Literature Review on Preprocessing Pipelines for Brain MRI Scans

Aayush Gajeshwar
Dabeet Das
Ram Daftari

June 24, 2025

1 Preprocessing Pipeline

The MRI preprocessing pipeline includes motion correction, Bias field correction, Skull stripping, Affine registration, Intensity normalization. Aside from motion correction, other techniques are done by nppy - a python library based on model NPP, which uses deep learning network.

1.1 1. Motion Correction

Head Movement during MRI scans can lead to blurry, misaligned images. ANTsPy and FSL FLIRT are two available tools for motion correction. For fMRI data, FSL MCFLIRT tool can be used. ANTsPy and FSL FLIRT are compared in Table 1.

Feature	**ANTsPy**	**FSL FLIRT**
Tool Type	Python interface to ANTs	Command-line tool (used via `subprocess` or Nipype)
Accuracy	High – excellent anatomical alignment	Moderate – good for fast rigid/affine alignment
Speed	Slower (precision-focused)	Fast (efficient rigid/affine transforms)
Transformation Handling	Full transform object (apply, invert, compose, etc.)	Saves `.mat` affine matrices

Table 1: Comparison of ANTsPy versus FSL FLIRT for structural MRI registration

1.2 2. Bias Field Correction

Bias Field Correction corrects for smooth, low-frequency variations in intensity across the image. It ensures that tissue types (e.g., gray matter) have consistent intensity values.

1.3 3. Skull Stripping

Skull Stripping removes non-brain tissues (e.g., skull, scalp, fat, eyes) from structural MRI scans, leaving only the brain region. Skull Stripping ensures that unnecessary information does damage the accuracy of our model. SynthStrip and NPP are two tools available for skull stripping which use deep learning techniques.

Feature	**NPP**	**SynthStrip**
Approach	Lightweight deep learning model	Deep learning on synthetic brain MRI data
GPU Runtime	~ 6 s	~ 55 s
Usability	Command-line or by using `subprocess`	Command-line or by using `subprocess`
Python Library	`nppy`	`nipreps-synthstrip`

Table 2: Comparison of NPP and SynthStrip for skull-stripping in brain MRI

1.4 4. Spatial Normalization

Spatial normalization is a key preprocessing step in neuroimaging that transforms individual brain images into a standard anatomical space, such as MNI (Montreal Neurological Institute) or Talairach space.

1.5 5. Intensity Normalization

Intensity normalization in MRI is the process of adjusting the voxel intensity values so that images become comparable across scans, subjects, or sessions. The NPP model uses Z-score normalization technique for intensity normalization.

$$I_{\text{norm}} = \frac{I - \mu_{\text{brain}}}{\sigma_{\text{brain}}}$$

2 Another Pipeline using Wavelet Transforms

Wavelet transforms are powerful tools for signal and image processing, particularly effective for analyzing non-stationary signals like MRI data. Unlike the Fourier transform, which provides only frequency information, wavelet transforms offer both spatial (or temporal) and frequency localization, making them suitable for medical imaging tasks where structural detail and frequency-based texture are both important.

2.1 Overview

Wavelet transforms decompose signals into components at multiple scales. They use small, wave-like basis functions (wavelets) that are localized in both time and frequency.

There are two main types:

- Continuous Wavelet Transform (CWT): Offers high resolution but is computationally intensive.
- Discrete Wavelet Transform (DWT): Uses dyadic scales and shifts (powers of 2), making it more efficient and suitable for digital applications like MRI image analysis.

2.2 Discrete Wavelet Transform (DWT)

DWT decomposes an image into approximation (low-frequency) and detail (high-frequency) components at multiple levels. It is typically implemented via filter banks:

- Low-pass filter (L): Captures the coarse, general structures.
- High-pass filter (H): Captures the fine details like edges and textures.

For a 2D image like an MRI scan, DWT generates:

- LL: Approximation (low-low)
- LH, HL, HH: Detail coefficients in vertical, horizontal, and diagonal directions respectively.

This hierarchical decomposition can be repeated on the LL component for multi-level analysis.

2.3 Why Use DWT for MRI Preprocessing in AD Detection?

Alzheimer's Disease (AD) causes structural and textural changes in brain MRI scans, particularly in the hippocampus and surrounding regions. The Discrete Wavelet Transform (DWT) is useful for preprocessing sMRI data due to the following reasons:

- **Noise Reduction:** DWT helps filter out high-frequency noise while retaining meaningful features, thereby improving the overall signal quality.
- **Compression:** By discarding less significant wavelet coefficients, DWT effectively reduces the dimensionality of the data, making downstream analysis more efficient.
- **Feature Extraction:** DWT isolates relevant texture and structural features that may indicate atrophy or other tissue changes associated with AD.
- **Multiscale Analysis:** Different frequency bands obtained through DWT enable the detection of changes at various spatial scales, which is especially helpful for identifying subtle early signs of AD.

2.4 Practical Use Cases

- **Preprocessing Pipeline:** Apply DWT to MRI scans before feeding them into machine learning models such as Convolutional Neural Networks (CNNs) or Support Vector Machines (SVMs).
- **Feature Engineering:** Extract wavelet coefficients and compute statistical descriptors such as mean, standard deviation, and entropy to form meaningful features for classification.
- **Dimensionality Reduction:** Retain only the approximation coefficients (LL) or select top-level detail coefficients to reduce the input feature size, thereby decreasing computational load while preserving important information.

2.5 Tools & Libraries

- **Python:** `PyWavelets (pywt)` is a widely used library for performing Discrete Wavelet Transforms in Python. It supports 1D, 2D, and 3D wavelet analysis and is suitable for preprocessing sMRI data.
- **MATLAB:** MATLAB provides built-in support for wavelet transforms through its Wavelet Toolbox, which includes extensive functionality for visualization, denoising, and multilevel decomposition.
- **Integration with Deep Learning:** DWT can be used in conjunction with convolutional neural networks (CNNs) either by preprocessing the input data or by integrating wavelet-based filters into the network architecture to enhance feature extraction.