**Préparation Echange – Preprocessing 3D MRI – INSA Lyon**

preprocess\_niigz\_images.py

Ce script est une chaîne de traitement pour le preprocessing des ensembles de données d'imagerie médicale au format nifti. Il se compose de trois classes principales :

1. **ImagePaths** : Automatise la récupération des chemins de fichiers nii.gz à partir d'un répertoire spécifié, principalement pour organiser les données d'entrée.
2. **StatImageShape** : Analyse les statistiques dimensionnelles de l'ensemble de données. Elle calcule la forme minimale, médiane, moyenne, maximale et une forme ‘balanced’ pondérée des images. Permet de comprendre la variabilité de dimension de l'ensemble de données et pour décider d'une forme cible pour un redimensionnement uniforme.
3. **DatasetTransformer** : Applique une série de transformations TorchIO pour standardiser l'ensemble de données.
   1. **Chargement des Sujets (Subjects)** :
      1. Utilise une liste de chemins d'images pour créer des objets Subject de TorchIO.
      2. Chaque Subject encapsule une image MRI en tant qu'ScalarImage, facilitant les transformations ultérieures.
   2. **Application des Transformations** :
      1. Définit une séquence de transformations TorchIO à appliquer sur les sujets. Ces transformations incluent :
         1. Resample : Uniformise l'échelle de voxel entre les sujets.
         2. Resize : Ajuste les sujets à une forme cible (balanced\_shape), en utilisant l'interpolation B-spline pour maintenir la qualité de l'image.
         3. RescaleIntensity : Normalise les intensités des images pour un intervalle commun, améliorant la cohérence des données.
         4. ZNormalization : Normalise chaque image pour avoir une moyenne de zéro et une variance de un, ce qui est une pratique courante dans le traitement des images médicales pour réduire les variations entre les patients.
      2. Les transformations sont combinées dans un pipeline à l'aide de Compose, et appliquées en créant un SubjectsDataset TorchIO avec les sujets et le pipeline de transformation.

Le script se termine par l'enregistrement des images transformées dans un répertoire de sortie spécifié, en maintenant une référence à leurs noms de fichiers originaux pour la traçabilité. Le pipeline est initié dans le bloc principal, où il traite un répertoire spécifié d'images NIfTI, calcule leurs statistiques de forme, applique des transformations et sauvegarde la sortie pour une analyse ultérieure.

Pour les masks, je mets en commentaire l’étape RescaleIntensity et ZNormalization.

nifti\_reslice\_automatic.py

Ce script en Python est conçu pour traiter des images médicales au format NIfTI. Il se concentre sur le recalage d'images, c'est-à-dire l'ajustement d'un ensemble d'images (définies comme "moving") pour qu'elles correspondent à l'espace d'un autre ensemble d'images (définies comme "fixed"). Voici une explication détaillée :

1. **Fonction niftireslice**:
   * **Objectif**: Recaler une image NIfTI (moving) dans l'espace d'une autre image NIfTI (fixed).
   * **Paramètres**:
     + fixed\_nii\_path: Chemin vers l'image de référence.
     + moving\_nii\_path: Chemin de l'image à recaler.
     + moving\_resliced\_nii\_path: Chemin pour sauvegarder l'image recalée.
   * **Processus**:
     + Les images 'fixed' et 'moving' sont chargées.
     + L'image 'moving' est recalée pour correspondre à l'espace de l'image 'fixed' en utilisant resample\_from\_to, qui ajuste les dimensions et l'orientation.
     + La nouvelle image recalée est sauvegardée au chemin spécifié.
2. **Bloc principal**:
   * Définit les chemins des dossiers pour les images 'SWI' (fixed) et 'TOF3D' (moving), ainsi que le dossier de sortie pour les images recalées.
   * Parcourt les images dans les dossiers SWI et TOF3D, et pour chaque paire d'images, appelle la fonction niftireslice pour recalibrer l'image TOF3D par rapport à l'image SWI correspondante.
   * Les images recalées sont sauvegardées dans le dossier spécifié avec un suffixe '\_resliced'.