

CS-GY 9223 / CS-UY 3943 - Neuroinformatics, Spring 2026

Project 1: Low-Field to High-Field MRI Enhancement

Important Links & Deadline

Kaggle Competition: [Join Competition \(Kaggle Link\)](#)

Team Sign-Up Sheet: [Sign Up Here \(Google Sheets\)](#)

Code Submission: Upload zipped code to **Brightspace**

Deadline: **February 13th, 2025 at 11:59 PM**
(Midnight before February 14th)

Note: You are allowed to use pre-built libraries (PyTorch, TensorFlow, scikit-image, etc.) to implement your solution. You are free to use any pretrained models or additional datasets as well.

Submission Instructions:

- Submit a **final zipped code file** as your project submission on Brightspace.
- Use **Markdown cells** in notebook or neatly comment your code to include explanations wherever needed.
- You are allowed to use **LLMs**. If you do:
 - Include the **chat history** you had with the LLM along with your submission.
 - This helps us understand your **thinking style**.
- Participate in the **Kaggle competition** and submit your `submission.csv` file there.
- **Participation in the Kaggle is mandatory, and the rankings on the private leader-board on Kaggle will be used for final scores.**

Competition At A Glance:

Item	Details
Training Data	18 paired low-field / high-field MRI volumes
Test Data	5 low-field volumes (high-field targets hidden)
Evaluation	SSIM + PSNR on all 200 axial slices per sample (1000 total)
Baseline Score	≈ 0.46 (bicubic interpolation)

Quick Start:

1. Sign up your team on the [Team Sign-Up Sheet](#)
2. Join the [Kaggle competition](#) and download the dataset
3. Train your model on 18 paired examples
4. Generate enhanced volumes for 5 test samples (output shape: $179 \times 221 \times 200$)
5. Use `extract_slices.py` to create `submission.csv` and submit to Kaggle
6. Submit your code/notebook to Brightspace before the deadline

Contents

1 Problem Statement	4
1.1 Background	4
1.2 Challenge Goal	4
1.3 Clinical Relevance	4
2 Data Description	5
2.1 Dataset Overview	5
2.2 Volume Specifications	5
2.3 Data Registration	5
2.4 File Structure	7
2.5 Row ID Format	7
3 Evaluation Metrics	7
3.1 Structural Similarity Index (SSIM)	7
3.2 Peak Signal-to-Noise Ratio (PSNR)	8
3.3 Combined Scoring Formula	8
3.4 Public/Private Leaderboard Split	9
3.5 Baseline Performance	9
4 Submission Instructions	9
4.1 Required Output	9
4.2 CSV Format	9
4.3 Using the Extraction Script	10
4.4 Important Requirements	10
5 Suggested Approaches	11
5.1 Problem Formulation	11
5.2 Baseline Methods	11
5.3 Deep Learning Approaches	11
5.4 Training Considerations	11
6 Glossary	12

1 Problem Statement

Given paired 64mT (portable) and 3T (high-field) brain MRI volumes, learn an algorithm that enhances the 64mT images to better match the 3T reference while preserving clinically relevant structure (edges, contrast, anatomy).

1.1 Background

Magnetic Resonance Imaging (MRI) is a fundamental tool in neuroimaging, providing detailed anatomical and functional information about the brain. High-field MRI scanners (e.g., 3 Tesla) produce excellent image quality with high signal-to-noise ratio (SNR) and fine spatial resolution. However, these systems are expensive to purchase and operate, require specialized infrastructure (shielded rooms, helium cooling), and are not equally available in all healthcare settings.

Low-field and ultra-low-field MRI systems (e.g., 64 milliTesla) offer compelling advantages:

- **Lower cost:** Significantly reduced capital and operating expenses
- **Easier deployment:** Smaller footprint, less infrastructure requirements
- **Improved accessibility:** Can enable MRI in resource-constrained settings
- **Potential portability:** Some designs allow bedside or mobile imaging

The trade-off is reduced image quality: lower SNR, decreased sharpness, and contrast differences compared to high-field acquisitions. This quality gap can limit both visual interpretation and downstream quantitative analysis.

1.2 Challenge Goal

This competition addresses a practical research question:

Can we algorithmically enhance low-field brain MRI to better match high-field image quality?

Task: Given a **low-field (64mT) T1-weighted MRI volume**, generate an **enhanced 3D image** that matches the corresponding **high-field (3T) MRI** as closely as possible.

1.3 Clinical Relevance

A reliable enhancement model could:

- Improve usability of low-field MRI in resource-constrained settings
- Increase robustness of downstream analysis pipelines (segmentation, registration)
- Reduce the “quality gap” while retaining practical advantages of low-field acquisition
- Enable better diagnostic confidence from portable or low-cost scanners

Important Disclaimer: This is a research and educational task. Model outputs are **not** medical advice and should not be used for clinical diagnosis or treatment decisions.

2 Data Description

2.1 Dataset Overview

The dataset contains paired low-field (64mT) and high-field (3T) T1-weighted brain MRI scans from healthy adult subjects. All volumes have been spatially registered (aligned) for fair comparison.

Table 1: Dataset Summary

Split	Subjects	Slices	Low-Field Input	High-Field Target
Train	18	3,600	Available	Available
Test	5	1,000	Available	Hidden

2.2 Volume Specifications

Table 2: MRI Volume Properties

Property	Low-Field (64mT)	High-Field (3T)
Volume Shape	$112 \times 138 \times 40$	$179 \times 221 \times 200$
Voxel Size	$1.6 \times 1.6 \times 5.0 \text{ mm}$	$1.0 \times 1.0 \times 1.0 \text{ mm}$
Number of Axial Slices	40	200
Each Axial Slice Shape	112×138	179×221

The low-field volume has anisotropic voxels (thick slices in z-direction) while the high-field volume has isotropic 1mm resolution. Your model must upsample and enhance the low-field input to produce a volume matching the high-field target dimensions.

2.3 Data Registration

To enable fair pixel-wise comparison:

- The **low-field volume is the original acquisition** (untouched)
- The **high-field volume has been registered** to the low-field coordinate space
- Registration uses rigid transformation to preserve anatomical accuracy

Figure 1 shows the registration quality through checkerboard overlays.

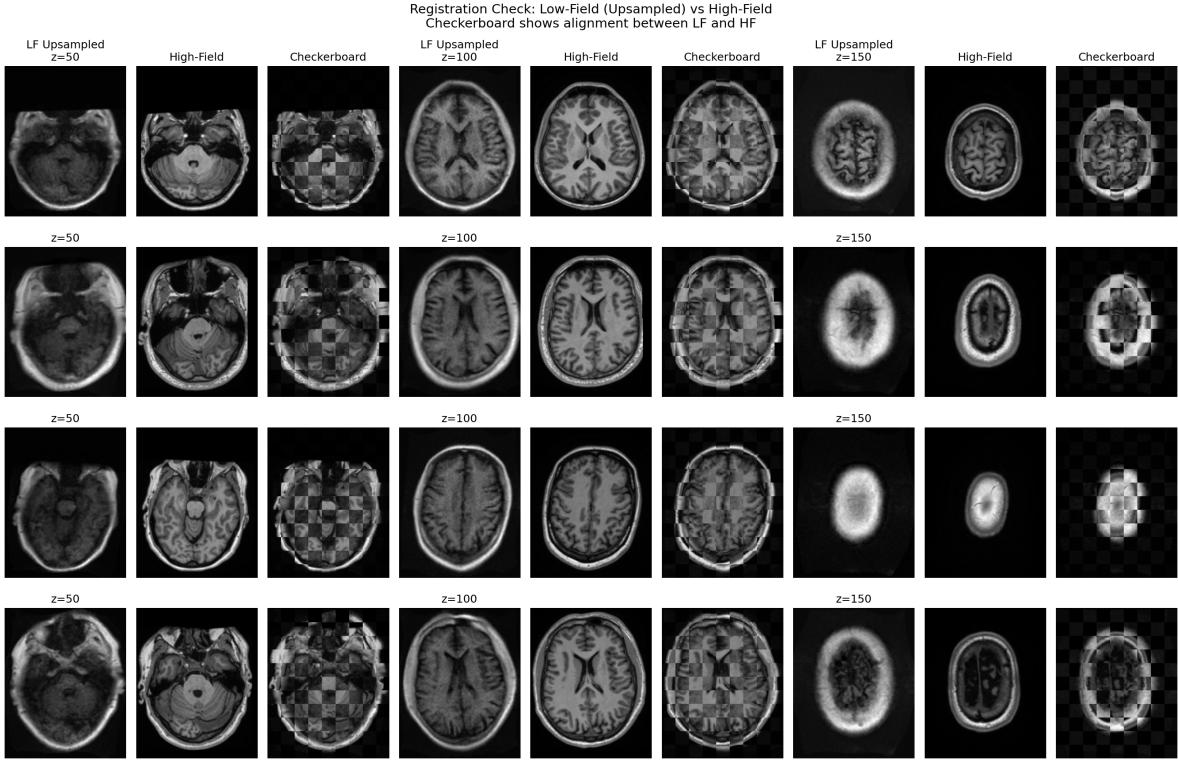


Figure 1: Registration quality check: Low-field images (bicubic upsampled) compared with high-field targets across multiple samples and slice positions. The checkerboard pattern demonstrates anatomical alignment between the two modalities.

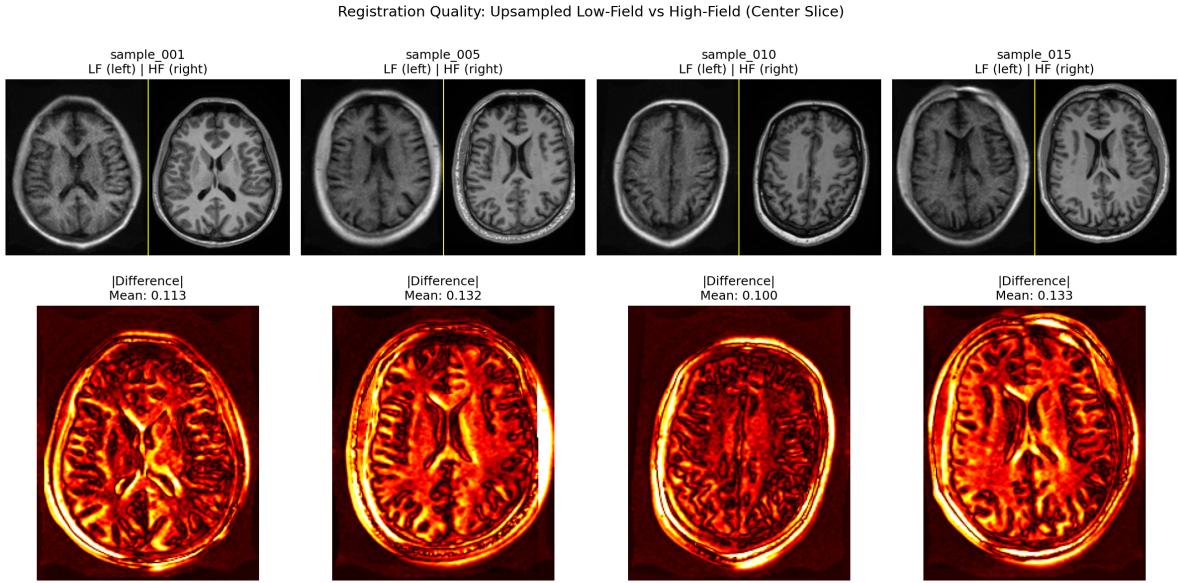


Figure 2: Registration quality: Side-by-side comparison (top row) and absolute difference maps (bottom row) for four training samples. The difference maps show that registration errors are minimal, with most differences arising from the enhancement challenge itself (SNR, sharpness, contrast).

2.4 File Structure

Listing 1: Dataset File Structure

```
mri.superres.dataset/
    train/
        low.field/
            sample.001.lowfield.nii.gz
            sample.002.lowfield.nii.gz
            ...
            (18 files)
        high.field/
            sample.001.highfield.nii.gz
            sample.002.highfield.nii.gz
            ...
            (18 files)
    test/
        low.field/
            sample.019.lowfield.nii.gz
            sample.020.lowfield.nii.gz
            ...
            (5 files)
    train.csv          # Ground truth (3600 rows)
    sample.submission.csv # Baseline submission (1000 rows)
    extract.slices.py   # Utility script
```

2.5 Row ID Format

Each row in the CSV files represents a single axial slice:

$$\text{row_id} = \text{sample_XXX_slice_YYY}$$

- XXX: Sample number (001–018 for train, 019–023 for test)
- YYY: Slice index (000–199)

Examples: `sample.001.slice.100` (center slice of first training sample), `sample.019.slice.050` (slice 50 of first test sample).

3 Evaluation Metrics

Submissions are evaluated by comparing predicted slices to the hidden high-field ground truth across **all 200 axial slices per sample** (1000 total slices for 5 test samples).

3.1 Structural Similarity Index (SSIM)

SSIM measures perceptual similarity by comparing luminance, contrast, and structure between two images. It is widely used in image quality assessment.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1)$$

Where:

- μ_x, μ_y = mean intensity of images x and y
- σ_x^2, σ_y^2 = variance of images x and y
- σ_{xy} = covariance between x and y
- $C_1 = (0.01)^2, C_2 = (0.03)^2$ = stability constants

Range: [0, 1] where 1 indicates identical images (higher is better).

3.2 Peak Signal-to-Noise Ratio (PSNR)

PSNR measures reconstruction fidelity based on the mean squared error (MSE) between the prediction and ground truth.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (2)$$

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}} \right) \quad (3)$$

Where:

- N = total number of pixels in the image
- x_i, y_i = pixel values in prediction and ground truth
- MAX = 1.0 for normalized images

Range: Typically 20–50 dB for MRI images (higher is better). PSNR is clamped to [0, 50] dB in evaluation.

3.3 Combined Scoring Formula

The final score combines both metrics with equal weighting:

$$\text{score}_{\text{slice}} = 0.5 \times \text{SSIM} + 0.5 \times \frac{\text{PSNR}}{50} \quad (4)$$

$$\text{Final Score} = \frac{1}{n} \sum_{i=1}^n \text{score}_{\text{slice},i} \quad (5)$$

Where n is the number of evaluated slices. PSNR is divided by 50 to normalize it to approximately [0, 1] range for fair weighting with SSIM.

3.4 Public/Private Leaderboard Split

Table 3: Leaderboard Configuration

Leaderboard	Slices	Percentage
Total Test Set	1000	100%
Public	500 (random)	50%
Private	500 (remaining)	50%

The random split ensures that slices from all samples and all z-positions contribute to both leaderboards, preventing overfitting to specific slice positions.

3.5 Baseline Performance

The sample submission provides a baseline using bicubic interpolation (simple upsampling without enhancement):

Table 4: Baseline Scores (Bicubic Interpolation)

Split	SSIM	PSNR (dB)	Combined Score
Public (500 slices)	0.593	15.90	0.455
Private (500 slices)	0.595	15.94	0.457
Overall (1000 slices)	0.594	15.92	0.456

Goal: Significantly exceed the baseline score of ≈ 0.46 .

4 Submission Instructions

4.1 Required Output

For each of the 5 test samples, you must:

1. Generate an enhanced 3D volume with shape (179, 221, 200)
2. Extract all 200 axial slices
3. Encode each slice to base64 format
4. Submit a CSV with 1000 rows total

4.2 CSV Format

Table 5: Submission File Format

Column	Description
row_id	Slice identifier: sample.XXX_slice.YYY
prediction	Base64-encoded numpy array (179 \times 221 pixels)

Example rows:

```
row·id,prediction
sample·019·slice·000,eJzVm3d4lUXWx7/z3vfeN...
sample·019·slice·001,eJzVm3d4lUXWx7/z3vfeN...
...
sample·023·slice·199,eJzVm3d4lUXWx7/z3vfeN...
```

4.3 Using the Extraction Script

The provided `extract_slices.py` handles slice extraction and encoding:

Listing 2: Creating a Submission

```
from extract·slices import (
    load·nifti,
    create·submission·df,
    volume·to·submission·rows
)

# Load test inputs and generate predictions
predictions = -
for sample·id in ['sample·019', 'sample·020', 'sample·021',
                 'sample·022', 'sample·023']:
    # Load low-field input
    lf·path = f'test/low·field/-sample·id'·lowfield.nii.gz'
    lf·volume = load·nifti(lf·path)

    # Your model produces enhanced volume
    # Output must be shape (179, 221, 200)
    enhanced·volume = your·model(lf·volume)

    predictions[sample·id] = enhanced·volume

# Create submission DataFrame (1000 rows)
submission·df = create·submission·df(predictions)
submission·df.to·csv('submission.csv', index=False)

print(f"Created submission with -len(submission·df)" rows")
```

4.4 Important Requirements

- 1. Output Shape:** Your predicted volume must be exactly (179, 221, 200)
- 2. All Slices:** You must submit predictions for all 200 slices per sample (1000 total)
- 3. Encoding:** Use the provided `slice_to_base64()` function—other encoding methods will cause errors
- 4. Row Order:** The order of rows doesn't matter; rows are matched by `row_id`
- 5. No NaN/Inf:** Predictions cannot contain NaN or infinite values

5 Suggested Approaches

5.1 Problem Formulation

This is a **paired 3D image enhancement** (or super-resolution) problem:

- **Input:** Low-field MRI volume (112, 138, 40)
- **Output:** Enhanced volume (179, 221, 200) matching high-field quality
- **Training:** 18 paired examples available

5.2 Baseline Methods

- **Bicubic interpolation:** Simple upsampling (provided baseline, score ≈ 0.46)
- **Trilinear interpolation:** Alternative classical upsampling

5.3 Deep Learning Approaches

- **3D U-Net:** Encoder-decoder architecture with skip connections
- **SRCNN/ESPCN:** Super-resolution CNNs (can be extended to 3D)
- **ESRGAN:** Enhanced super-resolution GAN for perceptual quality
- **Attention mechanisms:** Self-attention or transformer-based approaches
- **Residual learning:** Learn the difference between upsampled and target

5.4 Training Considerations

- Limited training data (18 samples)—consider data augmentation
- 3D context can help but increases memory requirements
- Patch-based training may be necessary for large volumes
- Loss functions: L1, L2, perceptual loss, adversarial loss

6 Glossary

Table 6: Terminology

Term	Definition
MRI	Magnetic Resonance Imaging
T1w	T1-weighted (a specific MRI contrast)
64mT	64 milliTesla (low magnetic field strength)
3T	3 Tesla (high magnetic field strength)
NIfTI	Neuroimaging Informatics Technology Initiative (file format)
SSIM	Structural Similarity Index Measure
PSNR	Peak Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio
Axial	Horizontal plane (top-down view)
Isotropic	Equal resolution in all directions

Good luck! We look forward to seeing innovative approaches to this challenging problem.