

Segmentation médicale d'un coeur

Abderraouf Haddouni, Omar Zoubir, Paul Sade, Samba Camara

Apprentissage profond pour la vision par ordinateur (MTI865)

Travail remis au professeur Jose Dolz, 23 Décembre 2023

École de Technologie Supérieure (ÉTS), Montréal, QC

Abstract

Dans le cadre du cours MTI865 sur l'apprentissage profond pour la vision par ordinateur, nous avons relevé le défi de la segmentation d'images médicales, également connu sous le nom de challenge ACDC. Le but est de développer le modèle le plus performant capable de segmenter des images de coeur en 2D selon le myocarde, le ventricule gauche et le ventricule droite. Les données d'entraînement comprennent 204 images labellisées et 1004 non labellisées, avec un jeu de validation de 74 images. Le jeu de données de test est réservé au professeur pour l'évaluation finale. Le projet s'inscrit dans le cadre d'une compétition entre 10 équipes. Toutes les équipes étaient constituées de 4-5 personnes et un code de base avec une implémentation du modèle Unet nous a été fourni par le professeur. Le classement de la compétition a été déterminé selon les métriques de performance suivantes: le coefficient Dice, la distance de Hausdorff (HDD) et la distance de surface moyenne (ASD). Les contraintes imposées incluent l'utilisation de PyTorch, une bibliothèque bien documentée. L'accent de ce projet est mis sur l'exploration de stratégies, l'optimisation des paramètres et la compréhension des facteurs influençant les performances du modèle. Enfin, notons que notre équipe s'est classée 4^{ème} dans cette compétition [9].

1 Introduction

L'imagerie médicale joue un rôle important dans la vie du patient. En effet, elle constitue un outil indispensable pour les professionnels de la santé, permettant d'analyser et de poser des diagnostics qui conduiront à un traitement adéquat. En ce sens, l'interprétation visuelle des médecins est souvent la seule stratégie utilisée. Néanmoins, les défis rencontrés dans le processus d'imagerie médicale peuvent parfois limiter l'efficacité de cette interprétation visuelle. Parmi ces défis, la qualité des images et la complexité de la localisation de certaines anomalies, comme les tumeurs, peuvent poser problème. Ainsi, les solutions proposées par l'apprentissage profond, telles que la segmentation

d’images, sont d’une grande aide. En détectant les zones potentielles de tumeur avec un certain pourcentage d’erreur, par exemple, les médecins disposent d’un outil les aidant dans la prise de décision [4]. Dans le cadre de notre travail, nous développerons un modèle capable de segmenter le cœur en trois parties (le ventricule droit (VD), le ventricule gauche (VG) et le myocarde (MYO)) [3]. Les sections suivantes aborderont la méthodologie, les résultats obtenus, et enfin, une bref conclusion.

2 Méthodologie

Dans cette section, vous trouverez la méthodologie utilisée pour notre projet de segmentation d’images cardiaques. Cette méthodologie englobe plusieurs aspects clés, tels que les hyperparamètres choisis pour l’entraînement, le choix de l’architecture du modèle, le type d’apprentissage, la fonction de coût, et enfin, les optimiseurs.

2.1 Type d’apprentissage

Dans cette section, nous allons expliquer la méthode d’entraînement semi-supervisé que nous avons choisie. Il est à noter que nous avons utilisé la technique de l’arrêt précoce (Early Stop) avec une patience de 10. En d’autres termes, si l’erreur sur l’ensemble de validation ne diminue pas pendant trois itérations consécutives, nous mettons fin à l’apprentissage et sauvegardons notre modèle [8].

2.1.1 Choix de l’algorithme

Dans le cours, nous avons vu diverses méthodologies, telles que le pseudo-labelling et le Mean Teacher. Plusieurs variantes inspirées de ces méthodes existent, notamment le Monte-Carlo Dropout pour le Mean Teacher. En ce qui concerne le pseudo-labelling, un exemple abordé dans le cours est Fix Match. Lors de la lecture de l’article ”FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence” [7], nous avons constaté que cette méthode propose une approche relativement simple à comprendre et semble offrir des résultats très intéressants pour la reconnaissance d’images, principalement pour la classification plutôt que la segmentation. Cependant, malgré sa simplicité apparente, cette approche intuitive suggère qu’elle pourrait également être appliquée au défi de la segmentation d’images.

Quelques résultats présentés dans l'article sont les suivants :

- **Quasiment pas supervisé (1 image choisie aléatoirement par classe)**
: Pour 4 entraînements, les résultats varient de 48,58% à 85,32% de précision, avec une médiane de 64,28%.
- **Quasiment pas supervisé (1 image caractéristique par classe)** :
Précision médiane de 78% avec un maximum de 84%.

Ci-dessous, nous comparons les résultats de Fix Match avec ceux d'autres modèles. Nous constatons que Fix Match obtient le meilleur résultat dans la majorité des cas ou un résultat très proche du meilleur.

Method	CIFAR-10			CIFAR-100			SVHN			STL-10
	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels	1000 labels
II-Model	-	54.26 \pm 3.97	14.01 \pm 0.38	-	57.25 \pm 0.48	37.88 \pm 0.11	-	18.96 \pm 1.92	7.54 \pm 0.36	26.23 \pm 0.82
Pseudo-Labeling	-	49.78 \pm 0.43	16.09 \pm 0.28	-	57.38 \pm 0.46	36.21 \pm 0.19	-	20.21 \pm 1.09	9.94 \pm 0.61	27.99 \pm 0.83
Mean Teacher	-	32.32 \pm 2.30	9.19 \pm 0.19	-	53.91 \pm 0.57	35.83 \pm 0.24	-	3.57 \pm 0.11	3.42 \pm 0.07	21.43 \pm 2.39
MixMatch	47.54 \pm 11.50	11.05 \pm 0.86	6.42 \pm 0.10	67.61 \pm 1.32	39.94 \pm 0.37	28.31 \pm 0.33	42.55 \pm 14.53	3.98 \pm 0.23	3.50 \pm 0.28	10.41 \pm 0.61
UDA	29.05 \pm 5.93	8.82 \pm 1.08	4.88 \pm 0.18	59.28 \pm 0.88	33.13 \pm 0.22	24.50 \pm 0.25	52.63 \pm 20.51	5.69 \pm 2.76	2.46 \pm 0.24	7.66 \pm 0.56
ReMixMatch	19.10 \pm 9.64	5.44 \pm 0.05	4.72 \pm 0.13	44.28 \pm 2.06	27.43 \pm 0.31	23.03 \pm 0.56	3.34 \pm 0.20	2.92 \pm 0.48	2.65 \pm 0.08	5.23 \pm 0.45
FixMatch (RA)	13.81 \pm 3.37	5.07 \pm 0.65	4.26 \pm 0.05	48.85 \pm 1.75	28.29 \pm 0.11	22.60 \pm 0.12	3.96 \pm 2.17	2.48 \pm 0.38	2.28 \pm 0.11	7.98 \pm 1.50
FixMatch (CTA)	11.39 \pm 3.35	5.07 \pm 0.33	4.31 \pm 0.15	49.95 \pm 3.01	28.64 \pm 0.24	23.18 \pm 0.11	7.65 \pm 7.65	2.64 \pm 0.64	2.36 \pm 0.19	5.17 \pm 0.63

Figure 1: FixMatch en comparaison avec les autres résultats

2.1.2 Fonctionnement de FixMatch

Cette méthode combine une approche avec un apprentissage supervisé et une régularisation. La régularisation (démonstration image ci-dessous) advient à chaque itération d'apprentissage supervisé. Comme dit précédemment, Fix Match a un fonctionnement intuitif, il se base sur le pseudo-labelling avec un léger correctif qui permet de grandement améliorer les résultats tout en évitant au maximum la propagation d'erreurs.

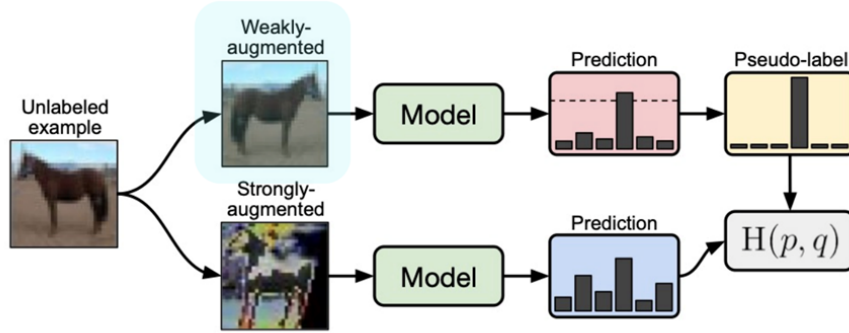


Figure 2: Diagramme du FixMtach

Pour mieux comprendre Fix Match, nous allons regarder l'image ci-dessus qui représente le pseudo-labelling. Pour cela, nous aurons besoin d'un set d'images labellisées et d'un set d'images non labellisées. Nous intégrons les images labellisées sans leurs labels au set d'images non labellisé. Ce processus est essentiel car il permettra un meilleur apprentissage de la part de notre modèle.

Pour la partie semi-supervisée, Fix Match fonctionne ainsi :

- Prendre une image non labellisée.
- Faire parcourir deux chemins à cette image :
 1. Appliquer peu d'augmentation sur l'image.
 2. Appliquer beaucoup d'augmentation sur l'image.
- Sur les deux chemins, obtenir les prédictions à partir du même modèle.
- Si la prédiction du chemin 1 est confiante, telle que $p > \tau$, avec τ la limite.
- Utiliser cette prédiction comme label (pseudo-label).
- Appliquer la fonction de coût.

C'est pour $p > \tau$ que l'ajout des images labellisées dans le set d'images non labellisées est important. De cette façon, nous sommes certains que le modèle commencera à se généraliser car il sera assez confiant dans au moins une partie des images.

Algorithm 1 FixMatch algorithm.

```

1: Input: Labeled batch  $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$ , unlabeled batch  $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$ ,
   confidence threshold  $\tau$ , unlabeled data ratio  $\mu$ , unlabeled loss weight  $\lambda_u$ .
2:  $\ell_s = \frac{1}{B} \sum_{b=1}^B H(p_b, \alpha(x_b))$  {Cross-entropy loss for labeled data}
3: for  $b = 1$  to  $\mu B$  do
4:    $q_b = p_m(y \mid \alpha(u_b); \theta)$  {Compute prediction after applying weak data augmentation of  $u_b$ }
5: end for
6:  $\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} H(\arg \max(q_b), p_m(y \mid \mathcal{A}(u_b)))$  {Cross-entropy loss with pseudo-label
   and confidence for unlabeled data}
7: return  $\ell_s + \lambda_u \ell_u$ 

```

Figure 3: Pseudo-code de l'algorithme FixMatch

2.1.3 Limitations de FixMatch

Fix Match est une méthode assez simple à mettre en place mais qui demande d'être très précis sur le choix des hyperparamètres pouvant influencer drastiquement les résultats obtenus par le modèle. Ici, le choix des augmentations n'est pas des plus pertinent concernant l'imagerie médicale. Dans un premier lieu nous faisons les transformations sur les images et non les Tensors demander une quantité importante de mémoire.

2.2 Augmentation des données

L'augmentation de données est une technique très populaire dans le domaine de l'apprentissage automatique. Elle permet d'augmenter la quantité de données pour l'entraînement du modèle. Dans notre cas, étant donné le faible volume de données supervisées, cela était nécessaire. En ce sens, nous avons choisi d'entraîner le modèle avec une image transformée une fois sur deux. De plus, nous avons choisi d'appliquer une seule transformation par image. Enfin, il y a 5 transformations et chacune a une chance égale de se réaliser. Vous trouverez ci-dessous une description des transformations.

1. **Flip (Miroir horizontal ou vertical)** : Une opération de retournement horizontal ou vertical est appliquée avec une probabilité de 0.5 pour introduire une variation spatiale dans l'ensemble de données.

2. **Rotation** : Une rotation aléatoire est appliquée avec une probabilité de 0.5 et un angle maximal de 30 degrés, diversifiant ainsi les angles de vue.

3. **Translation** : Une translation aléatoire est appliquée avec une probabilité de 0.5 et une amplitude maximale de 10 pixels, créant une diversité dans la position spatiale des objets.

4. **Ouverture (Opening)** : Une opération d'ouverture est appliquée à l'image avec une probabilité de 0.5 pour éliminer les petites régions indésirables et le bruit.

5. **Fermeture (Closing)** : Une opération de fermeture est appliquée à l'image avec une probabilité de 0.5 pour remplir les petites lacunes ou fissures et améliorer la cohérence de la segmentation.

2.3 Architecture des modèles

Voici les architectures des modèles que nous avons développés ou que nous avons considéré développer:

2.3.1 UNet

Le modèle, développé en 2015 par Olaf Ronneberger, Philipp Fischer, et Thomas Brox [5], possède plus de 2 millions de paramètres selon notre configuration. Il se compose d'un encodeur réduisant les dimensions de l'image via des couches convolutionnelles, suivi d'un décodeur utilisant des couches de suréchantillonnage et d'autres couches convolutionnelles.[2]

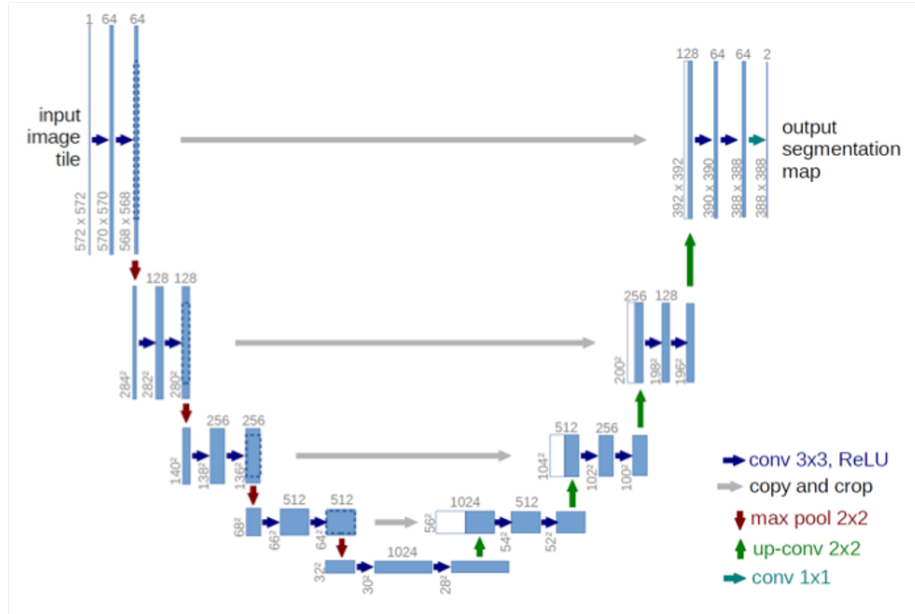


Figure 4: Architecture du modèle Unet

2.3.2 ResidualUNet

ResidualUNet, similaire au UNet, intègre des couches résiduelles qui le distinguent du modèle de base. Ces couches permettent au ResidualUNet d'apprendre les résidus entre l'entrée et la sortie de chaque couche, contribuant ainsi à résoudre le problème de la disparition des gradients pouvant survenir pendant l'entraînement, où les gradients des paramètres par rapport à la fonction de coût tendent vers zéro. Le ResidualUNet compte un peu plus de 2 millions de paramètres dans notre configuration. Il est à noter que nous avons utilisé l'implémentation de la bibliothèque Monai pour cette architecture.

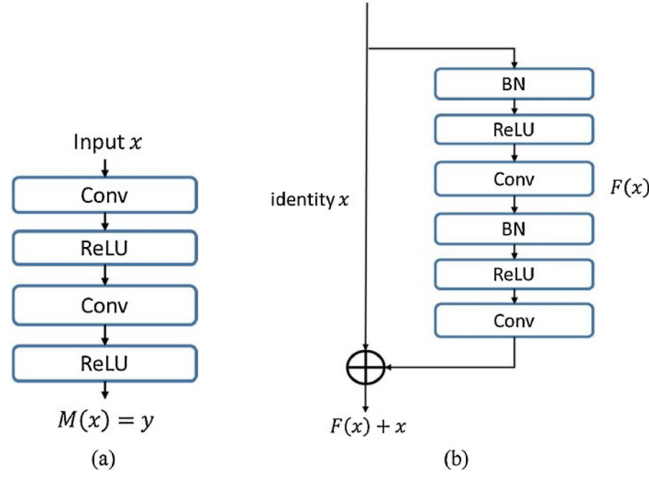


Figure 5: Architecture du modèle ResidualUNet

2.4 Hyperparamètres

Les hyperparamètres sont importants dans le processus d'apprentissage du modèle. Ils influencent de manière significative la performance globale du modèle. La recherche des combinaisons optimales nécessite de tester différentes valeurs. Cela représente un défi majeur, puisque nous avons expérimenté plusieurs options avec l'approche d'essai-erreur et que cela demande du temps. Enfin, la phase d'ajustement des hyperparamètres a été la plus exigeante en temps. Dans cette section, il sera question de la taille du lot, de la fonction de coût, du taux d'apprentissage, et de l'optimisateur.

2.4.1 Taille du batch

La taille du lot détermine le nombre d'échantillons d'image à utiliser par itération lors des phases d'entraînement et de validation. Le modèle est entraîné sur un lot de données à la fois. Par ailleurs, nous avons remarqué qu'une taille de lot plus grande accélère le processus d'entraînement, mais elle nécessite également une utilisation accrue de la mémoire de l'ordinateur. Ainsi, nous avons finalement opté pour une taille de lot de 4 à 8 images lors de l'entraînement et de 3 à 4 images lors de la validation.

2.4.2 L'évaluation de notre modèle d'apprentissage

La fonction de coût permet d'évaluer le modèle en mesurant l'écart entre les prédictions et les images réelles. Cela permet d'ajuster les poids du modèle durant l'entraînement (descente du gradient) et de minimiser l'erreur. Dans

notre cas, combiner l'Entropie-Croisée et le Dice Loss comme fonction de coût a été un avantage. Cette approche hybride améliore la capacité du modèle à bien s'ajuster aux tâches de segmentation, en considérant à la fois la correspondance pixel par pixel et la similarité de forme. [6] Pour la première fonction de coût (partie supervisée), nous utilisons 40% d'Entropie-Croisée

$$L_{CE}(y, \hat{y}) = -\frac{1}{n} \sum_i^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

et 60% de Dice Loss

$$L_{DSC}(y, \hat{y}) = 1 - 2 \frac{\sum_{p \in \Omega_i} y_p \hat{y}_p}{\sum_{p \in \Omega_i} y_p^2 + \sum_{p \in \Omega_i} \hat{y}_p^2}$$

Pour la partie semi-supervisée, nous n'utilisons que l'entropie-croisée. On a décidé d'attribuer une plus petite pondération pour l'entropie croisée que le diceloss parce que l'entropie croisée favorise la classe du background.

2.4.3 Taux d'apprentissage

Le taux d'apprentissage constitue un paramètre crucial en apprentissage machine, influant directement sur la progression du modèle lors de l'entraînement sur les images d'entraînement. Un taux trop élevé favorise une convergence rapide vers un minimum et requiert moins d'itérations. Cependant, il peut plus facilement dépasser le minimum global. À l'inverse, un taux trop petit induit une convergence plus lente, nécessitant davantage d'itérations pour atteindre le minimum, mais présente moins de risque de dépassement. Ainsi, le taux d'apprentissage doit être à la fois assez élevé pour converger rapidement, mais assez petit pour ne pas manquer le minimum optimal. Cette fois-ci, pour choisir la bonne valeur, nous avons consulté plusieurs articles sur le sujet tout en testant plusieurs valeurs par essai/erreur. Après avoir utilisé plusieurs techniques d'optimisation du taux d'apprentissage, nous avons abouti à un $lr = 0.001$.

2.4.4 Optimisateur

Les optimiseurs permettent d'améliorer la convergence vers des minimums globaux en évitant de rester bloqué à un minimum local. Comme vu en classe, il existe plusieurs optimiseurs, mais ils sont presque tous assez similaires en termes de performance. Ainsi, nous avons testé seulement deux optimiseurs. Le premier est SGD combiné avec Momentum. Nous avons fait cela dans l'optique d'avoir une meilleure généralisation grâce à SGD, mais cela prenait plus de temps à converger. Le deuxième était Adam. Nous avons remarqué que Adam donnait de meilleurs résultats en général. Enfin, notons que nous avons utilisé différents types de Scheduler, mais on obtenait de meilleurs résultats sans.

3 Résultats

Dans cette section, nous nous penchons sur les résultats de notre projet, un examen complet des techniques de pointe que nous avons mises en œuvre dans la segmentation des images médicales. Nous analyserons les performances de nos modèles, disséquons l’impact de l’apprentissage semi-supervisé et évaluerons l’efficacité de nos stratégies d’augmentation des données.

3.1 Aperçu des performances du modèle

Pendant l’entraînement, nous avons testé les modèles UNet et ResidualUNet (ce lui du cadriciel Monai) avec différentes configurations. Le tableau suivant montre les Dice obtenus pour chaque modèle et chaque classe:

<i>Modèle</i>	Classe 1	Classe 2	Classe 3	Moyenne
Unet (3M params)	0.557	0.755	0.847	0.720
Unet (1M params)	0.548	0.706	0.830	0.687
ResidualUNet (Monai)	0.566	0.757	0.846	0.723

Table 1: Tableau des résultats dice obtenus sur les images de tests

D’après le tableau, on observe que les modèles ResidualUNet et UNet ont des performances comparables sur les trois classes, avec une moyenne légèrement supérieure pour le UNet. La deuxième version du ResidualUNet montre une baisse de performance dans toutes les classes, particulièrement dans la classe 2. La performance constante du modèle UNet, surtout dans les classes 2 et 3, suggère qu’il pourrait être plus fiable pour cet ensemble de tests. Tous les modèles présentent néanmoins une performance élevée dans la classe 3, ce qui pourrait indiquer que cette classe est plus facile à segmenter ou que les modèles sont particulièrement bien ajustés à ses caractéristiques. Le tableau suivant montre les mesures comparatives des performances des modèles de segmentation d’images.

<i>Nom du model</i>	Train Loss	Validation Loss	Dist. Hausdorff
Unet (3M params)	0.028	0.078	6.081
Unet (1M params) (2)	0.038	0.089	8.872
ResidualUNet (Monai)	0.035	0.074	2.252

Table 2: Mesures comparatives des performances des modèles de segmentation d’images

Le modèle Unet se distingue par la perte d’entrainement la plus faible, ce qui indique une adaptation solide aux données. Il présente des caractéristiques avancées telles que l’average pooling, les blocs de double convolution et l’initialisation Kaiming, ce qui laisse entrevoir une architecture sophistiquée, adaptée pour la

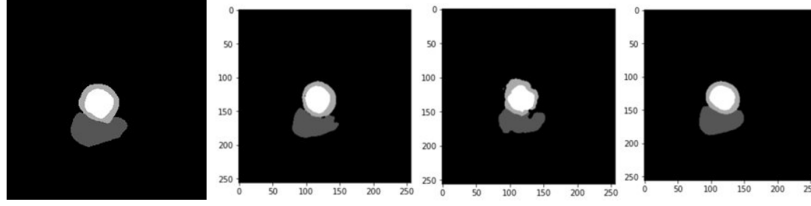


Figure 6: Comparaison de segmentation d’image médicale - l’image à gauche représente la vérité terrain, suivie des résultats de segmentation par les modèles UNet, Unet (2) et UNet (de gauche à droite).

généralisation. Le modèle Unet (2), bien que de conception similaire au modèle 1, présente une légère augmentation de la perte de validation et de la perte de Dice, ce qui suggère un problème potentiel de surajustement en augmentant le nombre de paramètres. Le modèle ResidualUNet (Monai), qui incorpore les Stides et Unités résiduelles, présente un compromis intrigant avec une perte de validation la plus petite et une distance de Hausdorff significativement plus faible, démontrant sa précision dans la délimitation de structures anatomiques complexes.

3.2 Résultats et comparaison visuelle

En comparant les résultats de la segmentation avec la ground truth, voir la figure 6, le modèle ResidualUNet et UNet semblent fournir la correspondance la plus proche, avec une déviation minimale dans la forme et la taille de la zone segmentée. Le modèle UNet (2), quant à lui, présente une segmentation plus conservatrice, sous-estimant légèrement la zone mais conservant un contour proche de la vérité.

3.3 Impact de l’apprentissage semi-supervisé

La mise en œuvre de l’apprentissage semi-supervisé dans notre modèle a révélé des effets notables sur les performances. En utilisant une taille de lot conservatrice (**Batch Size = 4**) et un taux d’apprentissage modéré (**Learning rate = 0.001**), nous avons observé une divergence significative entre les modèles semi-supervisés et les modèles entièrement supervisés [1]. L’approche semi-supervisée, dans son itération initiale avant l’augmentation des données, a produit une **perte d’entraînement de 0,644** et une **perte de validation de 0,879**, soit une diminution de dix fois de la performance de validation par rapport à son homologue supervisé, ce qui indique la nécessité d’affiner le traitement des données non étiquetées. Des expériences ultérieures avec une légère augmentation des données, telles que des inversions et des rotations, n’ont pas donné lieu à des améliorations substantielles, ce qui suggère que si l’apprentissage semi-supervisé peut être économe en ressources, son efficacité dépend fortement de

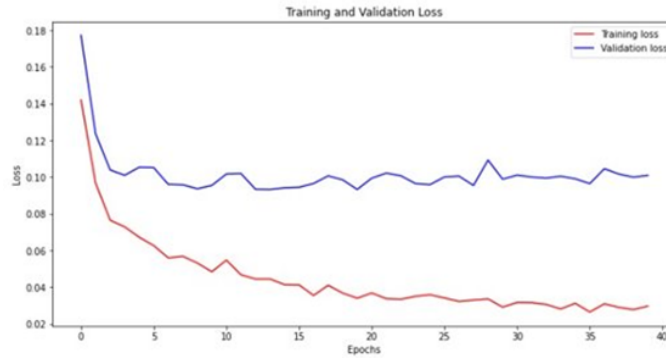


Figure 7: Loss de Training et Validation avant l'augmentation de données

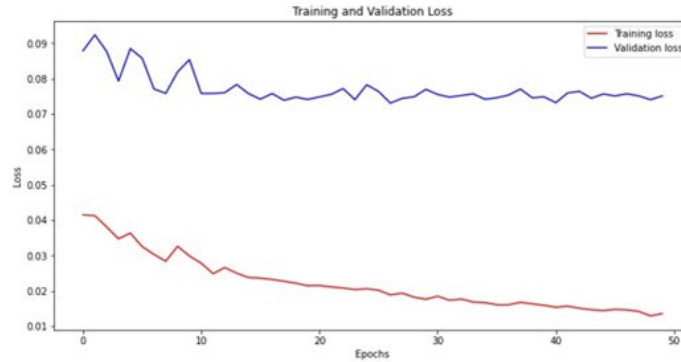


Figure 8: Loss de Training et Validation après l'augmentation de données

la complexité des techniques de transformation des données employées et de la capacité du modèle à se généraliser à partir de données moins étiquetées.

3.4 Efficacité de l'augmentation des données

Les résultats indiquent que l'augmentation des données a eu un effet positif sur le modèle ResidualUnet (Monai). Avant l'augmentation, les pertes d'entraînement et de validation diminuent régulièrement, mais la perte de validation plafonne, ce qui suggère que le modèle pourrait être surajusté par rapport aux données d'entraînement. Après l'augmentation, la perte d'entraînement connaît une légère augmentation, ce qui est courant lorsque le modèle apprend à partir d'un ensemble de données plus diversifié. Cependant, la perte de validation diminue et se stabilise à un niveau inférieur à celui d'avant l'augmentation, ce qui montre que la généralisation du modèle sur des données non vues s'est améliorée. Cela démontre que l'augmentation des données a permis d'atténuer l'overfitting et d'améliorer les performances prédictives du modèle.

4 Conclusion

En conclusion de notre recherche, les résultats obtenus mettent en évidence la puissance du modèle 3 (Monai), indépendamment des autres modèles. L'augmentation des données a renforcé ses performances, en témoigne la réduction de la distance de Hausdorff et l'amélioration de la perte de validation après augmentation. Ces avancées suggèrent que le modèle 3 est particulièrement efficace pour la segmentation d'images médicales, offrant une base solide pour des applications cliniques précises et fiables.

4.1 Pistes pour l'optimisation du Modèle

Pour améliorer davantage notre modèle, nous recommandons d'appliquer des augmentations de données courantes dans les contextes médicaux, de trier les images de manière que les tailles des lots correspondent aux images d'un même patient, et d'utiliser des métriques spécifiques comme le Dice Loss, la Distance de Hausdorff, et la Moyenne de Distance de Surface Symétrique (ASD) comme fonctions de coût pour améliorer les résultats de rendu du modèle. Il est conseillé de ne pas utiliser l'entropie croisée durant la phase semi-supervisée, mais plutôt des métriques telles que le Dice Loss, qui ne tiennent pas compte de la classe de fond, afin d'éviter la propagation d'erreurs et de permettre au modèle d'apprendre à partir des similarités de classe plutôt que des caractéristiques de fond.

References

- [1] Victor Ion Butoi, Jose Javier Gonzalez Ortiz, Tianyu Ma, Mert R. Sabuncu, John Guttag, and Adrian V. Dalca. Universeg: Universal medical image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21438–21451, October 2023.
- [2] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059. IEEE, 2020.
- [3] Shangzhu Jin, Sheng Yu, Jun Peng, Hongyi Wang, and Yan Zhao. A novel medical image segmentation approach by using multi-branch segmentation network based on local and global information synchronous learning. *Scientific Reports*, 13, 04 2023.
- [4] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images, 2023.
- [5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.
- [6] Pranav Singh, Luoyao Chen, Mei Chen, Jinqian Pan, Raviteja Chukkapalli, Shravan Chaudhari, and Jacopo Cirrone. Enhancing medical image segmentation: Optimizing cross-entropy weights and post-processing with autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2684–2693, October 2023.
- [7] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence, 2020.
- [8] Junde Wu, Jiayuan Zhu, Yuanpei Liu, Yueming Jin, and Min Xu. One-prompt to segment all medical images, 2023.
- [9] Ziyun Yang and Sina Farsiu. Directional connectivity-based segmentation of medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11525–11535, June 2023.