PARTICIPEZ A UNE COMPETITION KAGGLE

Benoit DA COSTA FERNANDES

27/05/2022

Table des matières

1. Introduction

Kaggle est une plateforme web qui accueille la plus grande communauté de Data Scientist au monde. Elle organise des compétitions en sciences des données. Sur cette plateforme, les entreprises proposent des problèmes en science des données et offrent un prix aux participants obtenant les meilleures performances.

2. Compétition

2-1. Contexte

En 2019, environ 5 millions de personnes ont été diagnostiquées d'un cancer du tractus gastro-intestinal dans le monde. Parmi ces patients, environ la moitié sont éligibles à la radiothérapie qui est généralement administrée entre 10 à 15 minutes par jour, pendant 1 à 6 semaines. Le traitement consiste à délivrer de fortes doses de rayonnement à l'aide de faisceaux de rayons X dirigés vers les tumeurs tout en évitant l'estomac et les intestins.

A l'aide de nouvelles technologies basée sur le couplage de l'imagerie par résonance magnétique et les systèmes d'accélérateur linéaire pour la radiothérapie (Magnetic Resonance Imaging Guided Linear Accelerator MRI-Linacs), les oncologues sont en mesure de visualiser la position quotidienne de la tumeur et des organes, qui peuvent varier d'un jour à l'autre. Afin d'éviter l'estomac et les intestins lors du traitement, les radio-oncologues doivent tracer manuellement la position de ces organes sur les scanners IRMs pour ajuster la direction des faisceaux de rayons X afin d'augmenter la dose administrée sur la tumeur. C'est un procédé long et laborieux qui peut prolonger les durées de traitement de 15 minutes à 1h par patient et ceci peut être difficile à tolérer pour les patients. Pour améliorer cela, l'apprentissage profond pourrait être utilisé afin de segmenter automatiquement l'estomac et les intestins et rendre les traitements beaucoup plus rapide et tolérable pour les patients et cela permettrait à un plus grand nombre de patients d'obtenir des traitements plus efficaces.

Le « Carbone Cancer Center » qui est situé dans l'Université de Wisconsin-Madison (UW-Madison), est un pionnier de la radiothérapie basée sur la technique du MRI-LINAC. Cette université publique a accepté de soutenir ce projet en proposant ce challenge sur la plateforme Kaggle et en livrant des scanners IRMs anonymisées des patients traités au UW-Madison Carbone Cancer Center.

Dans ce concours, Nous allons créer un modèle pour segmenter automatiquement l'estomac et les intestins sur les images IRMs. Les IRMs proviennent de vrais patients atteints de cancer qui ont subi 1 à 5 IRMs à des jours différents au cours de leur radiothérapie.

Les participants au challenge ont été invités, du 14 avril au 14 juillet 2022, à proposer des modèles de machine learning permettant de localiser automatiquement les intestins et l'estomac sur les clichés d'IRM. Pour évaluer les modèles le coefficient de Dice moyen et la distance de Hausdorff 3D ont été retenus.

2-2. Les données

Les organisateurs ont mis à disposition des participants un jeu de données comportant des clichés d'IRMs de vrais patients atteints de cancer. Ces clichés sont accompagnés d'annotations permettant de localiser sous formes de masques qui sont encodés avec le format RLE, les intestins et l'estomac.

Chaque cas de ce concours est représenté par plusieurs ensembles de tranches d'analyse (chaque ensemble est identifié par le jour où l'analyse a eu lieu).

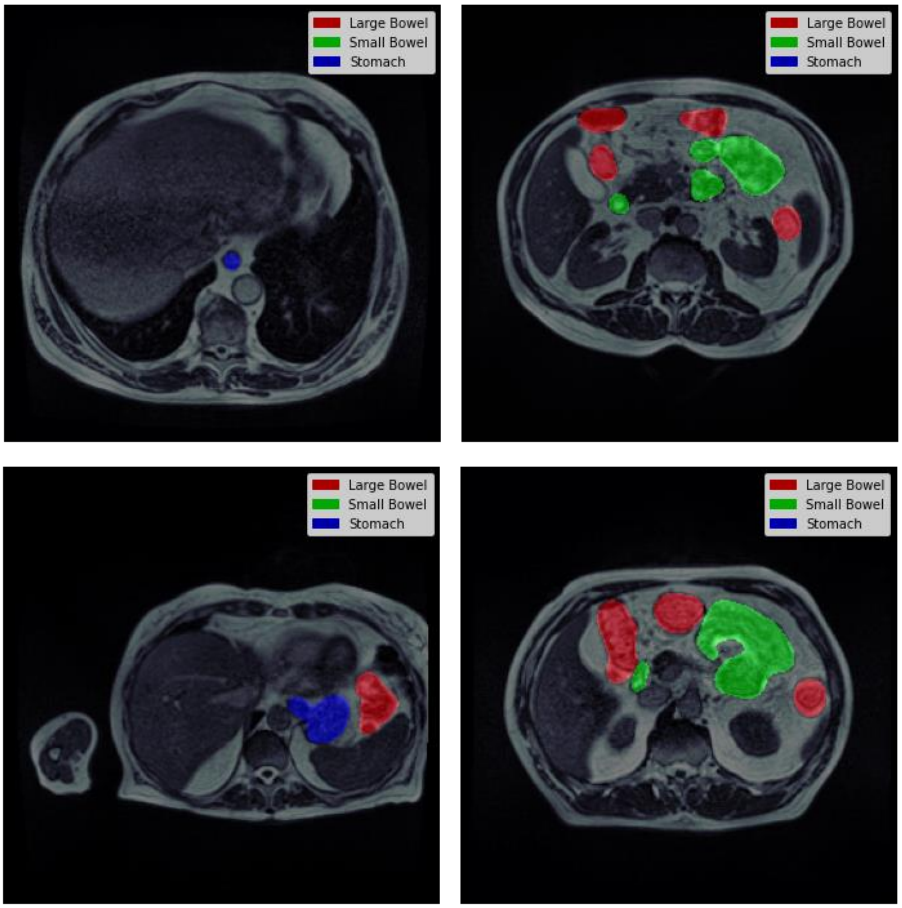

Figure 1 : Exemple de clichés IRMs du jeu d'entrainement avec l'application des masques pour localiser les intestins et l'estomac.

2-3. Contraintes

L'évaluation est réalisée à partir des notebooks soumis au concours. L'ensemble des opérations de ces derniers doivent pouvoir être réalisées pour un maximum de 9 heures d'utilisation CPU ou GPU, et avec un accès internet désactivé. L'utilisation de données libres et gratuites et de modèles pré-entrainés est autorisée.

3. Sources – Kernels sélectionnés

Pour débuter ce projet, je me suis appuyé sur les kernels uwmgi-unet-train-pytorch publié par « AWSAF » et uwmgi-segmentation-unet-keras-inference publié par « SAMUEL CORTINHAS ». Le premier étant le kernel ayant un des meilleurs scores au classement provisoire, à savoir 0.838. Ce score décrit la capacité du modèle à segmenter avec précision les différents organes d'intérêt (l'estomac et les intestins) sur les différents scanners IRMs. Il est calculé à partir du coefficient de Dice moyen et la distance de Hausdorff 3D qui sont combinées, avec un poids de 0,4 pour la métrique de Dice et de 0,6 pour la distance Hausdorff. Concernant le second kernel, aucun score n'a été attribué.

Ces deux kernels qui utilisent respectivement les bibliothèques « Pytorch » et « Keras « s'appuient sur l'architecture « U-NET » qui a été développé en premier lieu pour la segmentation d'images biomédicales. Cette architecture est basée sur réseau de neurones entièrement convolutifs. Il se

compose d'une partie de réduction, également appelé encodeur, et d'une partie expansive, appelé décodeur, qui est symétrique à la partie d'encodage, ce qui lui confère une architecture en forme de « U ».

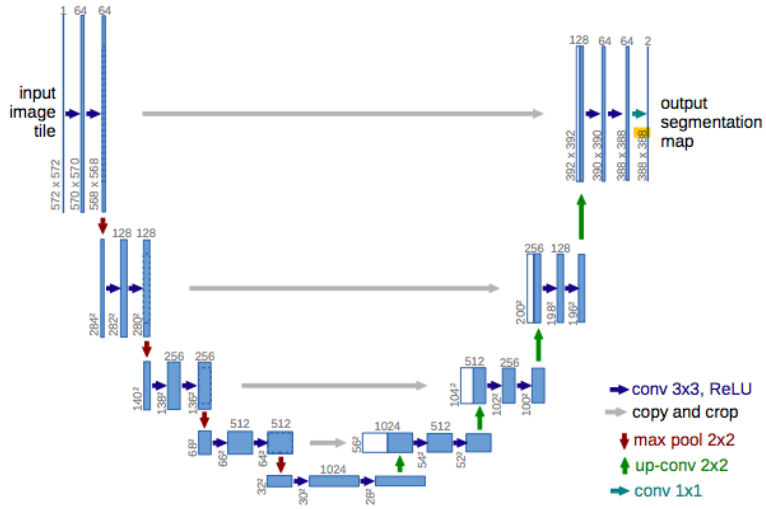

Figure 2 : Représentation schématique de l'architecture « U-NET »

La première partie de l'architecture est un réseau de neurones convolutifs typique où nous avons un empilement multiple de couches de convolutions et de MaxPooling permettant de réduire les informations spatiales tout en augmentant les informations sur les caractéristiques. Ceci permet, de ce fait, de capturer le contexte d'une image. La seconde partie permet de combiner les informations caractéristiques et spatiales à l'aide d'une séquence de convolutions transposées et de concaténations ascendantes issues de la voie contractante permettant de localiser plus précisément les objets.

4. Solutions apportées

4-1. Baseline

Dans un premier temps, suite aux diverses étapes de nettoyage et de mise en forme des données, j'ai implémenté une architecture U-NET avec la bibliothèque « Tensorflow ». J'ai ajouté des couches de « Dropout » et de « BatchNormalisation » qui sont des couches permettant de limiter le sur-apprentissage et de rendre les réseaux de neurones plus rapides et plus stables respectivement.

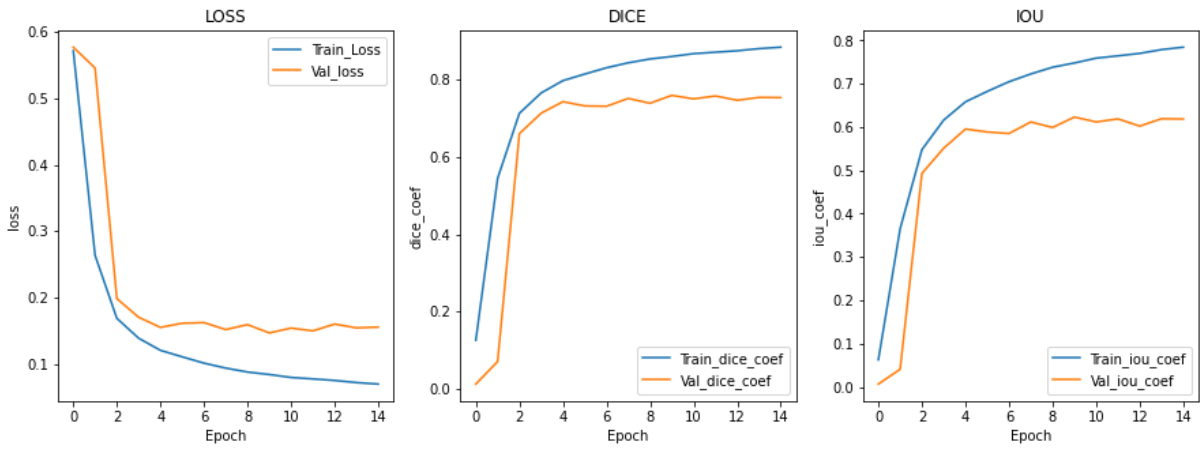

Figure 3 : Résultats obtenus lors de l'entrainement du modèle au niveau des pertes, du coefficient de Dice moyen et du score IoU (Intersection-Over-Union, Jaccard Index).

Ce modèle a ensuite été entrainé sur des images dont la résolution devait être de 128x128x3. De plus, nous avons testé la présence de couche de « Dropout » et de « BatchNormalisation ». Sur la Figure 3 sont présentés les résultats obtenus lors de l'entrainement du modèle où des couches de « Dropout » et de « BatchNormalisation » ont été utilisées. Nous obtenons pour le coefficient de Dice moyen un score de 0.88 pour le jeu d'entrainement et de 0.75 pour le jeu de validation. Le score obtenu suite à la soumission de ce code sur la plateforme Kaggle est de 0.72.

Ces résultats nous serviront de référence pour la suite, afin de comparer les diverses modifications apportées dans l'algorithmes.

4-2. Amélioration du modèle

4 2.1. Taille d'image d'entrée

Pour essayer d'améliorer les scores obtenus avec le modèle de référence pour cette étude (« Baseline »), j'ai commencé par augmenter la résolution des images d'entrée du modèle à (256x256x3) afin d'avoir une résolution d'images similaires à celle de la majorité des images brutes (Tableau 1) et limiter ainsi des pertes d'informations. De plus, j'ai ajouté une étape d'encodage et de décodage dans l'algorithme U-NET.

Dimension image	Nombre d'images
234x234	56
266x266	10955
276x276	718
360x310	4861

Tableau 1 : Nombre d'image en fonction de leur résolution.

Sur la Figure 4 sont présentés les résultats obtenus lors de l'entrainement du modèle. Nous obtenons pour le coefficient de Dice moyen un score de 0.88 pour le jeu d'entrainement et de 0.77 pour le jeu de validation. Le score obtenu suite à la soumission de ce code sur la plateforme Kaggle est de 0.748. Nous observons une légère amélioration du modèle.

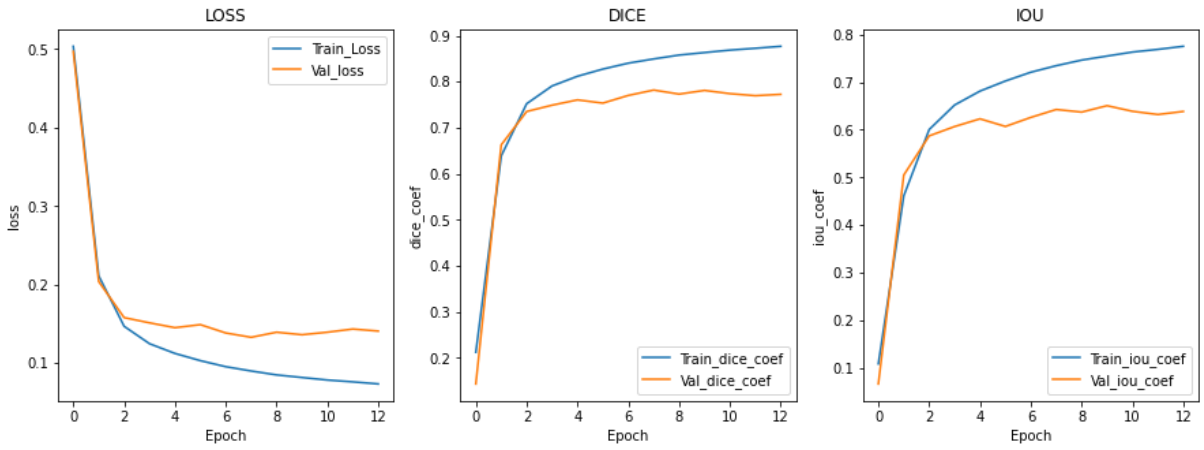

Figure 4 : Résultats obtenus lors de l'entrainement du modèle au niveau des pertes, du coefficient de Dice moyen et du score IoU (Intersection-Over-Union, Jaccard Index).

Sur le lot d'image que nous avons, nous avons un nombre d'images (41%) qui ne présentent pas la même résolution selon leurs deux axes. Lors de la phase de redimensionnement des images pour qu'elles aient toutes la même dimension lors de l'entrainement du modèle, les images ayant leurs

dimensions différentes en fonction de leur axe seront déformées. Une prise en compte de cette déformation permettrait d'améliorer le score du modèle.

4 2.2. Data Augmentation

Pour améliorer le modèle, j'ai également utilisé la méthode de « Data Augmentation ». Celle-ci permet, pour l'entrainement du modèle, d'accroitre artificiellement le nombre d'images à l'aide de diverses transformations (rotation, flip, …) et d'obtenir généralement, de meilleurs scores de prédiction.

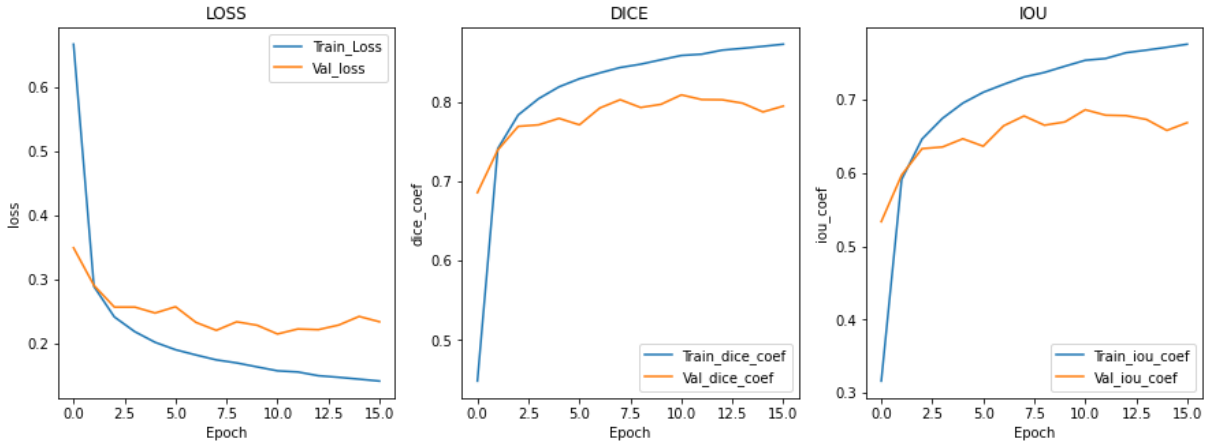

Figure 5 : Résultats obtenus lors de l'entrainement du modèle au niveau des pertes, du coefficient de Dice moyen et du score IoU (Intersection-Over-Union, Jaccard Index).

Sur la Figure 5 sont présentés les résultats obtenus lors de l'entrainement du modèle. Nous obtenons pour le coefficient de Dice moyen un score de 0.87 pour le jeu d'entrainement et de 0.795 pour le jeu de validation. Malheureusement, je n'ai pas pu obtenir de score sur la plateforme Kaggle suite à des problèmes de soumission de ce notebook.

4 2.3. Bilan

Sur le Tableau 2 sont résumés les différents scores obtenus des différents modèles testés. Le modèle 1 est le modèle servant de référence. Concernant le modèle 2, nous avons augmenté d'un facteur 2 la dimension des images en entrée du modèle et ajouté un bloc de couche de convolutions et de MaxPooling pour la partie encodage de l'architecture et un bloc de couche de convolutions transposés et de concaténation pour la partie décodage. Pour le modèle 3, nous avons implémenté la data augmentation afin d'augmenter artificiellement le nombre d'image pour l'entrainement du modèle.

	Score Dice (train - val)	Score Kaggle
Modèle 1 (baseline)	0.88 – 0.72	0.72
Modèle 2 (size inputs)	0.88 – 0.77	0.748
Modèle 3 (data augmentation)	0.87 – 0.795	

Tableau 2 : Bilan des différents scores obtenus sur les différents modèles testés.

5. Conclusions

Ce projet porte sur la participation à une compétition Kaggle qui traite de la segmentation automatique des organes tels que l'estomac et les intestins sur des scanners IRMs à l'aide d'algorithme de « Deep Learning ».

Pour débuter ce projet, je me suis appuyé sur différents notebooks publiés par différents compétiteurs

sur la plateforme Kaggle qui m'a permis d'effectuer les différentes étapes de nettoyage et de mise en forme des données plus rapidement. J'ai ensuite implémenté un modèle de segmentation d'images à partir de l'architecture « U-NET » qui a été développé pour la segmentation d'images médicales. Dans un premier temps, j'ai testé l'ajout de couche de « Dropout » et de « BatchNormalisation » et leur présence dans l'architecture a permis d'obtenir le meilleur résultat pour le coefficient de Dice moyen qui est une partie du score utilisé pour la compétition Kaggle (Tableau 3). Dans un second temps, pour essayer d'améliorer les scores obtenus avec le modèle de référence, j'ai augmenté la résolution des images d'entrée du modèle d'un facteur 2 et ajouté une couche d'encodage et de décodage dans l'architecture. Cette modification a permis d'améliorer légèrement les résultats (Tableau 3). Dans un troisième temps, j'ai utilisé la méthode de « Data Augmentation » afin d'augmenter artificiellement le nombre d'image et d'améliorer l'entrainement du modèle. Ceci nous a permis d'obtenir à nouveau de meilleurs résultats au niveau du score de Dice (Tableau 3) et probablement au niveau du score mesuré sur la plateforme Kaggle.

	Score Dice (train - val)	Score Kaggle
Modèle 1 (baseline)	0.88 – 0.72	0.72
Modèle 2 (size inputs)	0.88 – 0.77	0.748
Modèle 3 (data augmentation)	0.87 – 0.795	

Tableau 3 : Bilan des différents scores obtenus sur les différents modèles testés.

Cette première phase de test pour améliorer les scores obtenus m'a permis d'identifier des axes d'amélioration possible. En effet, sur le lot d'images mise à disposition, nous avons de nombreuses images ayant des résolutions différentes selon leurs axes. Ainsi lors de la l'étape de redimensionnement des images pour l'entrainement du modèle, ces images subissent une déformation entrainant également une déformation des zones d'intérêt. Une prise en compte de cette déformation permettrait d'améliorer le score du modèle.

6. Liens

Modèle 1 [train] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-train-01

Modèle 1 [infer] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-infer-01

Modèle 2 [train] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-infer-01

Modèle 2 [infer] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-infer-02

Modèle 3 [train] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-train-03

Modèle 3 [infer] : https://www.kaggle.com/code/benoitdacosta/uwmgtis-keras-infer-03