

Improved UNet for OASIS Brain Segmentation (Project 1)

Description of algorithm

UNet is a convolutional neural network (CNN) architecture introduced in 2015 by Ronneberger et al. at the Medical Image Computing and Computer-Assisted Intervention (MICCAI) conference. It was specifically designed for biomedical image segmentation—a task where precise localization of anatomical structures (e.g., brain tissues, tumors, organs) is critical, but labeled data is often scarce.

The Improved UNet builds on the original UNet with key enhancements to improve performance on brain MRI data. Core modifications include:

- Residual blocks in both encoder and decoder pathways to mitigate vanishing gradients in deep networks, enabling more stable training.
- Batch normalization after each convolutional layer to reduce internal covariate shift and accelerate convergence.
- Adjusted skip connections to better fuse high-resolution spatial details (from the encoder) with context-rich features (from the decoder), critical for distinguishing subtle brain tissue boundaries.

The model takes a grayscale 2D MRI slice as input and outputs a 4-channel segmentation mask, where each channel corresponds to a tissue class: background, cerebrospinal fluid (CSF), gray matter (GM), and white matter (WM).

Problem that the algorithm solves

Accurate segmentation of brain tissues (CSF, GM, WM) from MRI scans is a foundational task in neuroimaging. It enables:

- Quantification of tissue volumes (e.g., GM atrophy in Alzheimer's disease).
- Guiding surgical planning (e.g., targeting WM tracts).
- Monitoring disease progression over time.
- Manual segmentation is time-consuming (taking hours per scan) and prone to inter-rater variability. The Improved UNet automates this process, providing consistent, high-speed segmentation—critical for large-scale studies like the OASIS dataset (Open Access Series of Imaging Studies), which contains MRI scans of aging and neurodegenerative disease.

How the algorithm works

The Improved UNet follows an "encoder-decoder" architecture with three key components:

Encoder (Downsampling Pathway):

Converts the input MRI slice (256×256) into progressively smaller, context-rich feature maps via stacked residual blocks and max-pooling. Each residual block uses two 3×3 convolutions with batch normalization and ReLU activation, followed by a shortcut connection to preserve low-level features.

Bottleneck:

The deepest layer of the network, where high-level contextual features (e.g., global brain structure) are integrated.

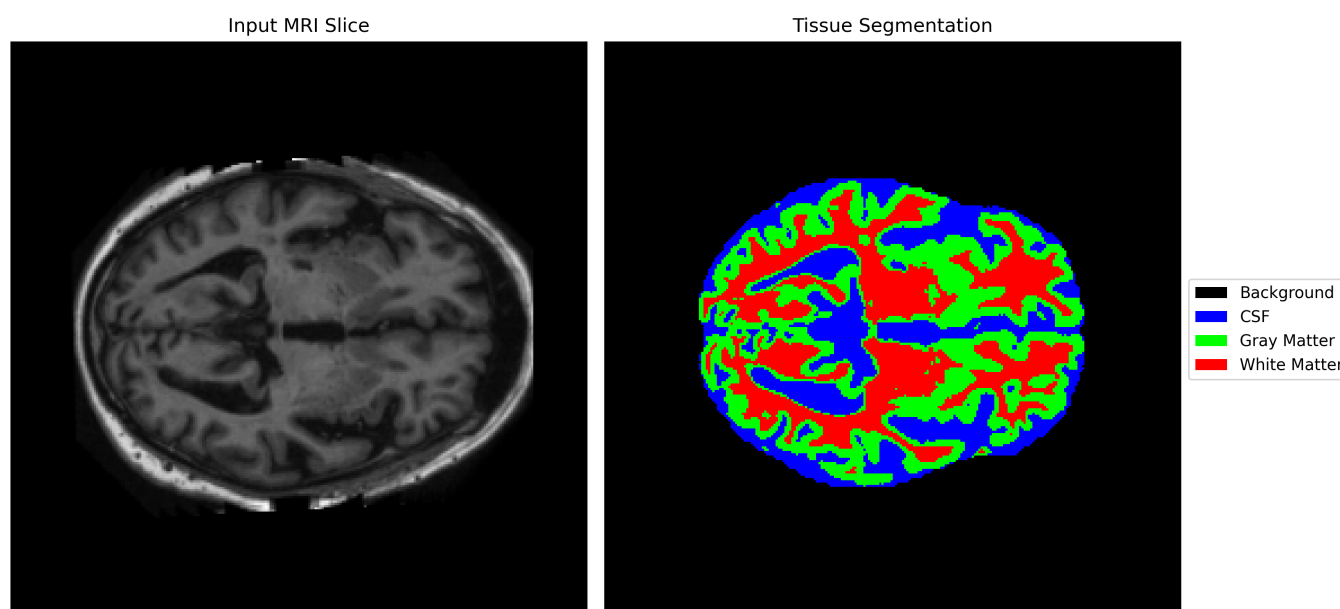
Decoder (Upsampling Pathway):

Recovers spatial resolution using transposed convolutions (upsampling) while fusing features from corresponding encoder layers via skip connections. This combines local details (e.g., tissue edges) with global context.

Output Layer:

A 1×1 convolution reduces the decoder output to 4 channels (one per tissue class). Softmax activation converts these to class probabilities, and argmax selects the most likely class for each pixel.

Figure/Visualization of the Solution



(Left: Input MRI slice | Right: Model output—colors correspond to background (black), CSF (blue), GM (green), WM (red))

Dependencies required

- Python 3.9.7
- PyTorch 1.12.1 (for model training/inference)
- OpenCV 4.5.5 (for image loading/preprocessing)
- NumPy 1.21.5 (for array operations)
- Matplotlib 3.5.2 (for visualization)

Example inputs, outputs and plots of your algorithm

Inputs

Example Input: A 256×256 grayscale MRI slice from the OASIS dataset (e.g., OAS1_0001_MR1_slice100.png), with pixel values in $[0, 255]$.

Segmentation Mask: A 256×256 array where each pixel is labeled 0 (background), 1 (CSF), 2 (GM), or 3 (WM).

```
Epoch 3/3  
Training: 100%|██████████████████████████████████████████████████████████████████████████████| 2416/2416 [06:59<00:00, 5.75it/s]  
Validation: 100%|██████████████████████████████████████████████████████████████████████████████| 280/280 [00:25<00:00, 11.17it/s]  
Train - Loss: 0.0234, Dice: 0.9716  
Val - Loss: 0.0230, Dice: 0.9717  
Saved best model (Val Dice: 0.9717)  
Curves saved to training_curves.png  
Testing: 100%|██████████████████████████████████████████████████████████████████████████████| 136/136 [00:16<00:00, 8.50it/s]  
  
Test Results - Loss: 0.0241, Dice: 0.9711
```

The figure consists of two side-by-side line plots. The left plot, titled 'Loss Curves', shows the training and validation loss over 2 epochs. The y-axis is labeled 'Loss' and ranges from 0.03 to 0.07. The x-axis is labeled 'Epoch' and ranges from 0.00 to 2.00. The training loss (blue line) starts at approximately 0.068 and decreases to about 0.024. The validation loss (orange line) starts at approximately 0.033 and decreases slightly to about 0.031. The right plot, titled 'Dice Score', shows the training and validation Dice score over 2 epochs. The y-axis is labeled 'Score (0-1)' and ranges from 0.9525 to 0.9700. The x-axis is labeled 'Epoch' and ranges from 0.00 to 2.00. The training Dice score (blue line) starts at approximately 0.9525 and increases to about 0.971. The validation Dice score (orange line) starts at approximately 0.963 and decreases slightly to about 0.961.

Epoch	Train Loss	Validation Loss	Train Dice Score	Validation Dice Score
0.00	0.068	0.033	0.9525	0.963
1.00	0.027	0.031	0.968	0.962
2.00	0.024	0.031	0.971	0.961

Preprocessing

- **Normalization:** MRI pixel values are scaled to $[0, 1]$ using min-max normalization.
- **Mask Conversion:** Ground truth masks (originally 0, 85, 170, 255 for 4 classes) are normalized to $[0, 1]$ and converted to integer labels (0–3) for cross-entropy loss.

Training, Validation, and Testing Data Splits

Dataset	Sample Count	Percentage	Purpose
Training	9,664	85.3%	Update model weights
Validation	1,120	9.9%	Tune hyperparameters (e.g., learning rate) and select the best model (based on validation Dice score)
Testing	544	4.8%	Held out for final generalization evaluation