcal{D}_t$, de cette façon un modèle ML peut étendre son savoir à des nouvelles distributions de données non vues lors de son apprentissage.

Quoique la procédure du DA n'ait pas besoin d'une annotation de la base cible pour faire l'adaptation du modèle, elle reste en besoin d'un accès à la base source qui ne peut pas être toujours faisable vu plusieurs contraintes comme celles reliées à la confidentialité des données.

1.3.2.3 Source Free Domain Adaptaion

Pour mitiger le problème de l'accessibilité à la base source pendant la phase de l'adaptation, certains travaux ont proposé la procédure Source Free Domain Adaptaion (SFDA) qui relaxe cette nécessité et adapte le modèle ML sans besoin d'avoir accès aux données source. Un exemple de ces travaux est celui de (Bateson *et al.* (2022a)) qui s'est basé uniquement sur les données cibles non annotées et des priori invariants par domaine pour faire le réentrainement d'un modèle ML. Cette méthode est appliquée à la segmentation d'organes au niveau des images médicales lors de l'inférence. Dans ce contexte, ces priori invariants par domaine ont été choisis pour représenter le ratio de chaque classe k dans l'image Ω_t du domaine cible $\mathcal{D}_t$ donnée par l'équation suivante :

$$\tau(k) = \frac{1}{|\Omega_t|} \sum_{i \in \Omega_t} y_i^k \tag{1.4}$$

Le problème avec l'équation 1.4 est qu'elle nécessite que l'image Ω_t soit totalement annotée pour avoir le vrai ratio $\tau(k)$ de chaque classe k. Ainsi pour faire face à l'indispensabilité de cette annotation l'auteur a remplacé $\tau(k)$ par une estimation $\tau_e(k)$ qui peut être obtenue à l'aide des priori sur le domaine cible $\mathcal{D}_t$. Ces estimations sont ensuite utilisées pendant la phase d'adaptation pour minimiser la divergence KL qui les relie avec les prédictions du modèle pour les ratios des classes $\hat{\tau}(\theta, k)$ selon l'équation suivante :

$$\min_{\theta} KL(\tau_e(k), \hat{\tau}(\theta, k)) \tag{1.5}$$

avec $\hat{\tau}(\theta, k)$ est donnée par :

$$\hat{\tau}(\theta, k) = \frac{1}{|\Omega_t|} \sum_{i \in \Omega_t} \hat{y}^{(k)}(i, \theta) \tag{1.6}$$

Bien que cette solution soulage le besoin d'accéder à la base de données d'origine, elle reste en besoin d'avoir des connaissances sur le domaine cible et de réaliser un réentrainement du modèle ce qui limite son utilisation dans des scénarios où on n'a pas des connaissances suffisantes sur le domaine cible ou on n'a pas suffisamment de temps pour faire un réentrainement comme pour les applications en temps réel.

1.3.2.4 Test Time Training

Contrairement aux autres procédures citées précédemment, la procédure Test Time Training (TTT) (Sun *et al.* (2020)) étend le processus d'ajustement de la phase d'entrainement à la phase d'inférence en adaptant le modèle ML à chaque donnée cible au moment du test avant de faire la prédiction pour cette même donnée. Pour atteindre cet objectif, les fondateurs de cette technique

ont recours à une architecture sous la forme de Y sous laquelle le modèle est formé de deux branches différentes qui partagent ceratins nombres de couches aux bas niveaux du modèle : une première branche principale spécifique à la tâche initiale du problème, c'est à dire une tâche de classification, et qui tente d'optimiser l'équation suivante :

$$\min_{\theta_p,\theta_s} \mathcal{L}_s(\mathcal{X}_s, \mathcal{Y}_s) \tag{1.7}$$

et une deuxième branche auxiliaire qui sert à résoudre un problème d'auto-apprentissage sous la forme :

$$\min_{\theta_p,\theta_{us}} \mathcal{L}_{us}(\mathcal{X}_s) \tag{1.8}$$

avec θ_p sont les paramètres du modèle partagées par les deux branches, θ_s les paramètres du modèle spécifique à la branche principale et θ_{us} les paramètres du modèle spécifique à la branche auxiliaire.

Comme indiqué au début de cette section, l'élément particulier de cette méthode est que l'adaptaion du modèle se fait sur deux étapes seprarées : une prémière étape qui se fait avec les données d'entrainement du domain source en utilisant le mode d'apprentissage multi-tâches pour combiner les deux équations 1.7 et 1.8 puis une deuxieme qui se fait pendant la phase d'infèrence au niveau de laquelle on adapte uniquement les paramètres partagées par les deux branches aux données cibles $\mathcal{X}_t$ selon l'équation suivante :

$$\min_{\theta_p} \mathcal{L}_{us}(\mathcal{X}_t) \tag{1.9}$$

Donc, si on admet que ${\theta_p}^*$ est la valeur optimale de l'équation 1.9, la prédiction finale du modèle pour une donnée test x_t va être $\hat{y}_t = f(x_t, \theta)$ avec $\theta = ({\theta_p}^*, \theta_s)$.

Quoique que cette méthode essaye de faire une adaptation spécifique du modèle $f(\theta)$ à chaque nouvelle donnée de test pendant la phase d'inférence, elle a besoin d'accès aux données du domaine source et de réentrainer le modèle avec ces données pour être compatible avec l'architecture Y déjà proposée, un scénario qui ne peut pas être faisable dans la majorité des cas.

1.3.2.5 Fully Test Time Adaptation

La procédure Fully Test Time Adaptation se caractérise par relaxer le besoin de faire l'adaptation du modèle pendant la phase d'entrainement et par la suite n'a pas besoin de données sources $\mathcal{X}_s$ ni à des connaissances sur ces données. Ainsi nous n'avons besoin que des données cibles $\mathcal{X}_t$ pour faire l'adaptation du modèle ML pendant la phase d'inférence avant de faire la prédiction. Cette adaptation peut toucher tous les paramètres du modèle ou impliquer uniquement certains paramètres en conservant les autres intouchées par optimiser une fonction de perte non supervisée. Dans ce contexte, (Wang *et al.* (2020)) ont optimisé l'entropie de *Shannon* 1.10 pour adapter les paramètres des couches de normalisation (BN) d'un modèle ML à chaque nouvelle donnée de test x_t avant de produire la classe finale $\hat{y}_t$ de cette donnée.

$$\mathcal{H}(\hat{y}_t) = -\hat{y}_t \cdot log(\hat{y}_t) \tag{1.10}$$

Le processus général de cette méthode, appelée Tent, est donné par l'algorithme suivant :

Algorithme 1.1 Algorithme de la méthode Tent

Algorithme : Algorithme de la méthode Tent

Input : Test samples $\mathcal{D}_t = \{x_t\}_{t=1}^{M}$, model $f(\theta)$ with Batch Norm parameters $\theta_{BN} \subset \theta$, learning rate $\eta > 0$, number of steps $n > 0$

Output : Predictions $\{\hat{y}_t\}_{t=1}^{M}$

Initialize $\theta_{BN_0} = \theta_{BN}$, $adaptation\_mode = False$; `// initialization`

for *test sample* $x_t \in \mathcal{D}_t$ **do**

  if $adaptation\_mode \neq False$ **then**

    Recover model weights : $\theta_{BN} = \theta_{BN_0}$; `// model recovery`

  end if

  for *step* $s \in n$ **do**

    Predict $\hat{y}_t = f(\theta, x_t)$; `// make prediction`

    Compute entropy $\mathcal{H}_t = \mathcal{H}(\hat{y}_t)$; `// entropy minimization eq 1.10`

    Compute gradient $\nabla_{\theta_{BN}} \mathcal{H}_t$; `// model updating`

    Update $\theta_{BN} = \theta_{BN} - \eta(\nabla_{\theta_{BN}} \mathcal{H}_t)$;

  end for

end for

Le choix de l'entropie de shannon comme fonction de perte pour faire l'adaptation était basé sur le fait qu'une telle optimisation va pousser le modèle à prendre des décisions plus confiantes en diminuant son incertitude sur les prédictions. Cependant, faire cette optimisation avec une seule donnée x_t va aboutir à une solution triviale, ce qui correspond à assigner la prédiction $\hat{y}_t$ à la classe la plus probable. Pour cela les auteurs de la méthode Tent ont choisi de faire l'adaptation en s'appuyant sur des lots.

L'utilisation des lots, malgré qu'elle représente une échappe à la solution triviale du problème d'optimisation 1.10, elle peut facilement influencer les statistiques des couches de normalisation, soit la moyenne $E(.)$ et la deviation standard $Var(.)$, surtout avec la présence de lots hétérogènes

ou de petits lots non représentatifs conduisant ainsi à des statistiques non précises et par suite une dégradation de la performance du modèle (Niu *et al.*, 2023). Dans ce cadre et pour rendre l'adaptation par lot plus robuste, dans ce travail les auteurs ont proposé de filtrer les exemplaires de chaque lot selon leurs valeurs d'entropies et par suite faire l'optimisation uniquement en se basant sur les données qui n'ont pas de très grandes valeurs d'entropies et/ou de gradients qui peuvent pousser le modèle ML à diverger.

Avec l'optimisation de l'entropie souvent utilisée dans le contexte du FTTA, une tendance de combiner cette fonction d'erreur avec d'autres composants afin de rendre l'optimisation plus robuste commence à apparaitre. Dans ce cadre, (Bateson *et al.*, 2022b) ont combiné l'entropie de Shannon avec des contraintes sur les priori de formes en utilisant les moments de forme. Ceci rend l'adaptation d'un modèle construit pour segmenter des images médicales plus performante. Ainsi, les descripteurs de formes choisis sont le ratio de classe $\mathcal{R}$, le centroid C et la distance de la centroid $\mathcal{D}$ définis successivement par les équations 1.11, 1.12 et 1.13.

$$\mathcal{R}(\Omega) = \frac{1}{|\Omega|}\mu_{0,0(\Omega)} \tag{1.11}$$

$$C(\Omega) = (\frac{\mu_{1,0}(\Omega)}{\mu_{0,0}(\Omega)}, \frac{\mu_{0,1}(\Omega)}{\mu_{0,0}(\Omega)}) \tag{1.12}$$

$$\mathcal{D}(\Omega) = (\sqrt[2]{\frac{\overline{\mu}_{2,0}(\Omega)}{\mu_{0,0}(\Omega)}}, \sqrt[2]{\frac{\overline{\mu}_{0,2}(\Omega)}{\mu_{0,0}(\Omega)}}) \tag{1.13}$$

avec $\mu_{p,q}$ est le moment d'image d'ordre $p, q \in \mathbf{N}$ et $\overline{\mu}_{p,q}$ est le moment central d'image d'ordre $p, q \in \mathbf{N}$ donnés successivement par les deux équations 1.14 et 1.15.

$$\mu_{p,q} = \sum_{u}\sum_{v} u^p v^q I(u, v) \tag{1.14}$$

$$\overline{\mu}_{p,q} = \sum_{u} \sum_{v} (u - \overline{v})^p (v - \overline{v})^q I(u, v) \tag{1.15}$$

Dans ce travail, la fonction de perte finale à optimiser lors de la phase d'inférence est donné par l'équation 1.16.

$$\sum_{i \in \Omega_t} \mathcal{H}(\hat{\mathcal{Y}}(i, \theta)) + KL(\mathcal{R}(\Omega_t), \overline{\mathcal{R}}) + \lambda \cdot \mathcal{F}(\mathcal{M}(\Omega_t), \overline{\mathcal{M}}) \tag{1.16}$$

avec $\mathcal{M} \in \{C, \mathcal{D}\}$, $\overline{\mathcal{M}}$, $\overline{R}$ représente les estimations des descripteurs de la forme et $\mathcal{F}$ une fonction de pénalité quadratique.

Pour résumer, on peut dire que la procédure FTTA a permis d'achever des résultats intéressants tout en relâchant plusieurs contraintes comme le besoin des données de la base source et le besoin de réentraîner le modèle ce qu'a augmenté l'intérêt de la communauté scientifique aux travaux exploitant cette procédure pour atteindre plus de performance tout en créant des nouveaux défis et problèmes à résoudre.

1.3.2.6 Comparaison des différentes procédures d'adaptation

Le tableau 1.1 donne une comparaison des différentes procédures d'adaptation citées dans la littérature. Clairement observé à travers ce tableau, ces méthodes peuvent être classées selon trois grands critères. La première en relation avec la phase d'adaptation en répondant à la question, est-ce que notre modèle va être adapté pendant la phase d'entrainement, la phase d'inférence ou pendant les deux phases ? La deuxième touche les données disponibles pour faire cette adaptation (sources, cibles...) et le type de ces données (annotées, non annotées...) et la troisième concerne la nature des fonctions de perte qui vont être optimisées pour faire l'adaptation.

Quoique chacune des procédures précédentes puisse être appliquée sous les conditions qu'elle impose pour donner des bons résultats, cependant la procédure FTTA reste la plus convaincante

à appliquer dans les scénarios réels vu qu'elle relâche beaucoup d'hypothèses tout en achevant des bonnes performances.

Tableau 1.1 Comparaison entre les différentes procédures proposées dans la littérature pour résoudre le problème du changement de la distribution de données entre la base d'entraînement et la base de test

Paramètres	**Phase d'adaptation**		**Données Disponibles**		**La fonction de perte**	
	Entraînement	**Test**	**Source**	**Cible**	**Entraînement**	**Test**
FT	oui	non	-	$\mathcal{X}_t$,$\mathcal{Y}_t$	$\mathcal{L}_s(\mathcal{X}_t, \mathcal{Y}_t)$	-
DA	oui	non	$\mathcal{X}_s$,$\mathcal{Y}_s$	$\mathcal{X}_t$	$\mathcal{L}_s(\mathcal{X}_s, \mathcal{Y}_s) + \mathcal{L}_{us}(\mathcal{X}_s, \mathcal{X}_t)$	-
SFDA	oui	non	$priori$	$\mathcal{X}_t$	$\mathcal{L}_{us}(priori, \mathcal{X}_t)$	-
TTT	oui	oui	$\mathcal{X}_s$,$\mathcal{Y}_s$	$\mathcal{X}_t$	$\mathcal{L}_s(\mathcal{X}_s, Y_s) + \mathcal{L}_{us}(\mathcal{X}_t)$	$\mathcal{L}_{us}(\mathcal{X}_t)$
FTTA	non	oui	-	$\mathcal{X}_t$	-	$\mathcal{L}_{us}(\mathcal{X}_t)$

CHAPITRE 2

MISE EN SITUATION

Ce chapitre est consacré pour définir notre problématique qui se concentre sur l'application du FTTA pour la segmentation des tumeurs de la tête et du cou. Il commence par une motivation sur le problème de la segmentation des tumeurs de la tête et du cou et son état d'art, avant de se clôturer par une présentation des différents défis qu'on peut rencontrer en appliquant la procédure FTTA pour ce problème.

2.1 Motivation

Les tumeurs de la tête et du cou sont des cancers qui peuvent se développer dans les voies aérodigestives supérieures, les glandes salivaires, le nasopharynx ou les sinus et la fosse nasale. Ces cancers comptent pour 4% de tous les cancers dans le monde et ils sont liés à plus que 70% au tabagisme et à la consommation d'alcool (ESMO (2015)). Pour diagnostiquer ces cancers et planifier le traitement, les médecins recourent à plusieurs pratiques cliniques dont l'analyse de la CT-Scan et la PET-Scan sont les plus communes. En effet, la tomodensitométrie permet d'analyser la région suspecte de point de vue anatomique alors que la tomographie par émission de positrons est utile pour faire une analyse métabolique et fonctionnelle de cette région. Ceci aide les oncologues à bien délimiter les tumeurs et les organes en risque en s'appuyant sur la complémentarité d'informations entre les deux modalités.

Cette segmentation manuelle a été souvent connue par les experts comme une tâche difficile, chronophage et qui présente une variabilité entre les observateurs dans la délinéation du volume tumoral brut (Gudi *et al.* (2017)), ce qui peut restreindre la thérapie à subir par le patient. Ces difficultés ont poussé la communauté scientifique à migrer vers l'intelligence artificielle pour faire apprendre la segmentation automatique afin d'aider les radiologues à accomplir leurs tâches. Ainsi la première version de la compétition Hecktor (HEad and neCK TumOR segmentation and outcome prediction in PET/CT images) a été organisée par la MICCAI (Medical Image Computing and Computer Assisted Intervention) en 2020 pour développer des

nouvelles approches pour segmenter les tumeurs de la tête et du cou. Cette compétition qui a été etendue vers deux autres éditions en 2021 et 2022 avec l'ajout d'une deuxième tâche pour prédire la survie sans récidive d'un patient en analysant les deux modalités CT-Scan et PET-Scan par les algorithmes IA.

En revenant à la littérature de la tâche de segmentation, on trouve que le meilleur indice de similarité Sørensen-Dice obtenu lors de la première édition de cette compétitio est 0.7591% (Andrearczyk *et al.* (2021)), le meilleur indice Dice de la deuxième édition est 0.7785% (Andrearczyk *et al.* (2022)) alors que le meilleur indice Dice de la troisième édition est 0.788% (Andrearczyk *et al.* (2023)). À noter que l'indice de Sørensen-Dice est un indicateur de performance statistique souvent utilisé pour évaluer les tâches de segmentation des images médicales et qui permet de mesurer le niveau de similarité entre deux ensemble $\mathcal{A}$ et $\mathcal{B}$ différents selon l'équation suivante :

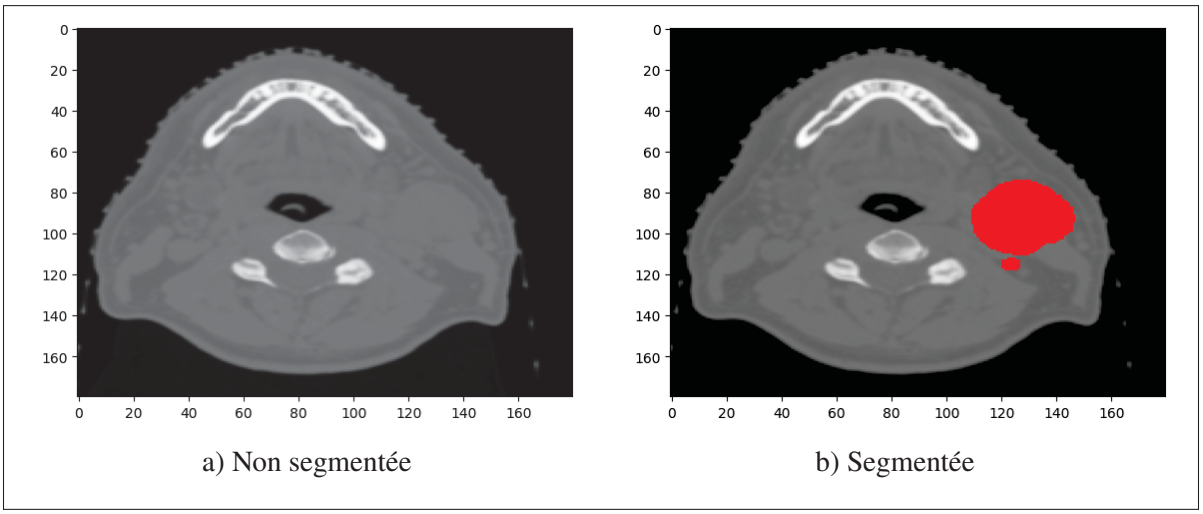
$$DSC(\mathcal{A}, \mathcal{B}) = \frac{2|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}| + |\mathcal{B}|} \tag{2.1}$$

L'augmentation lente de l'indice de similarité DSC en passant d'une édition à une autre de la compétition Hecktor avec l'augmentation de nombre de participants à chaque édition prouve la difficulté de la tâche de segmentation automatique des tumeurs de la tête et du cou à partir des images CT-Scan/PET-Scan. Cette difficulté est due en premier lieu à des limites liées aux deux modalités d'images elles-mêmes et puis à la vérité terrain définie par les spécialistes.

2.1.1 Les limites liées à la modalité CT-Scan

Selon (Li, Zhao, Lu & Tan (2020)), la CT-Scan est l'une des modalités utile pour faire l'analyse anatomique d'une région suspecte permettant de délimiter les tumeurs grâce à sa bonne résolution spatiale. Cependant, la présence d'un arrière-plan complexe avec une similarité d'intensités entre la partie cancéreuse et les tissus mous qui l'entourent rendent cette distinction difficile

même pour un expert. Ceci est clairement observé dans la figure 2.1 où on ne peut pas distinguer entre la tumeur de la tête et du cou et les tissus mous qui l'entourent.

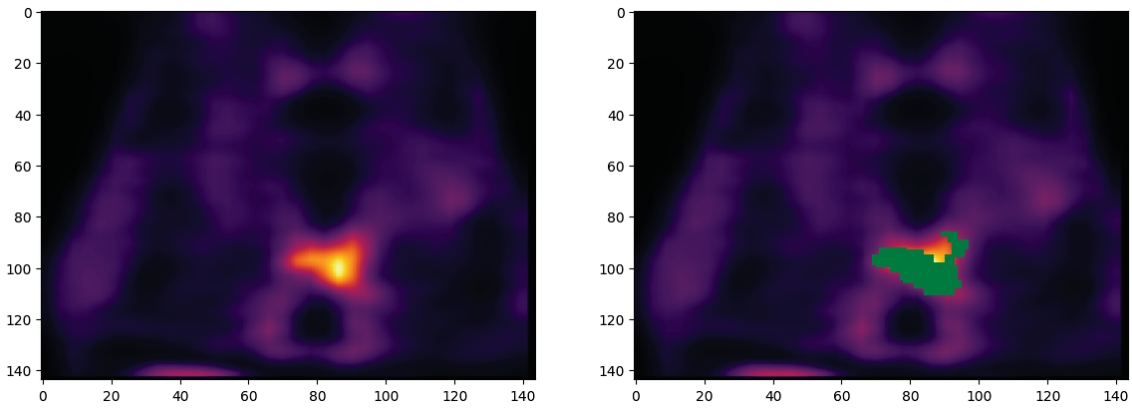


a) Non segmentée b) Segmentée

Figure 2.1 Exemple d'une CT-Scan d'un patient avec une tumeur de la tête et du cou entouré par les tissus mous présents dans cette région

2.1.2 Les limites liées à la modalité PET-Scan

Aussi selon (Li *et al.* (2020)), la PET-Scan est utile pour faire le diagnostic précoce des cancers à travers l'analyse fonctionnelle des tissus. Ainsi les tissus cancéreux vont apparaître sous la forme de régions de chaleur distinctes. Cependant, la résolution spatiale médiocre de cette modalité se dresse comme un obstacle pour définir avec précisément les vraies limites des tumeurs. Ceci est bien clair dans la figure 2.2 où on voit que les pixels du contour de la zone de chaleur ne font pas partie de la partie cancéreuse.



a) Non segmentée b) Segmentée

Figure 2.2 Exemple d'une PET-Scan d'un patient avec une tumeur de la tête et du cou qui se voit comme une zone de chaleur de résolution médiocre
Tirée de (Andrearczyk *et al.* (2022))

2.1.3 Les limites liées à la vérité terrain

La variabilité ente les observateurs peut-être définis comme une mesure qui reflète la différence entre les delineations manuelles des volumes par deux experts différents en procédant la même image. Ce phénomène a été souvent le sujet de plusieurs études cliniques puisqu'il permet de savoir à quel point la segmentation d'une région est précise et en concordance. Une de ces études est celle de (Gudi *et al.* (2017)) qui a étudié la variabilité entre les observateurs dans la délimitation des tumeurs et des organes à risque pour le traitement du carcinome squameux de la tête et du cou en utilisant les deux modalités CT-Scan et PET-Scan. Ce travail a abouti à un indice de similarité DSC égale à 0.69 lors de l'évaluation de cette variabilité. Dans un contexte similaire, les organisateurs de la compétition Hecktor ont obtenu un coefficient de similarité DSC égale à 0.61 lors de leur étude inter-observateurs qui a été effectuée sur une portion de la base de données de la première édition de la compétition (Andrearczyk *et al.* (2022)). Ces deux études mettent encore en évidence la difficulté de la segmentation manuelle des tumeurs de la tête et du cou suite à laquelle la vérité terrain peut ne pas être assez précise et représente ensuite une source d'erreurs pour le modèle à entraîner.

Avec la variabilité entre les observateurs, parmi les problèmes que présente la segmentation manuelle de ces tumeurs, est le coût très élevé de ce processus, quelle que soit en matière de temps ou d'effort. Pour cela l'une des pratiques cliniques les plus utilisées est d'enregistrer la vérité terrain sur une modalité à partir d'une autre déjà segmentée au lieu de refaire tout le processus de nouveau. Comme exemple on enregistre la vérité terrain sur la PET-Scan à partir de celle segmentée sur la CT-Scan ou on enregistre la vérité terrain sur la CT-Scan à partir de celle segmentée sur une autre CT-Scan de planification. Cette pratique peut facilement aboutir à des grandes réalités erronées qui confondent le modèle lors de sa phase d'apprentissage. Ceci est confirmée par (Andrearczyk *et al.* (2020b)) où la vérité terrain d'une modalité PET-Scan a inclus faussement la trachée comme partie de la tumeur. Dans ce même exemple la prédiction automatique a pu corriger cette erreur en distinguant la trachée de la tumeur en se basant sur les connaissances acquis par le modèle entrainé à partir d'autres exemples. Cette observation prouve la capacité de la segmentation automatique d'être même plus précise que la segmentation manuelle une fois le modèle est entraîné correctement. Ainsi la question qui se pose est comment on peut entraîner notre modèle correctement avec ses limites liées à la vérité terrain et les limites citées précédemment en relation avec les modalités ?

2.1.4 La co-ségmentation est la solution

Prenons en considération les limites citées précédemment, plusieurs travaux ont essayé de profiter de la complémentarité entre les deux modalités CT-Scan et PET-San afin de faire apprendre la segmentation automatique. En effet la majorité des travaux qui ont participé à la compétition Hecktor ont suivi cette démarche pour augmenter la performance de leurs modèles comme la méthode gagnante à la première édition (Iantsen *et al.* (2021)). Pour bien l'expliquer, cette technique souvent connue sous le nom de fusion ou de co-segmentation consiste à combiner les informations extraites de plusieurs modalités d'images médicales pour produire une segmentation plus précise et fiable de la structure d'intérêt. Ceci peut être achevé suivant différentes façons comme la combinaison des modalités à l'entrée du modèle ou la combinaison des prédictions du modèle pour chaque modalité afin de produire des prédictions plus raffinées ou d'autres façons

de combinaison. À noter que cette combinaison peut être pondérée par des poids pour donner plus d'importance à des modalités spécifiques par rapport à d'autres.

Par retour à la littérature de l'utilisation des techniques de fusion dans la segmentation des images médicales, (Jin *et al.* (2019)) ont montré que leur réseau de neurones basé sur deux fusions à la fois des deux modalités CT-Scan/Pet-Scan est plus performant que celui basé uniquement sur une seule fusion, et les deux sont plus performants qu'un troisième qui n'utilise que la modalité CT-Scan pour segmenter les cancers oesophagiens. Dans le même sens, (Kumar, Fulham, Feng & Kim (2019)) ont démontré que l'utilisation d'un composant spécifique qui apprend à quantifier les poids de fusion des caractéristiques extraites à partir de chacune des deux modalités CT-scan et PET-scan peut aboutir à des meilleurs résultats que les méthodes de fusion traditionnelles.

En relation avec la segmentation des tumeurs de la tête et du cou, (Moe *et al.* (2019)) et (Andrearczyk *et al.* (2020b)) ont prouvé que les réseaux convolutifs qui se basent sur les techniques de fusion pour apprendre à faire la segmentation automatique sont plus performants que ceux qui ne se bénéficient que d'une seule modalité, ce qui justifie le recours à la co-segmentation lors de la compétition Hecktor. Cependant, et même avec le recours à la co-segmentation plusieurs modèles trouvent encore des difficultés en phase d'inférence pour distinguer entre les tumeurs et d'autres petits organes comme la langue (Andrearczyk *et al.* (2021)). Ainsi s'éclaire l'idée d'appliquer la procédure FTTA pour la co-segmentation automatique des tumeurs de la tête et du cou afin d'obtenir des meilleures performances.

2.2 Les défis reliés à la procédure FTTA

Comme déjà parlé dans la section 1.3.2.5 les méthodes qui se basent sur la procédure FTTA ont gagné beaucoup d'attention dans les dernières années vu les résultats prometteurs qu'ils achèvent sur plusieurs tâches (Wang *et al.* (2020)), (Bateson *et al.* (2022b)). Cependant, beaucoup de défis sont encore à explorer notamment ceux en relation avec qu'est-ce qu'on adapte et quelle fonction de perte doit-on utiliser pour faire cette adaptation.

Dans la suite, une discussion concernant ces deux questions va être abordée.

2.2.1 Qu'est-ce qu'on adapte

Pour être modifiable par une méthode FTTA, un modèle d'intelligence artificielle $f(\theta)$ doit avoir obligatoirement un ensemble de paramètres entrainables $\theta_e \subset \theta$. Lors du processus de l'adaptation, ces paramètres entrainables vont être ajustés en fonction des données du test $\mathcal{X}_t$ de la manière suivante :

$$\theta'_e \leftarrow \arg\min_{\theta_e} \mathcal{L}(\mathcal{X}_t, \theta) \tag{2.2}$$

avec θ'_e sont les paramètres entrainables après l'adaptation pour les données du test $\mathcal{X}_t$.

Dans le cas d'un réseau neuronal, ces paramètres entrainables représentent un sous ensemble de l'ensemble de ses paramètres. En outre, un réseau neuronal n'est autre qu'un ensemble de couches de neurones artificiels interconnectés entre eux dont chacune traite le lot de données $\mathcal{X}$ qu'elle reçoit en appliquant la transformation suivante :

$$\mathcal{X}' = g(\mathcal{W} \cdot \mathcal{X} + \beta) \tag{2.3}$$

avec $\mathcal{W}$ sont les poids des neurones, β leurs biais et $g(.)$ une fonction d'activation non linéaire qui s'applique à la sortie d'une couche avant qui est ensuite utilisée comme entrée pour la couche suivante du réseau. Ainsi les paramètres entrainables d'un réseau neuronal peuvent être un sous-ensemble de ses poids, un sous-ensemble de ses biais ou un sous-ensemble de ses poids et de ses biais.

Avec les couches de neurones normaux, les couches de normalisation de données inventées par (Ioffe & Szegedy (2015)) sont devenues indispensables dans les réseaux de neurones afin d'accélérer l'apprentissage en normalisant les activations $\mathcal{X}'$ des autres couches de la manière suivante :

$$\alpha \cdot \frac{\mathcal{X}' - E(\mathcal{X}')}{Var(\mathcal{X}')} + \beta \tag{2.4}$$

Comme montre l'équation 2.4, les couches de normalisation possèdent deux types de paramètres, soit les paramètres appris par le réseau α et β, soit les paramètres statistiques $E(.)$ et $Var(.)$. Ces derniers sont calculés à partir du lot actuellement présent au niveau de la couche de normalisation lors de la phase d'entrainment ou déduites à partir de la population d'entrainment lors de la phase d'inférence. Indépendamment de la nature de ces paramètres, ils sont considérés des paramètres entrainables qui peuvent être ensuite ajustés par une méthode FTTA.

En revenant à la littérature, la plupart des travaux récents qui appliquent la procédure FTTA ont choisi d'ajuster uniquement les paramètres des couches de normalisation tout en conservant les poids et les biais des autres couches de neurones intouchables (Wang *et al.*, 2020), (Bateson *et al.*, 2022b), (Niu *et al.*, 2023). L'adaptation dans ces travaux s'articule principalement sur la mise à jour des paramètres α et β pour chaque nouveau lot de test $\mathcal{X}_t = \{x_1, ..., x_n\}$, aussi sur l'utilisation des statistiques de ce lot en inférence au lieu de celles de la population d'entrainment en calculant la moyenne $E(\mathcal{X}_t)$ et l'écart-type $Var(\mathcal{X}_t)$ successivement de la façon suivante :

$$E(\mathcal{X}_t) = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.5}$$

$$Var(\mathcal{X}_t) = \sqrt[2]{\frac{1}{n} \sum_{i=1}^{n} (x_i - E(\mathcal{X}_t))^2} \tag{2.6}$$

Le choix d'ajuster uniquement les paramètres des couches de normalisation dans ces travaux se justifie par le fait que comme ça le processus d'adaptation soit plus efficace en terme du temps et des ressources de calcul nécessaires, puis par les bons résultats achevés sur plusieurs scénarios.

2.2.2 Quelle fonction de perte doit-on utiliser pour faire l'adaptation

Dans la littérature plusieurs fonctions de pertes ont été utilisées en FTTA dont la plus commune est l'entropie de Shannon qui peut être optimisée toute seule (Wang *et al.*, 2020) ou en combinaison avec d'autres termes (Bateson *et al.*, 2022b) comme déjà expliqué dans la partie 1.3.2.5.

Dans le but d'étudier l'effet de la fonction de perte sur l'adaptation du modèle en inférence (Goyal, Sun, Raghunathan & Kolter, 2022) ont effectué une étude où ils ont posé la question suivante : Qu'est-ce que mets une fonction de perte donnée comme le bon choix pour faire l'adaptation pendant la phase d'inférence ? Dans ce travail, les auteurs ont employé le meta-apprentissage pour répondre à cette question. Cette méthode permet à un réseau de neurones qui peut représenter des fonctions de pertes complexes à déterminer les paramètres optimaux de la fonction de perte à utiliser en FTTA à partir des predictions du modèle en cours et les ground-truth qu'elles correspondent. Ainsi si on considère $\mathcal{X}_t$ le lot de test dont on essaie de lui adapter le modèle $f(\theta)$ en utilisant la fonction de loss $\mathcal{L}_{us}$ de paramètres ϕ, $\mathcal{Y}_t$ la ground-truth correspondante à $\mathcal{X}_t$ et $\mathcal{L}_s$ la fonction de perte supervisée utilisée par le réseau de neurones qui va ajuster $\mathcal{L}_{us}$, le méta-apprentissage en FTTA se fait de la façon suivante :

$$\theta' \leftarrow \theta - \sigma \frac{\partial \mathcal{L}_{us}(\phi, f_\theta(\mathcal{X}_t))}{\partial \theta}, \quad \phi' \leftarrow \phi - \delta \frac{\partial \mathcal{L}_s(f_{\theta'}(\mathcal{X}_t), \mathcal{Y}_t)}{\partial \phi} \tag{2.7}$$

La conclusion à laquelle aboutit cette étude est que la meilleure fonction de perte à utiliser en FTTA dépend de la fonction d'erreur employée lors de l'apprentissage du modèle à optimiser. Par exemple l'optimisation de l'équation 2.7 pour un modèle de classification entrainé avec l'entropie croisée donne une fonction de perte optimale à utiliser en FTTA qui mimique l'entropie de Shannon. De la même façon l'optimisation de la même fonction 2.7 en gardant tous les paramètres et en variant uniquement la fonction d'erreur de l'entrainement pour être la fonction de perte quadratique donne une fonction de perte optimale à optimiser en FTTA semblable à une fonction de perte quadratique négative.

2.3 Définition du problème

Pour conclure ce chapitre, on peut dire que le problème de la segmentation des tumeurs de la tête et du cou n'a pas encore atteint un niveau qui lui permet d'être considèrè comme un problème totalement résolu avec la présence de plusieurs challenges encore à explorer. Pour cette raison, et avec la tendance aujourd'hui à utiliser les méthodes FTTA dans l'espoir de rendre les modèles ML plus performants, et avec les différents défis qui présentent ces méthodes, on vise dans ce travail de mémoire à étudier l'effet du FTTA sur la segmentation automatique des tumeurs de la tête et du cou.

CHAPITRE 3

FTTA FOR HEAD AND NECK TUMORS SEGMENTATION

Dans ce chapitre, on va discuter l'application du FTTA pour les modèles ML entrainés pour faire la segmentation automatique des tumeurs de la tête et du cou. On va commencer avec la démarche suivie pour atteindre cet objectif avant d'exposer la base des données utilisée, les détails de l'implementation, les résultats obtenus et terminer par une étude comparative.

3.1 FTTA for Head and Neck Tumors Segmentation

Pour appliquer le FTTA à la segmentation automatique des tumeurs de la tête et du cou, on a inspiré de la méthode Tent (Wang *et al.*, 2020) décrite dans la section 1.3.2.5.

3.1.1 Phase de l'apprentissage

Pendant la phase de l'apprentissage, un modèle $f(\theta)$ est entraîné en utilisant un ensemble d'images sources $\mathcal{I}_m : \Omega_s \subset \mathcal{R}^\ni \rightarrow \mathcal{R}$ dans l'objectif de minimiser l'entropie croisée (CE) par rapport à ses paramètres θ selon l'équation suivante :

$$\min_{\theta} - \sum_{\mathcal{I}_m} \frac{1}{|\Omega_s|} \mathcal{Y}_i \log(\hat{\mathcal{Y}}(i, \theta)) \tag{3.1}$$

3.1.2 Phase de l'adaptation, l'application du FTTA

Dans la phase d'adaptation, on a choisi d'ajuster les paramètres θ_{BN} des couches de normalisation BN du modèle entrainé $f(\theta)$ de façon spécifique à chaque nouvelle image cible $\Omega_t : \mathcal{R}^3 \rightarrow \mathcal{R}$ en minimisant l'entropie de Shannon selon l'algorithme suivant :

Algorithme 3.1 Algorithme de la méthode Tent appliqué pour la segmentation des tumeurs de la tête et du cou, les parties colorées indiquent les différences entre la méthode originale et la méthode appliquée dans notre cas

Algorithme : Algorithme de la méthode Tent appliqué pour la segmentation des tumeurs de la tête et du cou.

Input : Single test Sample Ω_t, model $f(\theta)$ with Batch Norm parameters $\theta_{BN} \subset \theta$, learning rate $\eta > 0$, number of steps $n = 1$

Output : Predictions $\hat{\mathcal{Y}}(i,\theta) = (\hat{y}^{(1)}(i,\theta), ..., \hat{y}^{(K)}(i,\theta)) \in [0,1]^K$ for $i \in \Omega_t$

// Original method

Initialize $\theta_{BN_0} = \theta_{BN}$, $adaptation\_mode = False$; // initialization

for *test sample* $x_t \in \mathcal{D}_t$ **do**

  if $adaptation\_mode \neq False$ **then**

    Recover model weights : $\theta_{BN} = \theta_{BN_0}$; // model recovery

  end if

  for *step* $s \in n$ **do**

    Predict $\hat{y}_t = f(\theta, x_t)$; // make prediction

    Compute entropy $\mathcal{H}_t = \mathcal{H}(\hat{y}_t)$; // entropy minimization eq 1.10

    Compute gradient $\nabla_{\theta_{BN}} \mathcal{H}_t$; // model updating

    Update $\theta_{BN} = \theta_{BN} - \eta(\nabla_{\theta_{BN}} \mathcal{H}_t)$;

  end for

end for

// Applied method

Predict $\hat{\mathcal{Y}}_t = f(\theta, \Omega_t)$; // make prediction

Compute entropy $\mathcal{H}_t = \mathcal{H}(\hat{\mathcal{Y}}_t)$; // entropy minimization eq 1.10

Compute gradient $\nabla_{\theta_{BN}} \mathcal{H}_t$; // model updating

Update $\theta_{BN} = \theta_{BN} - \eta(\nabla_{\theta_{BN}} \mathcal{H}_t)$;

Re-predict $\hat{\mathcal{Y}}_t = f(\theta, \Omega_t)$; // make final prediction

Comme le montre l'algorithme, la configuration de la phase d'adaptation est choisie de façon à limiter la minimisation de l'entropie à une seule étape pour chaque nouveau sujet. Ainsi l'ajustement se fait d'une manière discontinue en considérant toutes les données du domaine cible. Cette configuration, nommée FTTA-Dis1 semble d'être plus réaliste puisque, d'une part, elle n'a pas besoin d'avoir accès à un ensemble de données du domaine cible afin de faire l'adaptation et d'autre part elle consomme moins du temps et moins de ressources computationnelles ce qui facilite son application dans le domaine médical.

3.2 Base de données

Dans ce travail, on a utilisé la base de données de la deuxième édition 2021 de la compétition Hecktor (Andrearczyk *et al.* (2022)). Cette base de données se compose de 224 sujets collectés de cinq centres différents. Chacun de ces sujets est formé d'une CT-Scan 3D et d'une Pet-Scan 3D qui sont totalement annotés de façon avec laquelle on se trouve avec deux différentes classes : une classe tumeur et une classe arrière-plan. En plus de ça, chaque sujet comporte les coordonnées d'une boîte englobante de taille $144 \times 144 \times 144$ qui indique la localisation de la partie de l'oropharynx et qui contient la tumeur avec un pourcentage supérieur à 0.9 (Andrearczyk, Oreiller & Depeursinge (2020a)). Le tableau 3.1 expose la liste des centres dont ont été collectés les données avec le nombre de sujets par centre et les scanneurs utilisés pour les collecter.

Tableau 3.1 La liste des centres dont été collectés les données la base de données de la deuxième édition 2021 de la compétition Hecktor et les machines scanneurs utilisées lors de la collection des données
Adapté de (Andrearczyk *et al.* (2022))

Nom du Centre	**Nombre de Sujets**	**Scanneur**
CHGJ : Hôpital Général Juif, Montréal, CA	55	Scanner (Discovery ST, GE Healthcare)
CHUS : Centre Hospitalier Universitaire de Sherbooke, Sherbrooke, CA	72	Scanner (Gemini GXL 16, Philips)
CHMR : Hôpital Maisonneuve-Rosemont, Montréal, CA	18	Scanner (scanner (Discovery STE, GE Healthcare)
CHUM : Centre Hospitalier de l'Université de Montréal, Montréal, CA	56	Scanner (Discovery STE, GE Healthcare)
CHUP : Centre Hospitalier Universitaire Poitiers, FR	23	Scanner (Biograph mCT 40 ToF, Siemens)

En observant cette base de données, on remarque qu'elle est fortement applicable à une étude qui explore l'effet d'un changement de la distribution de données entre la phase d'apprentissage et la phase d'inférence d'un modèle ML. En effet, les sujets de cette base viennent de cinq centres différents qui utilisent des différentes machines de Scanneurs et potentiellement différents protocoles. Dans ce contexte, on a décomposé notre étude pour qu'elle comporte cinq études de cas. Ainsi à chaque fois quatre centres soient utilisés pour faire l'apprentissage du modèle ML alors que le cinquième centre soit gardé pour le FTTA comme le montre le tableau 3.2. Comme cela, pour la décomposition CHUP-TTA par exemple, les données des quatres centres CHGJ, CHUS, CHMR, CHUM sont utilisées pour faire l'apprentissage du modèle alors que les données du centre CHUP sont laissées pour la phase d'inférence.

Tableau 3.2 La décomposition des données des cinq centres pour la phase d'apprentissage et la phase d'inférence

Paramètres	**Phase d'apprentissage**	**Phase d'inférence**
Décomp N°1 : CHUP-TTA	CHGJ, CHUS, CHMR, CHUM	CHUP
Décomp N°2 : CHUM-TTA	CHGJ, CHUS, CHMR, CHUP	CHUM
Décomp N°3 : CHMR-TTA	CHGJ, CHUS, CHUM, CHUP	CHMR
Décomp N°4 : CHUS-TTA	CHGJ, CHMR, CHUM, CHUP	CHUS
Décomp N°5 : CHGJ-TTA	CHUS, CHMR, CHUM, CHUP	CHGJ

3.3 Le réseau de neurones

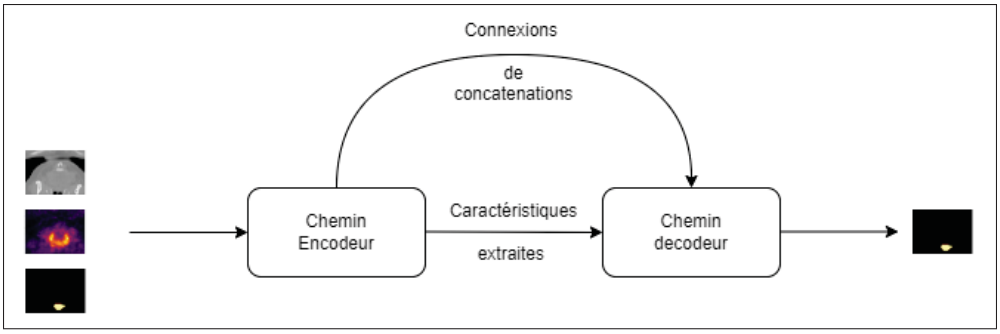


Figure 3.1 L'architecture UNET

Dans ce travail on a utilisé la fameuse architecture UNET (Ronneberger *et al.* (2015)) pour faire l'apprentissage automatique de la segmentation des tumeurs de la tête et du cou. Comme le montre la figure 3.1, cette architecture est sous la forme d'un chemin encodeur formé d'une succession de blocs contracteurs reliés par un chemin décodeur formé d'une succession de blocs extracteurs. Les deux chemins sont reliés avec des connexions de concatenations d'où vient la popularité de cette architecture. En effet, ces connexions permettent de ramener les détails fins

utiles pour faire la prédiction au niveau des pixels à la fin du chemin décodeur et qui peuvent être perdus lors du sous-échantillonnage au côté de l'encodeur.

Les figures 3.2 et 3.3 illustrent successivement un exemple d'un bloc contracteur et un bloc extracteur de l'architecture UNET.

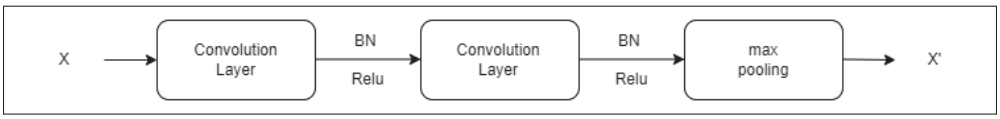


Figure 3.2 Bloc contracteur

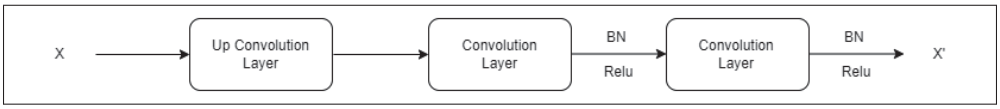


Figure 3.3 Bloc extracteur

3.4 Expérimentations et Résultats

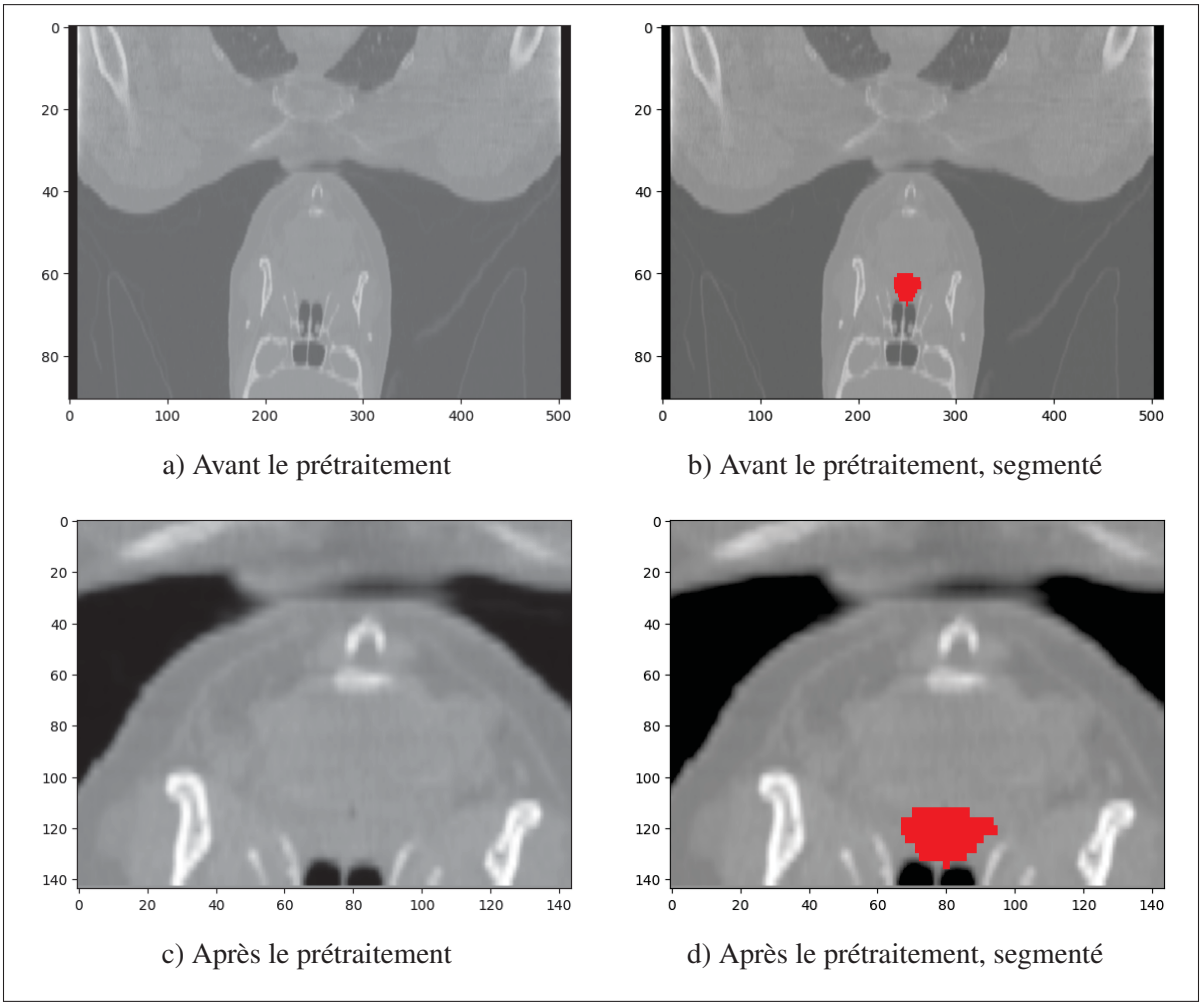
Dans cette partie on va exposer la procédure du prétraitement des données, les paramètres utilisés lors de l'apprentissage et l'adaptation, les résultats obtenus et une étude comparative qui touche les hypers paramètres du FTTA.

3.4.1 Prétraitement des données

Pour le prétraitement des données, on a appliqué la procédure déjà utilisée par la méthode gagnante de la première édition de la compétition Hecktor 2020 (Iantsen *et al.* (2021)). Suivant cette procédure, les images médicales provenant des cinq centres sont toutes d'abord transférées vers une résolution commune $1 \times 1 \times 1 mm^3$ avant d'être coupées en utilisant les coordonnées des boîtes englobantes fournies avec la base de données pour se trouver enfin avec des images de taille $144 \times 144 \times 144$. La dernière étape du prétraitement des données serait de normaliser les intensités des CT-Scans pour les transférer dans la plage des unités de Hounsfield $[-1024, 1024]$ puis dans la plage $[-1, 1]$ et normaliser les intensités des PET-Scans en s'appuyant sur la

normalisation Z-Score ainsi la moyenne des intensités de la PET-Scan serait 0 et l'écart type serait 1.

Les deux figures 3.4 et 3.5 illustrent un exemple des deux modalités CT-Scan et PET-Scan d'un même patient avant et après l'application du prétraitement des données. Comme montrent ces deux figures, après être coupées et normalisées, les images médicales deviennent moins complexes ce qui facilite l'apprentissage du modèle ML après.



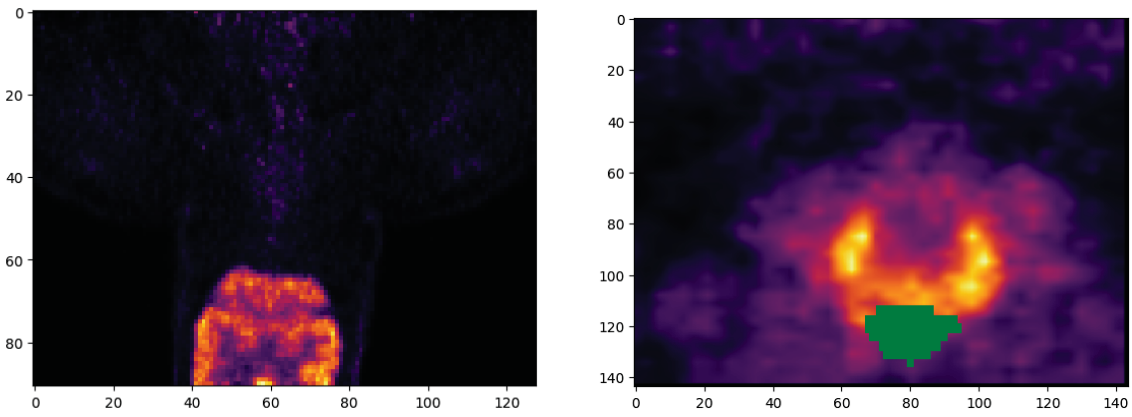
a) Avant le prétraitement

b) Avant le prétraitement, segmenté

c) Après le prétraitement

d) Après le prétraitement, segmenté

Figure 3.4 Exemple de la procédure de prétraitement des données appliquée à une CT-Scan
Tirée de (Andrearczyk *et al.* (2022))



a) Avant le prétraitement b) Après le prétraitement, segmenté

Figure 3.5 Exemple de la procédure de prétraitement des données appliquée à une PET-Scan
Tirée de (Andrearczyk *et al.* (2022))

3.4.2 Entraînement et FTTA

L'entraînement de notre réseau de neurones Unet a été fait sur 150 épisodes avec l'optimisateur Adam en utilisant un learning rate égale à 5e−6 qui se décroît par un facteur de 0.1 après chaque 20 épisodes. Pendant la phase d'apprentissage, une augmentation de données a été appliquée à certaines données d'entraînement de façon aléatoire sous la forme de rotations ou d'effet miroir.

Pour le FTTA, on a utilisé le même optimisateur Adam avec un learning rate égale à 1e−3. Comme expliqué dans les sections précédentes, l'ajustement se fait sur une seule étape d'une façon discontinue pour chaque nouveau sujet qui appartient à la base du test et uniquement les paramètres des couches de normalisation BN sont impliqués par la procédure d'adaptation.

Pour l'évaluation du modèle et l'étude comparative, on a utilisé l'indicateur de similarité Dice DSC donné par l'équation 2.1.

3.4.3 Résultats

Comme montre le tableau 3.3, l'application du FTTA pour la segmentation des tumeurs de la tête et du cou permet un gain de performances pour le modèle entrainé qui se situe entre 2% pour la décomposition CHGJ-TTA et 17% pour la décomposition CHUP-TTA. Ces résultats confirment bien notre assertion au niveau du 2ème chapitre où on a suggéré que l'association du FTTA à la co-segmentation des tumeurs de la tête et du cou peut aboutir à des meilleures performances du modèle ML.

Tableau 3.3 Les résultats de la méthode Tent appliquée à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue et qui se base sur une seule étape d'adaptation pour chaque nouveau sujet

Paramètres	**Baseline**	**FTTA-Dis1**
Décomp N°1 : CHUP-TTA	0.551	**0.721** (+ 17%)
Décomp N°2 : CHUM-TTA	0.504	**0.617** (+ 11%)
Décomp N°3 : CHMR-TTA	0.624	**0.684** (+ 6%)
Décomp N°4 : CHUS-TTA	0.649	**0.68** (+ 3%)
Décomp N°5 : CHGJ-TTA	0.72	**0.741** (+ 2%)

La figure 3.6 illustre une comparaison qualitative des performances de segmentation qui peuvent être achevées par le modèle entraîné sans et après la phase d'adaptation. Cette comparaison visuelle montre que pour certaines données du domaine cible, le modèle trouve des grandes difficultés à segmenter la tumeur en se référant uniquement à son apprentissage basé sur les données sources contrairement à sa performance après l'adaptation où il a pu récupérer une structure proche du ground truth. En effet, pour cet exemple présenté la performance du modèle a passé d'un score de similarité Dice égale à 0.006 sans adaptation à un score de similarité Dice égale à 0.65 après adaptation ce qui prouve la grande importance de la phase d'adaptation.

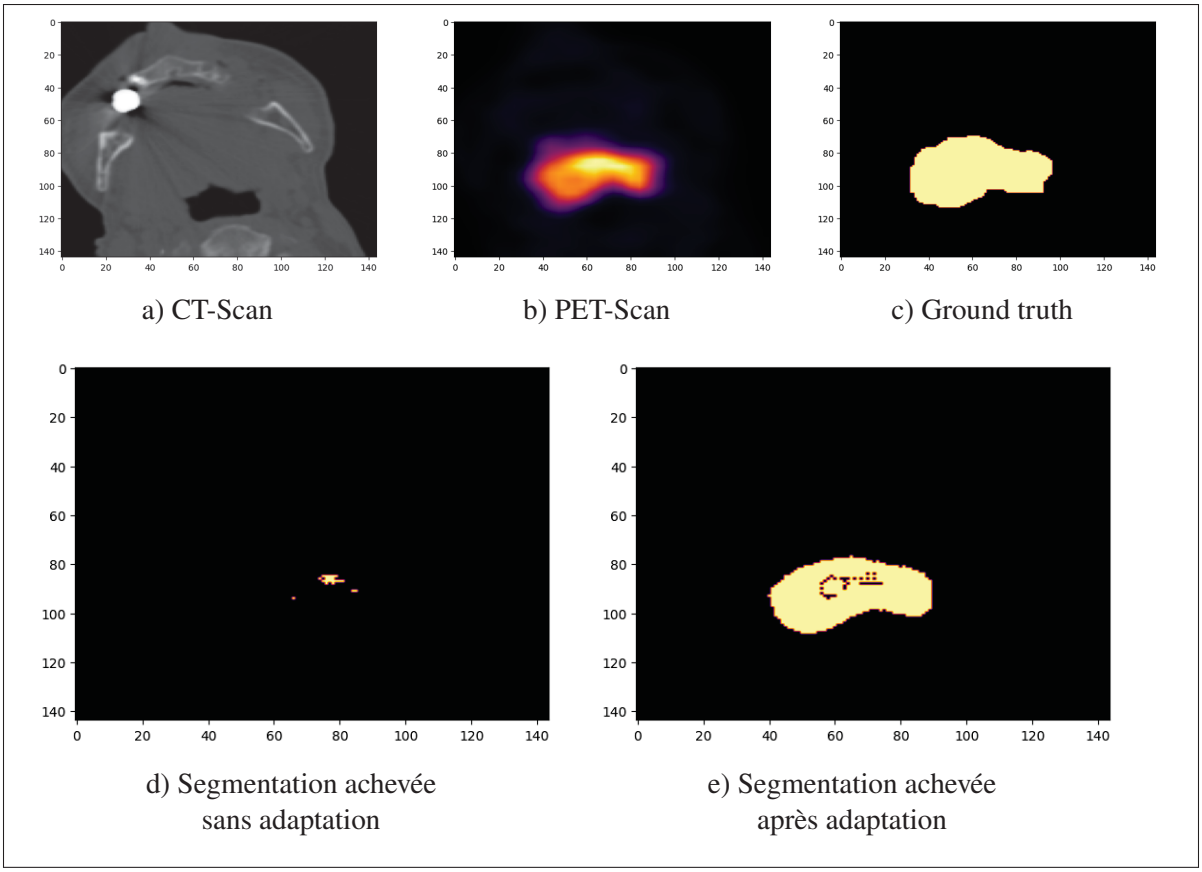


Figure 3.6 Comparaison qualitative entre les performances de segmentation achevée par le modèle entrainé avant et après l'adaptation

3.4.4 Étude Comparative

Les résultats obtenus lors de l'application du FTTA-Dis1 pour la segmentation des tumeurs de la tête et du cou ouvrent la porte pour plusieurs questions dont les plus importants sont : Premièrement, est-ce qu'il y a une différence entre l'adaptation du modèle de façon discontinue et continue ? Deuxièmement, quel est l'impact du nombre d'étapes d'adaptation et les fonctions des pertes utilisées lors de l'apprentissage et de l'inférence sur la performance du FTTA ? Enfin, est-ce-que le gain des performances achevé provient effectivement de l'adaptation des paramètres appris par le réseau des couches de normalisation du modèle entraîné ?

Dans cette étude comparative, on va discuter les questions précédentes pour comprendre comment fonctionne le FTTA et quels sont les paramètres les plus pertinents pour son application.

3.4.4.1 Mode d'adaptation, continu ou discontinu

Tout d'abord on va étudier l'impact du mode d'adaptation sur la performance du FTTA. Ainsi on va comparer l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou avec un ajustement qui se base sur une seule étape pour chaque nouveau sujet en mode continu FTTA-C1 et en mode discontinu FTTA-Dis1 pour comprendre l'évolution des connaissances apprises par le modèle pendant la phase d'inférence suivant les deux modes. Les résultats obtenus sont exprimés dans le tableau 3.4.

Tableau 3.4 Comparaison entre le mode continu et discontinu de la méthode Tent appliquée à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se base sur une seule étape d'adaptation par nouveau sujet

Paramètres	**Baseline**	**FTTA-Dis1**	**FTTA-C1**		
			Perm N°1	**Perm N°2**	**Perm N°3**
Décomp N°1 : CHUP-TTA	0.551	**0.721**	0.725 (+ 0.004)	0.72 (- 0.001)	0.724 (+ 0.003)
Décomp N°2 : CHUM-TTA	0.504	**0.617**	0.612 (- 0.005)	0.615 (- 0.002)	0.61 (- 0.007)
Décomp N°3 : CHMR-TTA	0.624	**0.684**	0.696 (+ 0.012)	0.691 (+ 0.007)	0.7 (+ 0.006)
Décomp N°4 : CHUS-TTA	0.649	**0.68**	0.676 (- 0.004)	0.668 (-0.012)	0.671 (- 0.009)
Décomp N°5 : CHGJ-TTA	0.72	**0.741**	0.719 (- 0.022)	0.715 (- 0.026)	0.718 (- 0.023)

Ces résultats montrent que sur plusieurs exécutions, la performance achevée après l'adaptation du modèle reste la même en mode discontinu alors qu'elle est très sensible à l'ordre avec lequel le modèle traite les données en inférence en mode continu. Ce comportement est bien explicable par le fait qu'en mode discontinu l'ajustement du modèle se fait d'une façon indépendante pour chaque nouvelle donnée de test. De cette façon, le modèle commence avec toutes ses connaissances apprises pendant la phase de l'entraînement avant de les adapter à cette donnée d'où on arrive avec les mêmes performances pour la même donnée. En contre partie, en mode continu, pour chaque nouvelle donnée de test, le modèle commence avec l'état qu'il a achevé après l'adaptation faite pour la donnée précédente d'où une variation de performance avec chaque nouvelle permutation lors de la phase d'inférence.

Quoique la variation de performance en mode continu est négligeable dans notre cas, mais cela est dû au petit nombre de données de test en premier lieu et en deuxième lieu à l'unicité de la source de ces données qui implique une certaine similitude entre les données de test. Ainsi le modèle peut améliorer ses performances en se basant sur les connaissances apprises lors de son adaptation continu en inférence comme le cas de la première permutation de la troisième décomposition. Dans un contre-exemple, cette évolution de connaissances en mode continu peut dégrader la performance du modèle comme pour la deuxième permutation de la cinquième décomposition qui peut être expliquée par la présence de certaines données aberrantes parmi les données du test. Cette dégradation peut être énorme avec l'augmentation du nombre de données aberrantes parmi les données de test où le modèle va diverger en oubliant les connaissances apprises lors de son entraînement.

Pour conclure, on peut dire que le mode d'adaptation discontinu semble d'être plus réel et plus applicable surtout lorsqu'on ne garantit pas qu'il n'y a pas de changement des distributions de données au niveau des données de la phase d'inférence et qu'il n'y a pas de données aberrantes parmi ces données.

3.4.4.2 L'effet du nombre d'étapes

Dans un deuxième temps, on va étudier l'effet du nombre d'étapes d'adaptation sur la performance du FTTA. Dans ce contexte, on va comparer l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou avec un ajustement qui se base sur une seule étape FTTA-Dis1, cinq étapes FTTA-Dis5, vingt étapes FTTA-Dis20 et cinquante étapes FTTA-Dis50 en mode discontinu. Cette démarche nous permet de comprendre l'effet du nombre d'étapes de l'optimisation de l'entropie de Shannon sur le niveau de confiance du modèle au moment de la décision pendant la phase de l'inférence. Le tableau 3.5 rend compte des résultats obtenus.

Tableau 3.5 Comparaison sur l'effet du nombre d'étapes lors de l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue

Paramètres	**Baseline**	**FTTA-Dis1**	**FTTA-Dis5**	**FTTA-Dis20**	**FTTA-Dis50**
Décomp N°1 : CHUP-TTA	0.551	**0.721**	0.726 (+ 0.005)	0.728 (+ 0.007)	0.715 (- 0.006)
Décomp N°2 : CHUM-TTA	0.504	**0.617**	0.618 (+ 0.001)	0.598 (- 0.019)	0.543 (- 0.074)
Décomp N°3 : CHMR-TTA	0.624	**0.684**	0.693 (+ 0.009)	0.695 (+ 0.011)	0.659 (- 0.025)
Décomp N°4 : CHUS-TTA	0.649	**0.68**	0.682 (+0.002)	0.676 (- 0.004)	0.624 (- 0.056)
Décomp N°5 : CHGJ-TTA	0.72	**0.741**	0.735 (- 0.006)	0.706 (- 0.035)	0.642 (- 0.099)

Ces résultats font apparaître que l'augmentation du nombre d'étapes de l'optimisation de la fonction de perte en adaptant le modèle entraîné pendant la phase d'inférence a un impact sur

l'état de performance achevé par ce dernier après l'adaptation. Cet impact est en corelation avec le nombre d'étapes. Ainsi il peut êter légèrement positif ou negatif lorsqu'on passe vers cinq étapes d'optimisation comme pour le cas de la troisième décomposition et la cinquième décomposition successivement. En contre partie, il devient plus grave et totalement négatif en passant vers cinquante étapes d'optimisation comme le cas de la décomposition cinq où la performance du modèle a diminué par 10%. Cette dégradation est expliquée par l'état de sur-confiance atteint par le modèle après 50 étapes d'optimisation et qui lui force d'assigner plusieurs pixels à des classes erronées lorsqu'il fait la segmentation après la phase d'adaptation.

Avec l'impact du nombre d'étapes sur la performance finale achevée par le modèle après l'application du FTTA, la courbe présentée par la figure 3.7 montre que ce paramètre a un grand impact aussi sur le temps nécessaire pour faire l'adaptation du modèle entrainé pour chaque nouveau sujet de la base de test. En effet, le temps nécessaire passe de 0.16 seconde par nouveau sujet lorsqu'on on se limite à une seule étape à 1.78 seconde par nouveau sujet lorsqu'on applique cinq étapes d'optimisation. Bien que cette augmentation du temps nécessaire pour faire l'adaptation ne pose pas un grand problème dans une telle application dédiée pour faire la segmentation des images médicales où la précision des résultats est le facteur le plus important, elle pose un grand problème pour les applications en temps réel qui cherche à adapter le modèle avec la même vitesse avec laquelle arrive le flux des données du test.

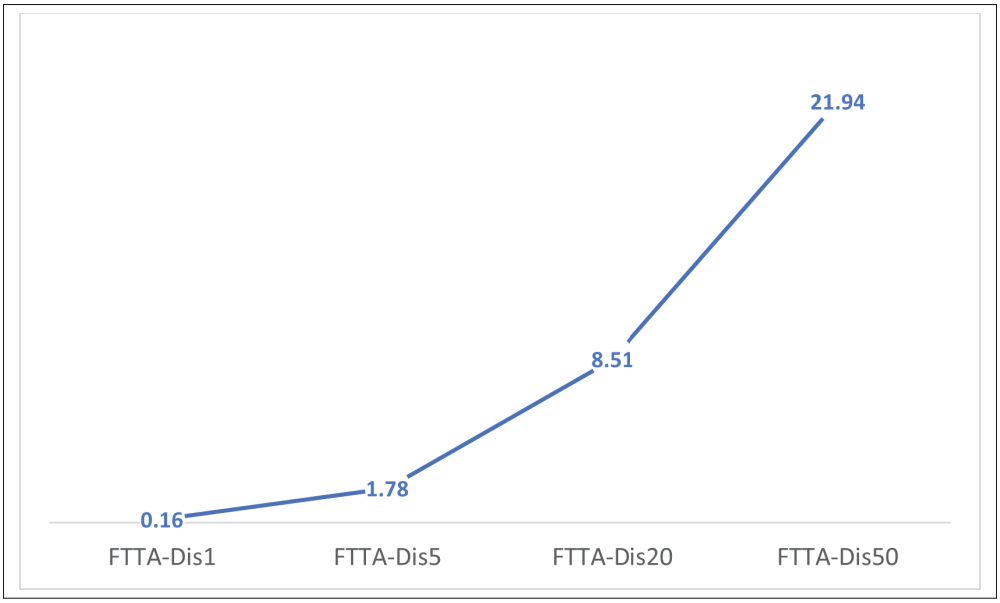


Figure 3.7 L'effet du nombre d'étapes sur le temps nécessaire pour effectuer l'adaptation à un seul sujet, donné en seconde

Pour conclure, on peut dire qu'un grand nombre d'étapes pendant l'optimisation de la fonction de perte en adaptant le modèle lors de l'inférence va aboutir à une sur-confiance qui va dégénérer la performance du modèle pour cela il faut se limiter à un petit nombre d'étapes. Cependant, un nombre d'étapes supérieur à un peut ne pas être applicable à une application temps réelle où l'optimisation de la fonction d'erreur doit se faire à une vitesse supérieure à celle du flux des données à traiter.

3.4.4.3 L'impact des fonctions de perte

Dans cette partie, on va étudier l'impact des fonctions de perte sur la performance du FTTA. Pour cela on va comparer l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou avec un ajustement qui se base sur une seule étape FTTA-Dis1 où le modèle est entrainé en optimisant l'entropie croisée et adapté en minimisant l'entropie de Shannon CE-Ent en premier lieu. En deuxième lieu, notre modèle est entrainé en optimisant l'erreur quadratique moyenne et adapté en minimisant une fonction de perte sous la forme d'une combinaison de

l'entropie de Shannon et une minimisation de la variation de l'intensité intra-classes MSE-IvEnt. Le choix des fonctions de perte selon cette manière est basé sur l'étude (Goyal *et al.* (2022)) qui montre que la fonction de perte optimale à utiliser lors de la phase d'adaptation dépend de celle utilisée lors de l'entrainement comme présenté dans la partie 2.2.2.

Avant de discuter les résultats, on commence par présenter les deux nouvelles fonctions de perte utilisées dans cette partie :

- L'erreur quadratique moyenne est une mesure fréquemment utilisée dans le domaine de l'apprentissage machine pour entraîner des modèles ML. Elle sert à comparer les valeurs des pixels prédites par le modèle avec les valeurs réelles des pixels correspondants dans les masques de segmentation dans le cas de l'apprentissage supervisé de la segmentation automatique des images médicales. Pour notre étude comparative, le modèle $f(\theta)$ sera entrainé en optimisant l'erreur quadratique moyenne comme fonction de perte donnée par l'équation 3.2.

$$\min_{\theta} \sum_{\mathcal{I}_m} -\frac{1}{|\Omega_s|} ||\mathcal{Y}_i - \hat{\mathcal{Y}}(i,\theta)|| \tag{3.2}$$

- La minimisation de la variation des intensités intra-classes est un terme de régularisation inspiré de l'étude (Luo *et al.* (2022)). Elle sert à minimiser la variation entre les intensités des différents pixels qui appartiennent à la même région dans les masques de segmentation prédits par le modèle. Pendant la phase d'adaptation lors de notre étude comparative, ce terme régularisateur est associé avec l'entropie de Shannon avec un facteur λ pour ajuster le modèle à chaque nouveau sujet de la base du test selon l'équation 3.3.

$$\min_{\theta} \sum_{\mathcal{I}_m} -\frac{1}{|\Omega_t|} \hat{\mathcal{Y}}(i,\theta) \log(\hat{\mathcal{Y}}(i,\theta)) + \lambda \sum_{\mathcal{I}_m} \frac{(\hat{\mathcal{Y}}(i,\theta) \cdot \mathcal{I}_i - \rho_t)^2}{\hat{\mathcal{Y}}(i,\theta)} \tag{3.3}$$

avec

$$\rho_t = \frac{\sum_{i \in \Omega_t} \hat{\mathcal{Y}}(i,\theta) \cdot \mathcal{I}_i}{\sum_{i \in \Omega_t} \hat{\mathcal{Y}}(i,\theta)} \tag{3.4}$$

Tableau 3.6 Comparaison sur l'impact des fonctions de perte pour l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou, avec un ajustement qui se fait de façon discontinue et qui se base sur une seule étape d'adaptation par nouveau sujet

Paramètres	**CE-Ent**		**MSE-IvEnt**	
	Baseline	**FTTA-Dis1**	**Baseline**	**FTTA-Dis1**
Décomp N°1 : CHUP-TTA	0.551	**0.721** (+ 17%)	0.575	**0.721** (+ 15%)
Décomp N°2 : CHUM-TTA	0.504	**0.617** (+ 11%)	0.532	**0.644** (+ 11%)
Décomp N°3 : CHMR-TTA	0.624	**0.684** (+ 6%)	0.57	**0.698** (+ 13%)
Décomp N°4 : CHUS-TTA	0.649	**0.68** (+ 3%)	0.609	**0.64** (+ 3%)
Décomp N°5 : CHGJ-TTA	0.72	**0.741** (+ 2%)	0.727	**0.767** (+ 4%)

Les résultats trouvés sont exprimés dans le tableau 3.6. À noter que les mêmes paramètres déjà presentés dans la section 3.4.2 ont été utilisés pour l'entraînement et l'adaptation du modèle dans le cas MSE-IvEnt avec un λ égale à 1e$-$4. Ce paramètre de pondération a été choisi en se basant sur trois sujets de validation de chaque centre laissés pour la phase d'inférence de chaque décomposition.

Les résultats observés montrent qu'un simple changement de la fonction de perte lors de l'apprentissage d'un modèle ML peut aboutir à une amélioration de ce modèle, comme pour la première décomposition où le baseline a passé d'une performance de $55,1\%$ lorsqu'il est entrainé avec l'entropie croisée à une performance de $57,5\%$ lorsqu'il est entrainé avec l'erreur quadratique moyenne. De la même façon il peut aboutir à une dégradation de ce modèle, comme pour la troisième décomposition où la performance du baseline a passé de $62,4\%$ en utilisant l'entropie croisée à 57% en utilisant l'erreur quadratique moyenne.

Bien que ce changement de performance ait un impact sur l'état initiale avec lequel le modèle va commencer la phase d'adaptation, on remarque que cette adaptation aboutit à un taux similaire d'amélioration de performances pour les décompositions numéro un, deux, trois et cinq dans les deux cas CE-Ent et MSE-IvEnt. Par contre, pour la décomposition numéro quatre, ce taux a subi un changement plus important passant de 6% à 13% en passant de la configuration CE-Ent vers la configuration MSE-IvEnt. Cette exception peut être explicable par le fait que cette décomposition a le plus petit nombre des sujets de test.

Une conclusion qu'on peut tirer de notre comparaison est que lorsqu'on utilise la fonction de perte optimale qui correspond à la fonction d'erreur avec laquelle notre modèle ML a fait son apprentissage selon (Goyal *et al.* (2022)), on va se trouver avec un taux d'amélioration légèrement différent quand on adapte notre modèle en passant d'une configuration à une autre ce qui néglige l'impact des fonctions de perte sur l'application du FTTA.

3.4.4.4 L'impact des paramètres à adapter

Pour clôturer notre étude comparative, on va étudier l'impact des paramètres adaptés sur le FTTA. Ainsi on va comparer l'application de la méthode Tent à la segmentation des tumeurs de la tête et du cou avec un ajustement qui se base sur une seule étape FTTA-Dis1 avec le modèle non adapté qui utilise les statistiques de la population d'apprentissage au niveau de ses couches de normalisation Basline d'une part et le modèle non adapté qui utilisent les statistiques du lot de test qu'il traite Baseline-BS de l'autre part.

Tableau 3.7 Comparaison à propos l'effet des paramètres des couches de normalisation sur la performance finale de l'ajustement du modèle

Paramètres	**Baseline**	**Baseline-BS**	**FTTA-Dis1**
Décomp N°1 : CHUP-TTA	0.551	0.719 **(+ 0.168)**	0.721 (+ 0.002)
Décomp N°2 : CHUM-TTA	0.504	0.616 **(+ 0.112)**	0.617 (+ 0.001)
Décomp N°3 : CHMR-TTA	0.624	0.681 **(+ 0.057)**	0.684 (+ 0.003)
Décomp N°4 : CHUS-TTA	0.649	0.679 **(+ 0.03)**	0.68 (+ 0.001)
Décomp N°5 : CHGJ-TTA	0.72	0.741 **(+ 0.021)**	0.741 (±0)

Les résultats obtenus montrent que sans adaptation et en forçant le modèle à utiliser les statistiques du sujet à traiter pendant la phase d'inférence (Baseline-BS) au lieu d'utiliser les statistiques de la population d'entraînement (Baseline), sa performance augmente d'une façon significative comme le cas de la première décomposition où on voit un taux d'amélioration égale à $16,8\%$. Par contre, ce taux d'amélioration devient négligeable en comparant le modèle non adapté qui utilise les statistiques du lot de test (Baseline-BS) avec celui ci adapté et qui utilisent aussi les statistiques du lot de test (FTTA-Dis1).

Ces observations nous permettent de tirer une première conclusion que les paramètres statistiques des couches de normalisation d'un modèle ML ont un effet plus lourd sur la performance de ce modèle en inférence que les paramètres α et β de ces couches. Cette observation, nous pousse à penser dans le futur à ignorer l'adaptation des paramètres α et β des couches de normalisation d'un modèle ML dans la phase d'inférence. Au lieu de ça on cherche les paramètres statistiques les plus robustes qu'il peut utiliser pour atteindre des meilleurs résultats.

Mais la question qui se pose avant ça, est-ce-que ces observations à propos les paramètres statistiques et les paramètres appris par le réseau des couches de normalisation reste valide pour d'autres scénarios ? La réponse à cette question vient dans le chapitre suivant.

CHAPITRE 4

FTTA FOR CIFAR10

Dans ce chapitre on va étendre nos expérimentations vers un autre scénario de classification d'images naturelles en utilisant l'ensemble de données CIFAR-10 pour généraliser les conclusions tirées dans la partie 3.4.4.4 du troisième chapitre à propos l'impact de la nature des paramètres adaptés sur le processus FTTA. Ainsi on va commencer par présenter la base de données et le réseau de neurones utilisés pour cette tâche avant de décrire notre démarche expérimentale et discuter les résultats trouvés.

4.1 Base de données

Dans cette partie, on a utilisé l'ensemble de données public CIFAR-10 (Krizhevsky, Hinton *et al.* (2009)) pour la phase d'apprentissage de notre modèle ML. Cet ensemble de données est constitué de 60000 images colorées de taille 32×32 qui appartiennent à 10 classes différents qui sont : avion, voiture, oiseau, chat, cerf, chien, camion, bateau, cheval, grenouille. Ainsi chaque classe contient 6000 images.

Pour la phase de l'adaptation, on a appliqué le FTTA du modèle ML en utilisant l'ensemble de données CIFAR-10-C (Hendrycks & Dietterich (2019)) dérivé de CIFAR-10 et qui contient différentes corruptions appliquées aux images d'origine avec cinq degrés de sévérité différentes comme illustrer dans la figure 4.1.

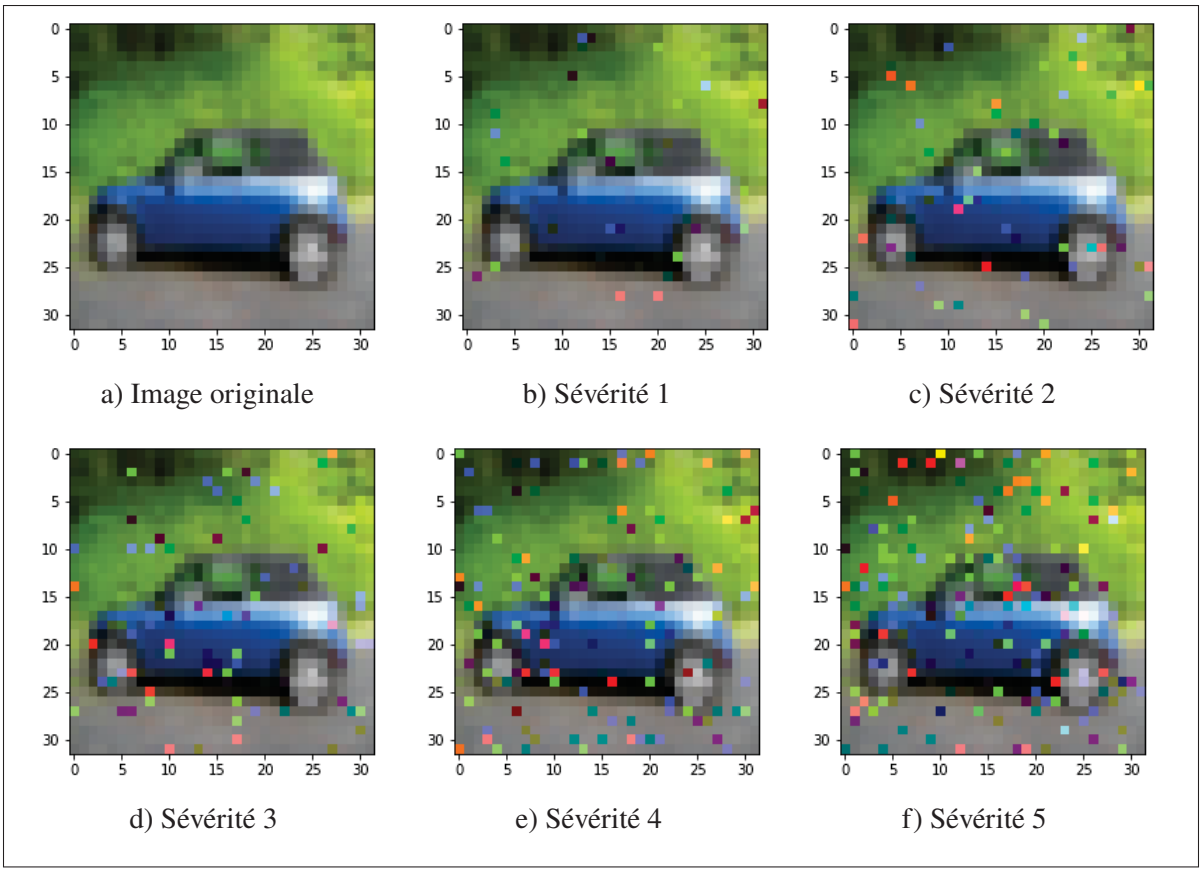


Figure 4.1 Comparaison qualitative entre les différentes sévérités de corruptions appliquées à la base CIFAR-10 pour un exemple de bruit impulsionnel

Comme le montre la figure 4.2 les corruptions utilisées dans ce travail sont les suivantes : bruit gaussien (gaussian noise), bruit de photon (shot noise), bruit impulsionnel (impulse noise), flou de mise au point (defocus blur), flou de vitre (glass blur), flou cinétique (motion blur), flou de zoom (zoom blur), corruption de neige (snow), corruption de givre (frost), corruption de brouillard (fog), corruption de luminosité (brightness), corruption de contraste (contrast), transformation élastique (elastic transform) et compression jpeg (jpeg compression). L'objectif de ces corruptions pendant la phase d'adaptation est de tester la capacité de notre modèle ML déjà entrainé à généraliser des nouvelles distributions de données non vues en appliquant la procédure FTTA.

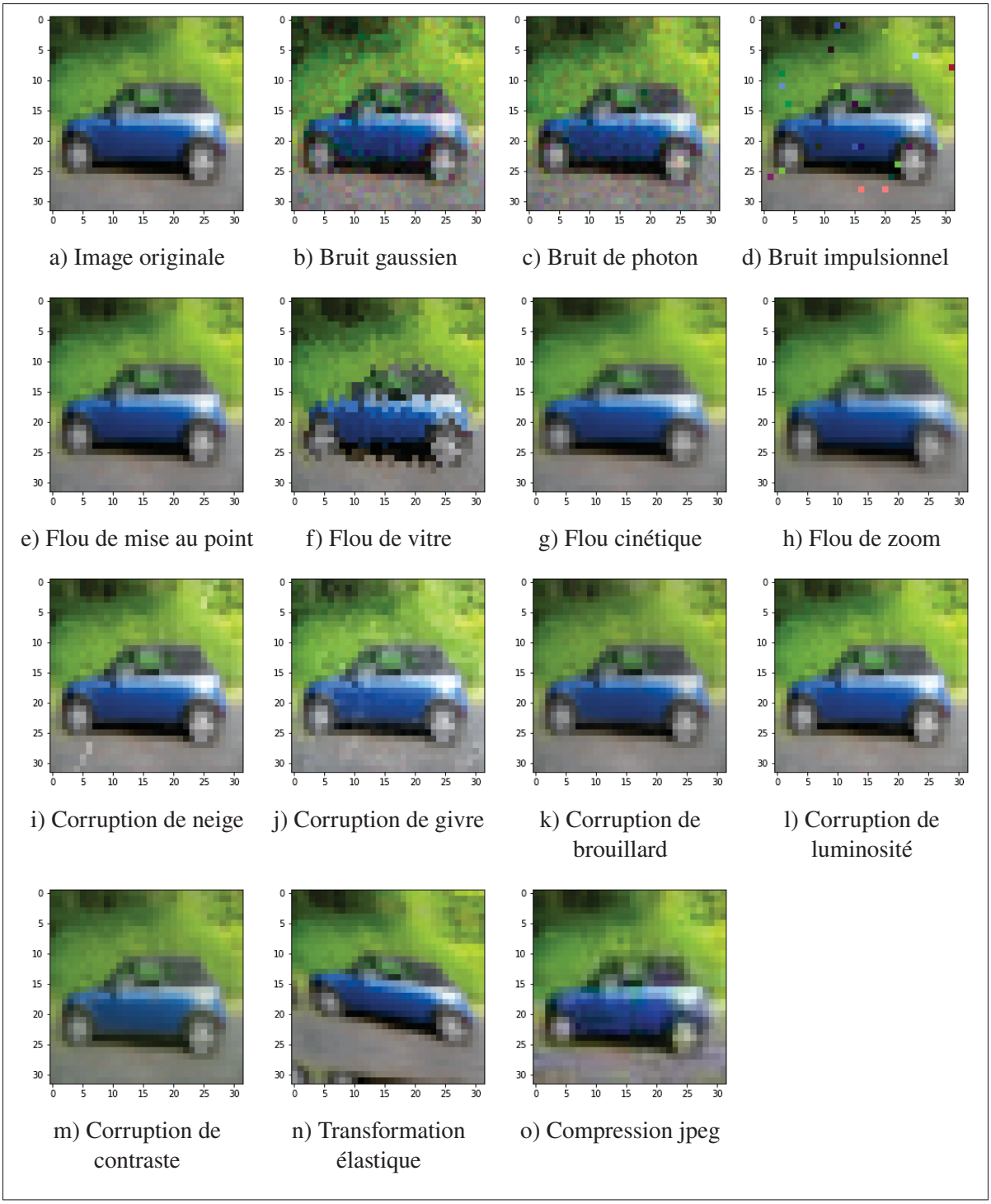


Figure 4.2 Comparaison qualitative entre les différentes corruptions utilisées dans ce travail

4.2 Le réseau de neurones

Dans notre travail, le réseau de neurones considéré pour faire la tâche de classification d'images est le Resnet-26. Cette architecture, formée de 26 couches de neurones, fait partie des variantes de la fameuse architecture Resnet (He, Zhang, Ren & Sun (2016)) inventée pour surmonter le problème de disparition du gradient qui se pose fortement quand les CNNs deviennent plus profonds. Ainsi la solution proposée par cette architecture est d'utiliser les connexions résiduelles. Ces connexions donnent la possibilité d'ignorer certaines couches dans le réseau en transmettant l'information de la couche d'entrée vers la couche de sortie directement comme illustré dans la figure 4.3. De cette façon on peut garder une valeur plus significative du gradient tout en considérant le nombre de couches du CNN.

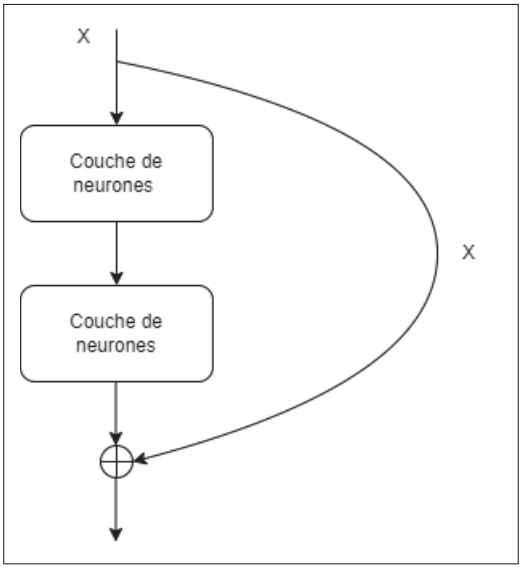


Figure 4.3 Bloc résiduel

4.3 La démarche expérimentale

L'apprentissage du modèle Resnet-26 s'est fait en optimisant l'entropie croisée sur 200 épisodes avec un batch size égale à 256, un optimisateur de descente de gradient stochastique (SGD) et un learning rate égale à 0.1 qui se met à jour lors de l'entrainement à l'aide de l'algorithme de régulation de taux d'apprentissage Cosine Annealing.

Pour le FTTA, on a utilisé l'optimisateur Adam avec un mearning rate égale à 1e−3. L'adaptation s'est fait avec un batch size égale à 200 d'une façon discontinue pour chaque nouveau lot de test lors du quelle uniquement les paramètres des couches de normalisation BN sont impliqués par l'optimisation de l'entropie de Shannon.

L'évaluation s'est fait en comparant le taux de classification (accuracy) du Baseline avec les configurations Baseline-BS, FTTA-Dis1 et FTTA-Dis20.

4.4 Résultats

Dans la suite, les résultats obtenus de l'adaptation du Resnet-26 aux différentes corruptions citées dans la partie 4.1 sont exprimés. Les résultats de chaque corruption sont exprimés dans un tableau spécifique qui contient une comparaison du taux de classification entre le Baseline-BS d'une part et le Baseline, la configuration FTTA-Dis1 et la configuration FTTA-Dis20 d'autre part. Aussi une comparaison de la moyenne de ce taux de classification par sévérité est donnée dans un tableau spécifique 4.15.

Les résultats trouvés permettent de tirer les conclusions suivantes :

- Pour la plupart des corruptions, en utilisant les statistiques du lot de test au niveau des couches de normalisation (Baseline-BS) au lieu d'utiliser les statistiques de la population d'entraînement (Baseline), le taux de classification subit une amélioration significative qui dépasse 29% pour la corruption de nature bruit gaussien. Cette amélioration dépend fortement de la nature de corruption et devient de plus en plus importante en passant vers des degrés de sévérités plus graves.
- Pour les corruptions où il y a une dégradation du performance du modèle en utilisant les statistiques du lot de test (Baseline-BS) comme le cas de la corruption de nature flou de mise au point, on remarque que la dégradation du taux de classification ne dépasse pas 1% dans le pire des cas. Pour ces cas, et celles où le taux de classification ne subit pas une grande amélioration, on peut rendre ça à l'utilisation de mauvaises valeurs statistiques au niveau des couches de normalisation du modèle dû à la présence de valeurs aberrantes au niveau

du lot de test. Ainsi un futur axe de recherche s'ouvre pour trouver des statistiques plus représentatives du lot de test à utiliser au niveau des couches de normalisation lors de la phase d'inférence.

- En passant vers l'adaptation des paramètres appris par le réseau α et β des couches de normalisations avec une seule étape d'optimisation (FTTA-Dis1), on remarque que le taux de classification subit une amélioration très négligeable à l'entourage de 0.09% par rapport au modèle non adapté qui utilise les statistiques du lot de test (Baseline-BS) et même lorsqu'on augmente le nombre d'étapes d'optimisation vers 20 étapes (FTTA-Dis20) le taux d'amélioration ne dépasse pas 0.21% dans le meilleur des cas par rapport au Baseline-BS ce qui pose une grande question sur la vraie utilité de la mise à jour des paramètres appris par le réseau α et β des couches de normalisation du modèle lors de l'adaptation d'un modèle ML.

Ces conclusions prouvent celles tirées à la fin du chapitre 3, selon lesquelles trouver des statistiques robustes à utiliser au niveau des couches de normalisation d'un modèle ML pendant la phase d'inférence semble d'être plus performant qu'adapter les paramètres α et β de ces couches.

Tableau 4.1 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit gaussien de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Bruit Gauss5**	**Bruit Gauss4**	**Bruit Gauss3**	**Bruit Gauss2**	**Bruit Gauss1**	**Moyenne**
Baseline	35.24 %	42.81 %	51.28 %	69.38 %	83.65 %	56.47 %
Baseline-BS	81.52 %	82.95 %	84.72 %	88.21 %	91.01 %	85.68 % (**+ 29.21 %**)
FTTA-Dis1	81.66 %	83.03 %	84.84 %	88.27 %	91.01 %	85.76 % (**+ 0.08 %**)
FTTA-Dis20	81.72 %	83.06 %	84.86 %	88.29 %	91.03 %	85.79 % (**+ 0.11 %**)

Tableau 4.2 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit de photon de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Bruit Photon5**	**Bruit Photon4**	**Bruit Photon3**	**Bruit Photon2**	**Bruit Photon1**	**Moyenne**
Baseline	40.88 %	53.98 %	63.19 %	81.65 %	88.29 %	65.60 %
Baseline-BS	81.68 %	84.83 %	86.42 %	90.86 %	91.75 %	87.11 % (**+ 21.51 %**)
FTTA-Dis1	81.84 %	84.92 %	86.58 %	90.91 %	91.71 %	87.19 % (**+ 0.08 %**)
FTTA-Dis20	81.90 %	84.94 %	86.55 %	90.91 %	91.74 %	87.21 % (**+ 0.1 %**)

Tableau 4.3 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature bruit impulsionnel de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Bruit Impul5**	**Bruit Impul4**	**Bruit Impul3**	**Bruit Impul2**	**Bruit Impul1**	**Moyenne**
Baseline	20.26 %	34.85 %	61.40 %	74.31 %	85.63 %	55.29 %
Baseline-BS	70.85 %	77.23 %	82.55 %	86.22 %	89.73 %	81.32 % (**+ 26.03 %**)
FTTA-Dis1	71.14 %	77.49 %	82.68 %	86.27 %	89.78 %	81.47 % (**+ 0.15 %**)
FTTA-Dis20	71.31 %	77.57 %	82.73 %	86.27 %	89.78 %	81.53 % (**+ 0.21 %**)

Tableau 4.4 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de mise au point de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Flou Defocus5**	**Flou Defocus4**	**Flou Defocus3**	**Flou Defocus2**	**Flou Defocus1**	**Moyenne**
Baseline	90.66 %	93.01 %	94.31 %	94.74 %	94.89 %	93.52 %
Baseline-BS	91.13 %	92.64 %	92.98 %	93.24 %	93.21 %	92.64 % (**- 0.88 %**)
FTTA-Dis1	91.18 %	92.63 %	93 %	93.24 %	93.23 %	92.66 % (**+ 0.02 %**)
FTTA-Dis20	91.18 %	92.63 %	92.98 %	93.26 %	93.24 %	92.66 % (**+ 0.02 %**)

Tableau 4.5 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de vitre de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Flou Vitre5**	**Flou Vitre4**	**Flou Vitre3**	**Flou Vitre2**	**Flou Vitre1**	**Moyenne**
Baseline	66.14 %	65.04 %	78.79 %	77.68 %	77.18 %	72.97 %
Baseline-BS	79.94 %	79.85 %	86.93 %	87.05 %	87.17 %	84.19 % (**+ 11.22 %**)
FTTA-Dis1	80.08 %	80.05 %	86.91 %	87.13 %	87.29 %	84.29 % (**+ 0.1 %**)
FTTA-Dis20	80.09 %	80.16 %	86.95 %	87.20 %	87.35 %	84.35 % (**+ 0.16 %**)

Tableau 4.6 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou cinétique de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Flou Cinétique5**	**Flou Cinétique4**	**Flou Cinétique3**	**Flou Cinétique2**	**Flou Cinétique1**	**Moyenne**
Baseline	83.32 %	88.02 %	88.09 %	91.10 %	93.30 %	88.77 %
Baseline-BS	87.78 %	89.58 %	89.55 %	91.07 %	92.28 %	90.05 % (**+ 1.28 %**)
FTTA-Dis1	87.87 %	89.64 %	89.56 %	91.11 %	92.38 %	90.11 % (**+ 0.06 %**)
FTTA-Dis20	87.88 %	89.67 %	89.58 %	91.11 %	92.36 %	90.12 % (**+ 0.07 %**)

Tableau 4.7 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature flou de zoom de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Flou Zoom5**	**Flou Zoom4**	**Flou Zoom3**	**Flou Zoom2**	**Flou Zoom1**	**Moyenne**
Baseline	91.61 %	93.14 %	93.45 %	94.05 %	93.54 %	93.16 %
Baseline-BS	92.49 %	93.26 %	93.36 %	93.51 %	93.08 %	93.14 % (**- 0.02 %**)
FTTA-Dis1	92.55 %	93.31 %	93.35 %	93.51 %	93.05 %	93.15 % (**+ 0.01 %**)
FTTA-Dis20	92.57 %	93.32 %	93.34 %	93.54 %	93.05 %	93.16 % (**+ 0.02 %**)

Tableau 4.8 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de neige de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Neige5**	**Neige4**	**Neige3**	**Neige2**	**Neige1**	**Moyenne**
Baseline	83.08 %	84.93 %	87.24 %	87.57 %	92.22 %	87.01 %
Baseline-BS	85.7 %	86.18 %	88.54 %	89.38 %	91.84 %	88.33 % (**+ 1.32 %**)
FTTA-Dis1	85.85 %	86.28 %	88.67 %	89.5 %	91.92 %	88.44 % (**+ 0.11 %**)
FTTA-Dis20	85.86 %	86.29 %	88.72 %	89.48 %	91.92 %	88.45 % (**+ 0.12 %**)

Tableau 4.9 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de givre de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Givre5**	**Givre4**	**Givre3**	**Givre2**	**Givre1**	**Moyenne**
Baseline	79.19 %	84.74 %	85.8 %	89.77 %	92.79 %	86.46 %
Baseline-BS	88.67 %	89.83 %	89.59 %	90.78 %	92.49 %	90.27 % (**+ 3.81 %**)
FTTA-Dis1	88.76 %	89.9 %	89.65 %	90.84 %	92.59 %	90.35 % (**+ 0.08 %**)
FTTA-Dis20	88.79 %	89.91 %	89.63 %	90.86 %	92.61 %	90.36 % (**+ 0.09 %**)

Tableau 4.10 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de brouillard de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Brouil5**	**Brouil4**	**Brouil3**	**Brouil2**	**Brouil1**	**Moyenne**
Baseline	71.9 %	87.42 %	91.31 %	93.16 %	94.69 %	87.70 %
Baseline-BS	83.45 %	89.38 %	91.26 %	92.50 %	93.18 %	89.95 % (**+ 2.25 %**)
FTTA-Dis1	83.76 %	89.46 %	91.29 %	92.52 %	93.16 %	90.04 % (**+ 0.09 %**)
FTTA-Dis20	83.78 %	89.51 %	91.31 %	92.55 %	93.15 %	90.06 % (**+ 0.11 %**)

Tableau 4.11 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de luminosité de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Lumino5**	**Lumino4**	**Lumino3**	**Lumino2**	**Lumino1**	**Moyenne**
Baseline	90.67 %	92.96 %	93.69 %	94.42 %	94.76 %	93.3 %
Baseline-BS	91.43 %	92.73 %	93.18 %	93.31 %	93.32 %	92.79 % (**-0.51 %**)
FTTA-Dis1	91.41 %	92.8 %	93.2 %	93.39 %	93.36 %	92.83 % (**+ 0.04 %**)
FTTA-Dis20	91.45 %	92.79 %	93.20 %	93.39 %	93.36 %	92.84 % (**+ 0.05 %**)

Tableau 4.12 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de contraste de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Contrast5**	**Contrast4**	**Contrast3**	**Contrast2**	**Contrast1**	**Moyenne**
Baseline	55.05 %	82.47 %	88.87 %	91.67 %	94.35 %	82.48 %
Baseline-BS	89.39 %	91.8 %	92.38 %	92.73 %	93.16 %	91.89 % (**+ 9.41 %**)
FTTA-Dis1	89.49 %	91.79 %	92.45 %	92.81 %	93.18 %	91.94 % (**+ 0.05 %**)
FTTA-Dis20	89.50 %	91.82 %	92.47 %	92.81 %	93.17 %	91.95 % (**+ 0.06 %**)

Tableau 4.13 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature transformation élastique de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Élastique Transf5**	**Élastique Transf4**	**Élastique Transf3**	**Élastique Transf2**	**Élastique Transf1**	**Moyenne**
Baseline	82.57 %	88.63 %	92.46 %	92.66 %	92.67 %	89.80 %
Baseline-BS	84 %	87.85 %	91.41 %	91.53 %	91.09 %	89.18 % (**- 0.61 %**)
FTTA-Dis1	84.23 %	87.98 %	91.47 %	91.60 %	91.23 %	89.30 % (**+ 0.12 %**)
FTTA-Dis20	84.26 %	87.98 %	91.50 %	91.59 %	91.23 %	89.31 % (**+ 0.13 %**)

Tableau 4.14 Comparaison du taux de classification pour les différentes configurations du FTTA d'un CNN Resnet-26 à une corruption de nature compression jpeg de sévérités différentes appliquée à la base CIFAR-10

Paramètres	**Comp JPEG5**	**Comp JPEG4**	**Comp JPEG3**	**Comp JPEG2**	**Comp JPEG1**	**Moyenne**
Baseline	76.13 %	79.07 %	81.35 %	82.92 %	87.41 %	81.38 %
Baseline-BS	80.14 %	82.86 %	84.72 %	85.64 %	88.90 %	84.45 % (**+ 3.07 %**)
FTTA-Dis1	80.33 %	83.13 %	84.90 %	85.70 %	89.04 %	84.62 % (**+ 0.17 %**)
FTTA-Dis20	80.36 %	83.20 %	84.93 %	85.73 %	89.09 %	84.66 % (**+ 0.21 %**)

Tableau 4.15 Moyenne du taux de classification par sévérité pour les différentes configurations du FTTA d'un CNN Resnet-26 aux corruptions citées dans la partie 4.1 appliquées à la base CIFAR-10

Paramètres	**Sévérité5**	**Sévérité4**	**Sévérité3**	**Sévérité2**	**Sévérité1**	**Moyenne**
Baseline	69.05 %	76.51 %	82.23 %	86.79 %	90.38 %	80.99 %
Baseline-BS	84.87 %	87.21 %	89.11 %	90.43 %	91.59 %	88.64 % (**+ 7.65 %**)
FTTA-Dis1	85.01 %	87.32 %	89.18 %	90.49 %	91.64 %	88.73 % (**+ 0.09 %**)
FTTA-Dis20	85.05 %	87.34 %	89.20 %	90.50 %	91.65 %	88.75 % (**+ 0.11 %**)

CONCLUSION ET RECOMMANDATIONS

Dans ce travail de recherche on a appliqué le Fully Test Time Adaptation pour adapter des réseaux de neurones déjà entrainés pour faire la segmentation automatique des tumeurs de la tête et du cou à des nouvelles distributions de données. Les résultats obtenus montrent que cette procédure d'adaptation permet d'augmenter le taux de précision de ces modèles ce qui confirme les résultats prometteurs cités dans la littérature.

Par la suite, une comparaison qui vise l'impact du mode d'adaptation employé, la fonction de perte utilisée, le nombre d'étapes suivi et les paramètres à adapter concernés a été réaliser tout en appliquant le Fully Test Time Adaptation à la segmentation des tumeurs de la tête et du cou. Cette étude comparative a permis de conclure que les paramètres à adapter représentent l'élément le plus pertinent à la performance finale achevée après l'adaptation tout en montrant que les paramètres statistiques des couches de normalisation d'un modèle ML ont un impact plus lourd sur ce niveau de performance que les paramètres appris par le réseau de ces couches.

Pour généraliser les résultats obtenus par l'étude comparative, on a étendu nos expérimentations vers un scénario de classification d'image naturelle dans le but d'adapter un modèle ML entrainé avec la base CIFAR à des nouvelles distributions obtenues en appliquant des corruptions différentes à la base d'origine. Les résultas trouvés suite à ces nouvelles expérimentations ont confirmé ceux obtenus précédemment à propos l'effet des paramètres statistiques des couches de normalisation par rapport aux paramètres appris par le réseau des couches de normalisation d'un modèle ML lors de son adaptation en inférence.

Les résultats obtenus lors de ce projet de recherche mettent en question la vraie efficacité de mettre à jour les paramètres appris par le réseau des couches de normalisation d'un modèle ML lors de l'application du Fully Test Time Adaptation fortement utilisée dans la littérature ouvrant en même temps la porte vers les axes de recherche focalisant sur trouver les statistiques les plus

robustes à utiliser au niveau des couches de normalisation d'un modèle ML pour l'adapter à des nouvelles distributions de données.

BIBLIOGRAPHIE

Andrearczyk, V., Oreiller, V. & Depeursinge, A. (2020a). Oropharynx detection in PET-CT for tumor segmentation. *Irish Machine Vision and Image Processing*, 188.

Andrearczyk, V., Oreiller, V., Vallières, M., Castelli, J., Elhalawani, H., Jreige, M., Boughdad, S., Prior, J. O. & Depeursinge, A. (2020b). Automatic segmentation of head and neck tumors and nodal metastases in PET-CT scans. *Medical imaging with deep learning*, pp. 33–43.

Andrearczyk, V., Oreiller, V., Jreige, M., Vallières, M., Castelli, J., Elhalawani, H., Boughdad, S., Prior, J. O. & Depeursinge, A. (2021). Overview of the HECKTOR challenge at MICCAI 2020 : automatic head and neck tumor segmentation in PET/CT. *Head and Neck Tumor Segmentation : First Challenge, HECKTOR 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings 1*, pp. 1–21.

Andrearczyk, V., Oreiller, V., Boughdad, S., Rest, C. C. L., Elhalawani, H., Jreige, M., Prior, J. O., Vallières, M., Visvikis, D., Hatt, M. et al. (2022). Overview of the HECKTOR challenge at MICCAI 2021 : automatic head and neck tumor segmentation and outcome prediction in PET/CT images. Dans *Head and Neck Tumor Segmentation and Outcome Prediction : Second Challenge, HECKTOR 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings* (pp. 1–37). Springer.

Andrearczyk, V., Oreiller, V., Abobakr, M., Akhavanallaf, A., Balermpas, P., Boughdad, S., Capriotti, L., Castelli, J., Cheze Le Rest, C., Decazes, P. et al. (2023). Overview of the HECKTOR challenge at MICCAI 2022 : automatic head and neck tumor segmentation and outcome prediction in PET/CT. Dans *Head and Neck Tumor Segmentation and Outcome Prediction : Third Challenge, HECKTOR 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings* (pp. 1–30). Springer.

Bateson, M., Kervadec, H., Dolz, J., Lombaert, H. & Ayed, I. B. [arXiv :2108.03152 [cs]]. (2022a, May). Source-Free Domain Adaptation for Image Segmentation. arXiv. Repéré le 2023-02-04 à http://arxiv.org/abs/2108.03152.

Bateson, M., Lombaert, H. & Ben Ayed, I. (2022b). Test-time adaptation with shape moments for image segmentation. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022 : 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part IV*, pp. 736–745.

Boudiaf, M., Mueller, R., Ayed, I. B. & Bertinetto, L. (2022). *Parameter-free Online Test-time Adaptation* (Rapport n°arXiv :2201.05718). Repéré le 2022-05-17 à http://arxiv.org/abs/2201.05718.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

Cheplygina, V., de Bruijne, M. & Pluim, J. P. (2019). Not-so-supervised : a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, 54, 280–296.

ESMO. (2015). Cancers De La Tête et du cou : un guide pour les patients. Repéré le 2023-04-05 à https://www.esmo.org/content/download/65719/1182654/1/ESMO-ACF-Cancers-de-la-Tete-et-du-Cou-Guide-pour-les-Patients.pdf.

Goyal, S., Sun, M., Raghunathan, A. & Kolter, Z. (2022). Test-time adaptation via conjugate pseudo-labels. *arXiv preprint arXiv :2207.09640*.

Gudi, S., Ghosh-Laskar, S., Agarwal, J. P., Chaudhari, S., Rangarajan, V., Paul, S. N., Upreti, R., Murthy, V., Budrukkar, A. & Gupta, T. (2017). Interobserver variability in the delineation of gross tumour volume and specified organs-at-risk during IMRT for head and neck cancers and the impact of FDG-PET/CT on such variability at the primary site. *Journal of medical imaging and radiation sciences*, 48(2), 184–192.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hendrycks, D. & Dietterich, T. (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *Proceedings of the International Conference on Learning Representations*.

Iantsen, A., Visvikis, D. & Hatt, M. (2021). Squeeze-and-excitation normalization for automated delineation of head and neck primary tumors in combined PET and CT images. *Head and Neck Tumor Segmentation : First Challenge, HECKTOR 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings 1*, pp. 37–43.

Ioffe, S. & Szegedy, C. (2015). Batch normalization : Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, pp. 448–456.

Jin, D., Guo, D., Ho, T.-Y., Harrison, A. P., Xiao, J., Tseng, C.-K. & Lu, L. (2019). Accurate esophageal gross tumor volume segmentation in PET/CT using two-stream chained 3D deep network fusion. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019 : 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II 22*, pp. 182–191.

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images.

Kumar, A., Fulham, M., Feng, D. & Kim, J. (2019). Co-learning feature fusion maps from PET-CT images of lung cancer. *IEEE Transactions on Medical Imaging*, 39(1), 204–217.

Li, L., Zhao, X., Lu, W. & Tan, S. (2020). Deep learning for variational multimodality tumor segmentation in PET/CT. *Neurocomputing*, 392, 277–295.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Laak, J. A. W. M. v. d., Ginneken, B. v. & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. doi : https://doi.org/10.1016/j.media.2017.07.005.

Luo, X., Liao, W., Xiao, J., Chen, J., Song, T., Zhang, X., Li, K., Metaxas, D. N., Wang, G. & Zhang, S. (2022). WORD : A large scale dataset, benchmark and clinical applicable study for abdominal organ segmentation from CT image. *Medical Image Analysis*, 82, 102642. doi : 10.1016/j.media.2022.102642.

Moe, Y. M., Groendahl, A. R., Mulstad, M., Tomic, O., Indahl, U., Dale, E., Malinen, E. & Futsaether, C. M. (2019). Deep learning for automatic tumour segmentation in PET/CT images of patients with head and neck cancers. *arXiv preprint arXiv :1908.00841*.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P. & Tan, M. (2023). Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv :2302.12400*.

Niyas, S., Pawan, S., Kumar, M. A. & Rajan, J. (2022). Medical image segmentation with 3D convolutional neural networks : A survey. *Neurocomputing*, 493, 397–413.

Oreiller, V., Andrearczyk, V., Jreige, M., Boughdad, S., Elhalawani, H., Castelli, J., Vallieres, M., Zhu, S., Xie, J., Peng, Y. et al. (2022). Head and neck tumor segmentation in PET/CT : the HECKTOR challenge. *Medical image analysis*, 77, 102336.

Pan, S. J. & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. doi : 10.1109/TKDE.2009.191. Conference Name : IEEE Transactions on Knowledge and Data Engineering.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M. & Dean, J. (2021). Carbon Emissions and Large Neural Network Training.

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net : Convolutional Networks for Biomedical Image Segmentation. Dans Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F. (Éds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (vol. 9351, pp. 234–241). Cham : Springer International Publishing. doi : 10.1007/978-3-319-24574-4_28.

Sivanesan, U., Braga, L. H., Sonnadara, R. R. & Dhindsa, K. [arXiv :1911.05140 [cs, eess]]. (2019, nov). Unsupervised Medical Image Segmentation with Adversarial Networks : From Edge Diagrams to Segmentation Maps. arXiv. Repéré le 2023-03-02 à http://arxiv.org/abs/1911.05140.

Storkey, A. et al. (2009). When training and test sets are different : characterizing learning transfer. *Dataset shift in machine learning*, 30, 3–28.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. & Hardt, M. (2020). Test-time training with self-supervision for generalization under distribution shifts. *International conference on machine learning*, pp. 9229–9248.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B. & Darrell, T. (2020). Tent : Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv :2006.10726*.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014). How transferable are features in deep neural networks ? *Advances in neural information processing systems*, 27, 9.

Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. (2017, Oct). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251. doi : 10.1109/ICCV.2017.244.