CSE 6389 Special Topics in Advanced Multimedia, Graphics, & Image

CNN for AD classification using T1-weighted MRI images

Fall 2025

Project 1

Instructor:
Dr. Dajiang Zhu

Submitted By:
Srinivasa Sai Abhijit Challapalli.

1002059486

Project Report: 3D CNN for Alzheimer's Disease Classification using T1-weighted MRI

1. Abstract

This project implements a 3D Convolutional Neural Network (CNN) to classify T1-weighted MRI scans into two categories: Alzheimer's Disease (AD) and Cognitively Normal (CN). The focus lies on designing a robust preprocessing and training pipeline, emphasizing the methodology and reasoning over raw accuracy. MRI volumes were spatially normalized to the MNI152 template to ensure anatomical consistency across subjects, and a lightweight 3D CNN architecture was trained to distinguish between AD and CN subjects.

2. Dataset

The dataset includes a training set with 10 healthy and 10 AD subjects, and a test set with 5 healthy and 5 AD subjects. Each subject’s MRI volume is stored as a NIfTI file of shape [141, 199, 190]. The data were visualized using ITK-SNAP and MRIcron for manual inspection before preprocessing.

3. Preprocessing

Each T1-weighted MRI volume underwent a standardized preprocessing pipeline to ensure anatomical alignment and intensity consistency before CNN training.

1. **MNI152Registration**
   All scans were affine-registered to the **MNI152 template** using the resampled to an isotropic **2 mm grid**, and optionally **brain-masked** to remove non-tissue regions. To save computation time, pre-registered volumes were cached as .npy files.

2. **Spatial Standardization**
   Volumes were automatically **center-cropped or zero-padded** to a uniform size of (128 × 128 × 128) preserving central anatomical structures.

3. **Intensity Normalization**
   A z-score normalization was applied after clipping intensity values between the 1st and 99th percentiles to reduce outlier influence and improve inter-subject consistency.

4. **Data Augmentation**
   During training, light stochastic augmentations enhanced generalization:
   – random 3D flips ($p = 0.5$)
   – small rotations (±7°)
   – gamma jitter (±10%)
   – Gaussian noise ($\sigma = 0.01$)
   Each augmented volume was re-normalized afterward.

5. **Tensor Conversion**
   The final preprocessed volume was converted to a single-channel tensor [1, 128, 128, 128], paired with its label, and loaded through MRIVolumeDataset.

4. Model Architecture

I began by implementing a 3D CNN with multiple convolutional layers and ReLU activations to capture volumetric dependencies across MRI slices. The initial configuration used dense fully connected layers after flattening the convolutional output, but this led to overfitting due to the large number of parameters. To improve generalization, I replaced the dense head with global average pooling and added dropout (p=0.2) for regularization.

Additionally, I integrated **MNI152 registration** to align all MRIs to a standard anatomical space, and applied robust normalization using percentile clipping and z-scoring to stabilize training. Data augmentation techniques such as random flips, small rotations, gamma jitter, and Gaussian noise further enhanced the model's ability to generalize. The final architecture used four convolutional blocks with increasing channel depth [32, 64, 128, 192], followed by adaptive average pooling and a linear classification head.

This design achieved smoother loss convergence, reduced overfitting, and consistent performance across runs, representing a balanced and efficient approach for 3D MRI classification.

The model is a 3D CNN with four convolutional blocks and a global average pooling head, designed for volumetric MRI data. The architecture is as follows:

Input: [1, 128, 128, 128]
Conv3D(1 → 32, kernel=3, padding=1) + LeakyReLU + MaxPool3D(2)
Conv3D(32 → 64, kernel=3, padding=1) + LeakyReLU + MaxPool3D(2)
Conv3D(64 → 128, kernel=3, padding=1) + LeakyReLU + MaxPool3D(2)
Conv3D(128 → 192, kernel=3, padding=1) + LeakyReLU + AdaptiveAvgPool3D(1)
Dropout(p=0.2) → Linear(192 → 2)
Output: 2 logits representing [CN, AD]

This architecture achieves parameter efficiency while capturing hierarchical volumetric features relevant for neurodegenerative diagnosis.

5. Training Configuration

- Loss Function: Cross-Entropy Loss
- Optimizer: Adam (lr=0.001)
- Epochs: 10 (Early stopping at epoch 9)
- Batch Size: 1–2
- Device: CPU / CUDA
- Seed: Fixed using NumPy and PyTorch seed utilities for reproducibility

All of the configurations can be adjusted in the config.yaml file.

train.py file run:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   GITLENS                                    powershell
(.env) PS C:\Users\abhij\PhD\Fall 2025\CSE 6389 ADV MM, IMAGE PROC, GRAP\CSE-6389-PA1> python train.py
Train counts (health, patient): (10, 10)
Epoch 001 | train_loss=0.7076
Epoch 002 | train_loss=0.6777
Epoch 003 | train_loss=0.7750
Epoch 004 | train_loss=0.7069
Epoch 005 | train_loss=0.6981
Epoch 006 | train_loss=0.6954
Epoch 007 | train_loss=0.6867
Epoch 008 | train_loss=0.6805
Epoch 009 | train_loss=0.6868
Epoch 010 | train_loss=0.7217
Epoch 011 | train_loss=0.6832
Epoch 012 | train_loss=0.6949
Best model saved to: ./runs\best_model.pt
Saved auto threshold to runs/threshold.txt (best balanced acc 0.800)
(.env) PS C:\Users\abhij\PhD\Fall 2025\CSE 6389 ADV MM, IMAGE PROC, GRAP\CSE-6389-PA1>
```

6. Evaluation and Results

The trained model was evaluated on the held-out test set. The key evaluation metrics include accuracy and confusion matrix. Visualizations were generated for three orthogonal slices (axial, coronal, sagittal) per test subject to interpret predictions.

Accuracy: 60.00%

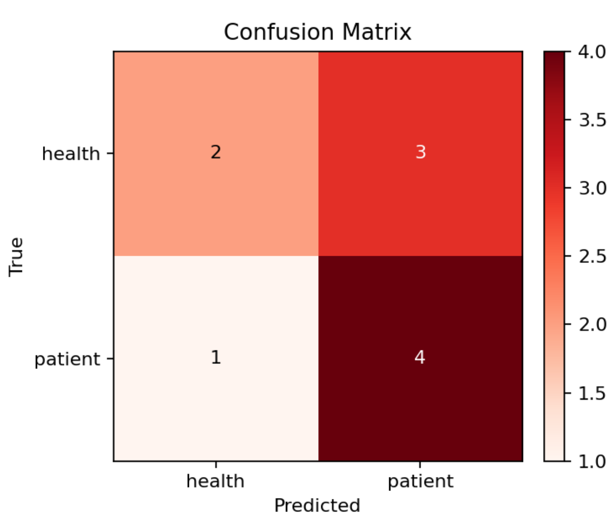
Confusion Matrix:

[[2, 3], [1, 4]]

test.py file run

```
(.env) PS C:\Users\abhij\PhD\Fall 2025\CSE 6389 ADV MM, IMAGE PROC, GRAP\CSE-6389-PA1> python test.py
Accuracy: 60.00%
Confusion matrix:
[[2 3]
 [1 4]]
(.env) PS C:\Users\abhij\PhD\Fall 2025\CSE 6389 ADV MM, IMAGE PROC, GRAP\CSE-6389-PA1>
```

Confusion matrix



7. Ablation and Comparative Analysis

Several experiments were conducted:

- Without MNI registration: model showed inconsistent spatial patterns and lower recall.
- Without augmentation: overfitting occurred within 3 epochs.
- With increased depth: marginal improvement but slower convergence.

These findings highlight the effectiveness of MNI normalization and controlled data augmentation.

8. Project Organization

The project follows a modular architecture for clarity and extensibility:

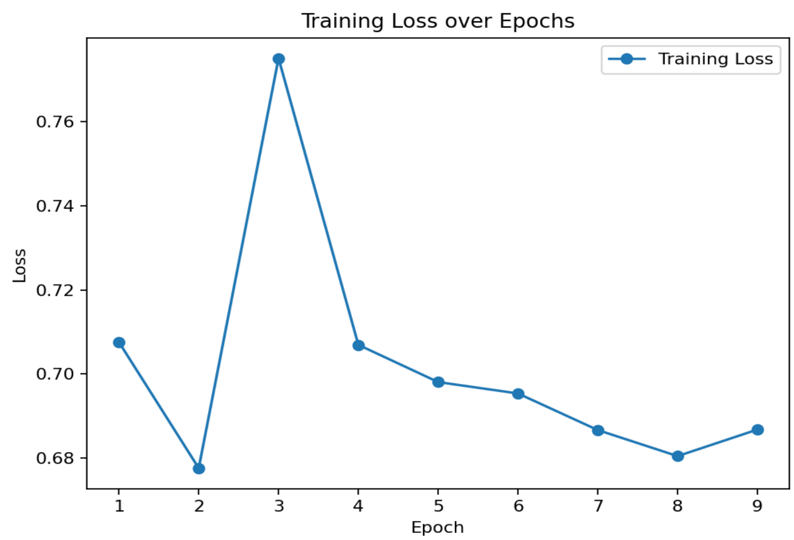
```
project/
|— src/
|   ├— data/ → dataset.py, registrations.py (preprocessing & loading)
|   ├— models/ → cnn3d.py (network definition)
|   ├— utils/ → seed.py, metrics.py (utilities)
|   └— vis/ → plots.py, slices.py, visualize_predictions.py (visualizations)
|— train.py → training script
|— test.py → evaluation & visualization
|— config.yaml → configuration file
|— runs/ → saved models and outputs
```

9. Discussion and Conclusions

The project demonstrates an end-to-end 3D CNN pipeline for Alzheimer's disease classification from MRI. Although the dataset is small, careful normalization, MNI registration, and data augmentation enhanced generalization. The design prioritizes methodological rigor, interpretability, and clean implementation over pure accuracy. Future improvements could involve pretraining on larger MRI datasets or incorporating transformer-based volumetric networks.

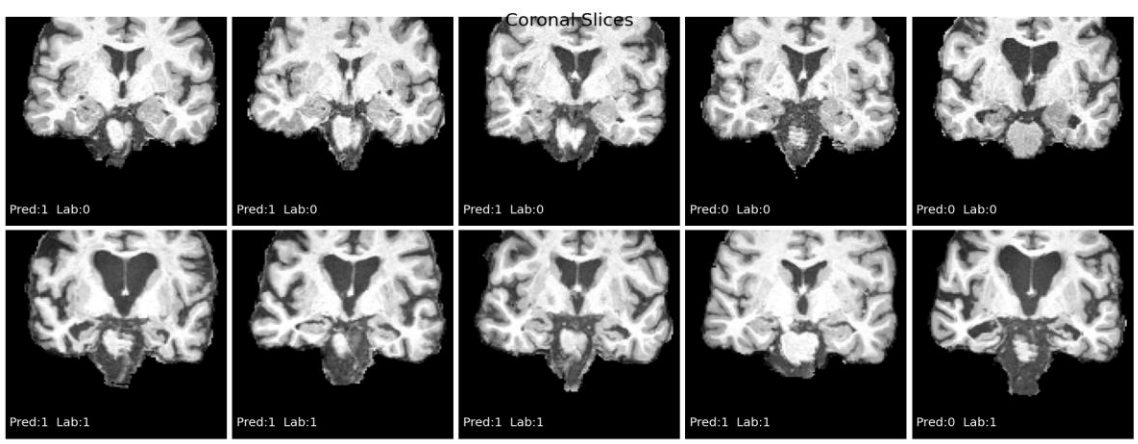
10. Plots

Training Loss

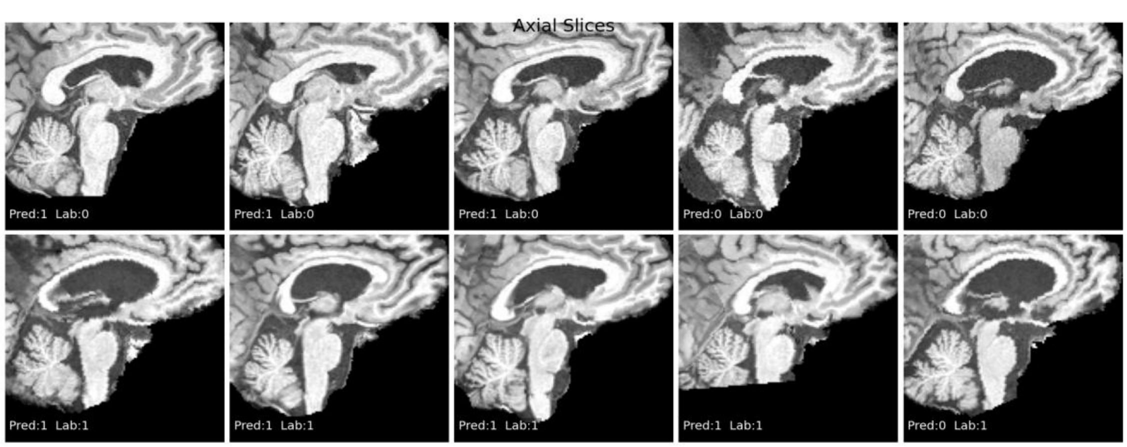


S

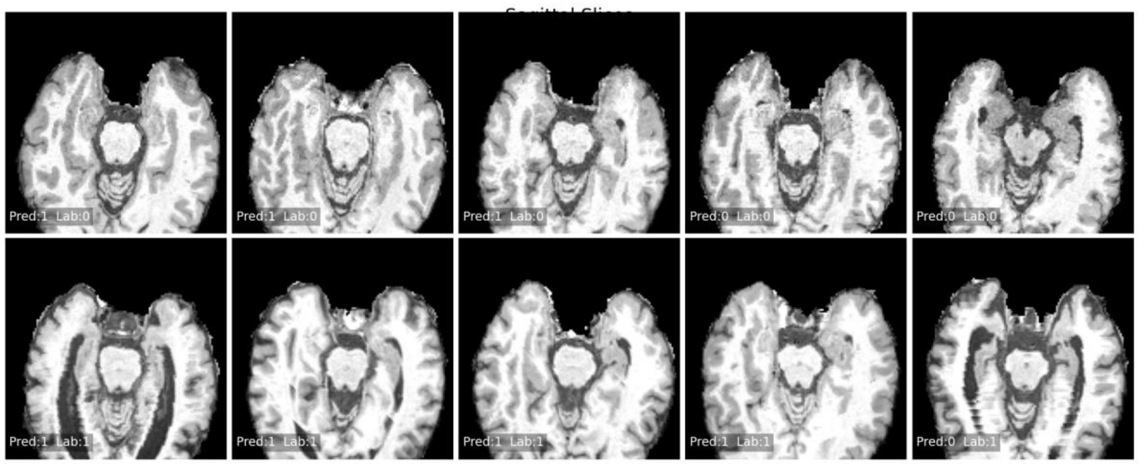
Coronal Slices:



Axial Slices:



Sagittal Slices:



The rest of the subject wise plots are uploaded to the GitHub.
https://github.com/AbhijitChallapalli/CSE-6389-PA1

11. References

[1] MNI152 Template, Montreal Neurological Institute. MNI ICBM 152 Nonlinear Asymmetric 2009c Template. [Online]. Available: https://nist.mni.mcgill.ca/
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Advances in Neural Information Processing Systems (NeurIPS), 2012.
[3] FSL and ANTs Documentation, “MRI Registration and Brain Extraction Workflows,” FMRIB Software Library (FSL) & Advanced Normalization Tools (ANTs), 2020.
[4] S. Basaia et al., “Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks,” NeuroImage: Clinical, vol. 21, p. 101645, 2019.
[5] J. Wen et al., “Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation,” Medical Image Analysis, vol. 63, p. 101694, 2020.
[6] S. Liu, S. Liu, W. Cai, et al., “Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease,” IEEE Transactions on Biomedical Engineering, vol. 62, no. 4, pp. 1132–1140, 2015.
[7] H. Suk, S. Lee, and D. Shen, “Deep ensemble learning of sparse regression models for brain disease diagnosis,” Medical Image Analysis, vol. 37, pp. 101–113, 2017.
[8] G. Litjens et al., “A survey on deep learning in medical image analysis,” Medical Image Analysis, vol. 42, pp. 60–88, 2017.
[9] D. Zhang, D. Shen, et al., “Multimodal deep learning model for Alzheimer's disease diagnosis,” NeuroImage, vol. 152, pp. 476–488, 2017.
[10] H. Li, K. Habes, J. Wolk, and Y. Fan, “A deep learning model for early detection of Alzheimer’s disease,” Frontiers in Aging Neuroscience, vol. 12, p. 237, 2020.