MRI Preprocessing and Data Augmentation Pipeline for Alzheimer's Disease Prediction

Preprocessing Pipeline

Preprocessing serves to standardize the dataset, reduce noise, and isolate relevant brain structures. The steps below detail the standard practices adopted in neuroimaging research.

1 Skull Stripping (Brain Extraction)

Skull stripping removes non-brain tissues such as the scalp, skull, and dura from the MRI image. This step is essential to reduce irrelevant features and computational overhead while focusing model attention solely on the brain.

- **Tool Used**: hd-bet library which identifies and returns only the extracted brain from the input .nii files

Non-brain structure and dark regions, such as the cerebrospinal fluid (the liquid between the skull and the brain) add noise and can mislead deep learning models, especially CNNs, into learning irrelevant patterns.

2 Bias Field Correction

MRI scanners (especially old machines) often introduce low-frequency intensity variations called bias fields, which creates noise in the scans.

- **Tool Used**: N4BiasFieldCorrection algorithm of the SimpleITK library

This ensures that the model learns genuine features instead of scanner-induced artifacts.

3 Normalization

Intensity values across MRIs can differ due to scanner settings. Normalizing voxel intensities improves model robustness.

- **Technique Used**: Z-score normalization: `(x - mean) / std`

Standardizes the range of intensity values, aiding in more stable and faster model training.

4. Data Augmentation Pipeline

Data augmentation enhances generalizability by artificially expanding the dataset through realistic transformations.

- **Tool Used**: TorchIO, a medical imaging augmentation library built for PyTorch.

Augmentation Techniques:

- **RandomFlip**: Increases rotational invariance.
- **RandomAffine**: Simulates scanner and positional variance.
- **Elastic Deformation**: Models anatomical variability.
- **Noise and BiasField**: Increases robustness to real-world distortions.
- **CropOrPad**: Ensures consistent input size for 3D CNNs.