UNIVERSITY OF
BIRMINGHAM

Transfer Learning for Alzheimer's Disease Detection: Adapting Video Classification Models for MRI Scans

Rhys W. Alexander (2458177)

Final project report submitted
in partial fulfilment for the degree of
B.SCI. IN ARTIFICIAL INTELLIGENCE AND COMPUTER SCIENCE

Date: 2nd April 2025
Word count: X,XXX

Project supervisor:
Dr Rickson Mesquita

Contents

1 Abstract

2 Introduction

3 Literature review

4 Methodology

4.1 Data Acquisition and Characteristics

The Alzheimer's Disease Neuroimaging Initiative (ADNI) database served as the primary data source for this study. ADNI represents a comprehensive, longitudinal dataset specifically designed for Alzheimer's disease research, offering rigorously standardized MRI acquisitions with corresponding clinical diagnoses and metadata.

4.1.1 Dataset Selection and Access

After evaluating potential neuroimaging repositories (including OASIS), ADNI was selected for its comprehensive coverage, standardized acquisition protocols, and expert-validated diagnoses. Access was obtained through a formal application process describing the research objectives. The analysis initially utilized data from ADNI-1, later expanding to incorporate volumes from ADNI-2, ADNI-3, and ADNI-4 to increase sample diversity and size.

4.1.2 Image Acquisition Parameters

All selected scans were T1-weighted Magnetization Prepared Rapid Gradient Echo (MPRAGE) sequences, chosen for their optimal gray/white matter contrast which facilitates hippocampal visualization, standardized acquisition parameters across multiple ADNI sites, high signal-to-noise ratio for structural analysis, and sensitivity to hippocampal atrophy, a primary biomarker for AD progression. Additionally, the widespread clinical availability and established role of MPRAGE in AD assessment made it an ideal choice for this study.

The T1w MPRAGE sequences typically featured field strengths of 1.5T or 3T with approximately $1mm^3$ isotropic resolution. The acquisition matrix was approximately $256\times256\times170$, with TR/TE parameters standardized according to ADNI protocol to ensure consistency across imaging sites.

4.1.3 Subject Demographics and Diagnostic Criteria

Subjects were classified into two distinct diagnostic categories: Alzheimer's Disease (AD) and Cognitively Normal (CN). The AD cohort consisted of subjects meeting NINCDS-ADRDA criteria for probable AD, while the CN cohort comprised control subjects without significant cognitive impairment. The original dataset distribution was approximately 33% AD and 67% CN cases. After initial testing revealed potential overfitting issues (discussed later), additional scans were incorporated. During this expansion phase, all available new AD scans were included, with CN subjects carefully sampled to achieve a balanced 50/50 diagnostic distribution in the final dataset to optimize model training.

4.1.4 Data Distribution Analysis

The final dataset contained 1,300 T1w MRI scans from 408 unique subjects:

- AD cohort: 650 scans from 203 subjects
- CN cohort: 650 scans from 205 subjects

Following subject-level splitting (detailed in Data Splitting Strategy section), the distribution across partitions was:

- Training set: 1,023 scans (512 AD, 511 CN) from 248 subjects (133 AD, 115 CN)
- Validation set: 139 scans (69 AD, 70 CN) from 80 subjects (35 AD, 45 CN)
- Test set: 138 scans (69 AD, 69 CN) from 80 subjects (35 AD, 45 CN)

This distribution ensured each partition contained sufficient samples for robust model training and evaluation while maintaining diagnostic balance. The deliberate focus on AD versus CN classification (excluding Mild Cognitive Impairment) reflects the clearer structural changes observable in established AD, particularly the hippocampal atrophy that serves as a primary biomarker for disease progression.

4.2 Preprocessing Pipeline

The preprocessing pipeline was meticulously designed to prepare structural MRI data for optimal deep learning model performance while preserving clinically relevant features. Each stage was selected based on neuroimaging best practices and computational considerations specific to 3D neural network training.

4.2.1 DICOM to NIfTI Conversion

The initial step involved converting the native DICOM format files from ADNI to NIfTI format. This conversion was essential as NIfTI provides a consolidated volumetric rep-

resentation of brain scans, facilitating 3D processing compared to the slice-by-slice arrangement of DICOM files. The conversion preserved header information while creating unified volumetric files using the `dicom2nifti` library with reorientation applied during conversion to ensure consistent initial alignment.

```
dicom2nifti.convert_directory(root, nii_output_dir, compression=True, reorient=True)
```

This compression parameter was enabled to reduce storage requirements without information loss, particularly important given the large dataset size (1,300 scans).

4.2.2 Skull Stripping

Skull stripping was implemented using SynthStrip, a deep learning-based method that represents the current state-of-the-art for brain extraction. The selection of SynthStrip over traditional alternatives like Brain Extraction Tool (BET) was justified by several key advantages. SynthStrip demonstrates superior robustness to variations across diverse acquisition parameters and pathological conditions, which is critical for a heterogeneous dataset spanning multiple ADNI phases. The deep learning foundation of SynthStrip provides more consistent results across subjects compared to threshold-based methods, as more primitive approaches were shown to inaccurately crop atrophied regions, leading to significant information loss. Additionally, SynthStrip better preserves the detailed cortical boundaries that may contain relevant structural information for AD classification. As a synthetic data-trained model, SynthStrip also handles the variability in ADNI data more effectively than traditional algorithms, offering stronger generalization capability.

While SynthStrip required approximately 2.5 minutes per scan on the available hardware, this processing time was justified by the quality of results, as inconsistent skull stripping could introduce confounding artifacts that might be misinterpreted as disease-related changes.

4.2.3 Voxel Standardization

Spatial resolution standardization was performed using ANTs (Advanced Normalization Tools) to resample all volumes to isotropic $1\times1\times1$mm voxels:

```
resampled_img = ants.resample_image(img, (1,1,1), use_voxels=False)
```

This standardization step was crucial for three primary reasons:

1. **Eliminating resolution variability**: Although ADNI enforces acquisition protocols, some variation in voxel dimensions exists across scanners and timepoints.
2. **Isotropic representation**: Consistent cubic voxels ensure that convolutional filters operate uniformly across all three dimensions, preventing directional bias.

3. **Model compatibility**: Standardized resolution simplifies the implementation of 3D convolutional operations and ensures consistent spatial feature extraction.

The resampling was implemented using third-order spline interpolation to maintain structural integrity during resolution adjustment.

4.2.4 Cropping and Reshaping Strategy

A critical preprocessing innovation was an adaptive cropping procedure followed by reshaping to 128×128×128 dimensions. This approach was developed after initial experiments revealed significant information loss when using simple interpolation:

```
# Crop the brain with padding
cropped_img, crop_coords = crop_brain_from_mri(img_data, padding=3)

# Reshape using cubic interpolation
zoom_factors = [t / s for t, s in zip(target_shape, cropped_img.shape)]
final_img = zoom(cropped_img, zoom_factors, order=3)
```

The implemented method:

1. Automatically identifies brain-containing regions using intensity thresholding
2. Crops to these regions with a 3-voxel padding to ensure complete brain coverage
3. Applies cubic interpolation to the cropped volume to reach target dimensions

This approach preserved significantly more anatomical detail compared to naive downsampling of the entire volume, as demonstrated by validation experiments showing that this cropping strategy retained approximately 35% more effective resolution for critical structures like the hippocampus.

The 128×128×128 dimension was selected based on:

- Sufficient resolution to preserve hippocampal and ventricular details
- Memory constraints for model training
- Compatibility with deep network architectures
- Balanced compromise between resolution and computational efficiency

4.2.5 Bias Field Correction and Orientation Standardization

N4 bias field correction was applied to mitigate intensity inhomogeneities resulting from magnetic field variations:

```
bias_corrected = ants.n4_bias_field_correction(input_image)
```

This correction is particularly important for AD classification as it prevents intensity variations that might be misinterpreted as structural changes. Similarly, all volumes were reoriented to Right-Anterior-Superior (RAS) orientation to ensure consistent directionality across the dataset:

```
canonical_img = nib.as_closest_canonical(img)
```

Standardized orientation eliminates the potential confound of different brain orientations influencing the learning process, allowing the model to focus solely on relevant structural differences.

4.2.6 Spatial Normalization

While conventional neuroimaging pipelines often include registration to a standard template space (e.g., MNI152), this step was deliberately omitted for several key reasons:

1. **Preservation of native atrophy patterns**: Spatial normalization can distort or obscure the very atrophic changes that differentiate AD patients from controls, particularly in the hippocampus.
2. **Model capability**: Deep convolutional networks demonstrate inherent translation invariance and can learn to identify relevant structures regardless of precise alignment, making explicit normalization potentially redundant.
3. **Avoiding interpolation artifacts**: The registration process introduces additional interpolation steps that can smooth subtle structural boundaries critical for classification.
4. **Computational efficiency**: Omitting this intensive processing step significantly reduced preprocessing time without compromising classification performance.

Validation experiments confirmed that models trained on native-space data performed comparably to or better than those trained on normalized data, supporting this methodological decision. This approach is aligned with recent literature suggesting that deep learning models for brain MRI classification benefit from learning in subject-native space rather than standardized space.

The comprehensive pipeline ultimately produced a dataset of 1,300 preprocessed volumes with consistent dimensions, orientation, and intensity characteristics while preserving the structural variations essential for AD classification. This carefully crafted preprocessing strategy balances computational constraints with the preservation of clinically relevant features, providing an optimal foundation for the subsequent neural network training.

4.3 Data Splitting Strategy

The data splitting strategy was carefully designed to prevent data leakage while ensuring balanced representation of diagnostic groups across training, validation, and test sets. Unlike conventional image classification tasks, neuroimaging datasets present unique challenges as multiple scans often exist for the same subject across different timepoints, requiring subject-level rather than scan-level splitting.

4.3.1 Subject-Level Isolation

A strict subject-level isolation approach was implemented to ensure no individual subject appeared in multiple dataset partitions. This critical methodological decision was motivated by initial experiments that revealed artificially inflated performance metrics (90% accuracy) when subjects were allowed to appear across partitions. By completely isolating subjects between splits, a more realistic performance assessment (70% accuracy) was achieved, better reflecting the model's true generalization capability to unseen individuals.

4.3.2 Round-Robin Approach for Balanced Distribution

A round-robin selection algorithm was implemented to ensure balanced representation across dataset partitions while maintaining diagnostic class balance. This approach methodically cycled through subjects, allocating them to train, validation, and test sets according to predetermined ratios (80% training, 10% validation, 10% test) while ensuring an equal number of scans from each diagnostic category:

1. Subjects were first grouped by diagnostic condition (AD or CN).
2. Within each condition, subjects are sorted in ascending order by the number of scans that pertain to them.
3. The round-robin algorithm allocated subjects to each partition, test, then validation, the train, until target scan counts were achieved.
4. Final scan counts were balanced to prevent class imbalance (650 scans per diagnostic category).

This approach ensured that even with minimal data there was a large enough subject diversity in the validation and test sets to give a fair evaluation.

4.3.3 Final Distribution Statistics

The final dataset distribution across partitions after implementing the subject-level isolation and round-robin approach was:

- **Alzheimer's Disease (AD) cohort:**
  - Training set: 512 scans from 133 unique subjects
  - Validation set: 69 scans from 35 unique subjects
  - Test set: 69 scans from 35 unique subjects
- **Cognitively Normal (CN) cohort:**
  - Training set: 511 scans from 115 unique subjects
  - Validation set: 70 scans from 45 unique subjects
  - Test set: 69 scans from 45 unique subjects

This distribution ensured approximately 79% of scans were allocated to training, with the remaining 21% evenly divided between validation and test sets, while maintaining diagnostic balance within each partition.

Data Leakage Prevention Special attention was devoted to preventing subtle forms of data leakage that could compromise model evaluation. The subject-level isolation was rigorously enforced through tracking of unique subject identifiers, and all preprocessing parameters (such as intensity normalization statistics) were computed independently within each partition to prevent information bleeding across splits.

This methodologically sound splitting approach provided a robust foundation for model training and evaluation, ensuring that performance metrics would accurately reflect the model's ability to generalize to entirely new subjects rather than merely recognizing previously seen individuals in different scans.

4.4 Data Augmentation

Data augmentation was strategically implemented to improve model generalization by exposing the network to controlled variations while preserving clinically relevant features. The augmentation pipeline evolved through experimental validation to balance diversity enhancement with preservation of diagnostic information.

4.4.1 Augmentation Strategy Development

The augmentation approach underwent several iterations, beginning with a comprehensive set of transformations adapted from general computer vision practices. Through systematic evaluation, the final pipeline was refined to include only those transformations that demonstrably improved generalization without distorting critical diagnostic features:

```
tio.Compose(
```

```
[
    tio.RandomNoise(mean=0.0, std=0.1, p=0.3),
    tio.RandomGamma(log_gamma=(-0.2, 0.2), p=0.3),
    tio.ZNormalization(),
]
```

This minimalist approach was adopted after observing that more aggressive transformations either failed to improve performance or actively degraded it. The augmentation pipeline was applied exclusively to the training set, while validation and test sets received only intensity normalization to maintain evaluation consistency.

4.4.2 Justification for Selected Techniques

Each augmentation technique was selected based on specific neuroimaging considerations:

1. **Random Noise Addition**
    - Simulates natural scanner variability and noise artifacts
    - Promotes robustness to image quality variations across scanning sites
    - Implemented with a moderate noise level to preserve structural integrity
    - 30% probability prevents overreliance on noise-resilient features
2. **Random Gamma Adjustment**
    - Simulates intensity variations common in MRI acquisition
    - Enhances model robustness to contrast differences between scanners
    - Restricted to a narrow range to preserve anatomical relationships
    - Complements the bias field correction applied during preprocessing
3. **Z-Score Normalization** (applied to all volumes)
    - Standardizes intensity values to zero mean and unit variance
    - Critical for consistent feature extraction across scans
    - Mitigates the effect of scanner-specific intensity scales
    - Applied to all datasets (not just training) to ensure consistent input distribution

4.4.3 Augmentation Impact Analysis

Notably, several common augmentation techniques were deliberately excluded after experimental evaluation showed they either provided no benefit or negatively impacted

performance:

1. **Geometric Transformations** (rotations, flips):
    - Initial experiments included rotations ($\pm 90°$) and random flips
    - These transformations significantly increased training time ($\sim$20 epochs vs. $\sim$5 epochs to converge)
    - Provided no measurable improvement in validation accuracy
    - Likely redundant given the inherent orientation variability already present in MRI data
    - May have introduced unrealistic transformations not encountered in clinical settings
2. **Random Scaling**:
    - Initially tested with scale factors of 0.9-1.1
    - Showed no significant improvement in generalization
    - Potentially disrupted the carefully standardized voxel dimensions established during preprocessing

The progression from extensive transformations (using MONAI's comprehensive augmentation library) to a more focused set (using TorchIO's targeted medical imaging augmentations) and finally to the minimal set described above reflected an evidence-based refinement process. This evolution was guided by systematically monitoring validation performance and convergence speed after each modification.

The final augmentation strategy represents an optimal balance between enhancing model robustness and preserving the clinically significant structural features essential for accurate AD classification, particularly the hippocampal atrophy patterns that serve as primary biomarkers.

4.5 Model Architectures

4.5.1 3D ResNet Architecture

The primary model architecture employed was a modified 3D ResNet-18 (r3d_18), selected for several key characteristics:

1. **Residual connections**: These skip connections mitigate the vanishing gradient problem in deep networks, allowing effective training even with limited data.

2. **3D convolutions**: Unlike 2D approaches that process each slice independently, 3D convolutions capture volumetric patterns across all three dimensions, preserving spatial relationships critical for detecting hippocampal atrophy.

3. **Parameter efficiency**: With approximately 33 million parameters, ResNet-18 offered a balance between model capacity and computational efficiency, enabling training on consumer hardware.

4. **Proven effectiveness**: The ResNet architecture family has demonstrated robust performance across numerous computer vision tasks, including medical imaging applications.

The implementation utilized the PyTorch `torchvision.models.video` module, specifically the `r3d_18` model pre-trained on the Kinetics400 action recognition dataset.The first convolutional layer was modified to accept single-channel MRI volumes rather than the three-channel RGB videos used in the original architecture. The final fully connected layer was replaced to output two classes (AD vs. CN) instead of the 400 action classes in the original model. These architectural modifications preserved the core feature extraction capabilities of the ResNet model while adapting it to the specific requirements of binary volumetric MRI classification.

4.5.2 Transfer Learning Strategy

A systematic transfer learning approach was implemented with layer freezing to leverage the pre-trained weights from video classification. Early convolutional layers (stem, layer1, layer2, layer3) were frozen to preserve general low-level feature detectors learned from video data. The final residual block (layer4) and fully connected layer were unfrozen to allow adaptation to MRI-specific features. This approach maintained approximately 25% of parameters as frozen (8.2 million) while fine-tuning the remaining 75% (24.9 million), striking a balance between preserving pre-trained knowledge and adapting to the target domain. Initial experiments with more aggressive freezing (keeping only the final fully connected layer trainable) resulted in numerical instabilities during training, manifested as NaN losses, suggesting that significant domain adaptation was necessary given the substantial differences between video action recognition and MRI classification.

Notably, the learning rate strategy was aligned with this transfer learning approach, implementing a higher learning rate (10×) for the newly initialized fully connected layer compared to the pre-trained but unfrozen convolutional layers. This differential learning rate strategy allowed more aggressive adaptation in the task-specific output layer while making more conservative updates to the pre-trained feature extraction layers.

4.5.3 Alternative Architecture Exploration

To validate the architectural choices, and to justify using 3D convolutioons as opposed to the previously researched 2D methods, alternative models were explored:

1. **Mixed Convolution 3D Network**: This model (MC3-18) uses a hybrid approach combining 2D and 3D convolutions, hypothesized to potentially offer computational efficiency while maintaining performance.

   Experimental results with MC3-18 showed less stable training dynamics and inferior performance compared to the pure 3D approach of R3D-18, supporting the importance of fully volumetric feature extraction for structural MRI analysis. The differences in performance provided empirical justification for the primary architectural choice.

2. **(2+1)D Convolution Network**: Following the investigation of MC3-18, a (2+1)D architecture was also evaluated. This approach decomposes 3D convolutions into separate spatial (2D) and temporal (1D) convolutions, a technique that has shown promise in video classification tasks.

   Results with the (2+1)D architecture revealed performance that was slightly worse than MC3-18, continuing the observed trend that classification accuracy decreased as the model architecture incorporated more 2D elements. This progression (R3D $>$ MC3 $>$ (2+1)D) strongly suggests that preserving the full 3D spatial context through pure 3D convolutions is critical for detecting the subtle volumetric patterns associated with Alzheimer's disease in MRI data.

3. **Multiscale Vision Transformer**: Recent advances in vision transformers prompted investigation of their potential for 3D MRI classification. However, initial implementation attempts revealed significant computational barriers:

   (a) Memory requirements exceeded available hardware capabilities (32GB RAM requirement for 128×128×128 volumes)

   (b) Architectural mismatch between the input dimensions required by MViT (designed for 16×224×224 video clips) and the cubical 128×128×128 MRI volumes

   (c) Transformer architectures typically require substantially larger training datasets than were available

   These constraints prevented full evaluation of transformer-based approaches, highlighting an important practical limitation in applying state-of-the-art vision models to medical imaging with limited computational resources.

4.5.4 Parameter Counts and Computational Considerations

The final model architecture parameters were:

- **Total parameters**: 33,148,482
- **Trainable parameters**: 24,909,826 (75.15%)
- **Frozen parameters**: 8,238,656 (24.85%)

These figures represent a significant reduction compared to larger architectures like ResNet-50 or ViT variants, making training feasible on consumer-grade hardware while maintaining sufficient capacity for the classification task. The reduced parameter count also potentially mitigated overfitting given the relatively small dataset size.

4.6 Training Framework and Implementation

4.6.1 Computational Environment

Model training was conducted on an M1 Mac using the Metal Performance Shaders (MPS) acceleration framework. This hardware configuration imposed certain constraints on the implementation, with each epoch requiring approximately one hour of computation time and total training runs taking upwards of 20 hours. These hardware limitations influenced several implementation decisions, including batch size selection and model architecture choices.

While attempts were made to optimize training speed through techniques like mixed precision training and CPU-GPU synchronization optimization, performance improvements were minimal. The majority of computational time was consumed by the model's forward pass through the 3D volumes, which could not be significantly accelerated without more powerful hardware.

4.6.2 Hyperparameter Selection

Hyperparameters were carefully selected through systematic experimentation and validation performance tracking. The final configuration included:

- **Learning rate strategy**: A dual learning rate approach was implemented, with the newly initialized fully connected layer receiving a 10× higher learning rate (0.001) compared to the fine-tuned convolutional layers (0.0001). This differential strategy allowed more aggressive adaptation in the task-specific output layer while making conservative updates to the pre-trained feature extraction layers.
- **Optimizer**: AdamW with weight decay (0.01) was selected to mitigate overfitting given the relatively small dataset size. The weight decay regularization helped

constrain the model's parameter space, particularly important when working with pre-trained features.

- **Batch size**: Limited to 2 samples per batch due to memory constraints of processing 128×128×128 volumetric inputs. Despite the small batch size, training stability was maintained through appropriate learning rate selection.
- **Learning rate scheduling**: Cosine annealing with warm restarts ($T_0 = 5$) was implemented to prevent convergence to local minima. This scheduling strategy periodically reduces the learning rate following a cosine curve before resetting it, allowing the model to escape suboptimal solutions.

The hyperparameter selection process was guided by both theoretical considerations and empirical validation, with each configuration tracked in Weights & Biases to enable systematic comparison.

4.6.3 Loss Function and Class Weighting

A weighted cross-entropy loss function was implemented to address potential class imbalance, particularly important during initial experiments when the dataset had not yet been fully balanced:

```
# Calculate class weights for imbalanced data
num_ad = train_dataset.labels.count(1)
num_cn = train_dataset.labels.count(0)
total = num_ad + num_cn

# Inverse frequency weighting
weight_cn = total / (2 * num_cn) if num_cn > 0 else 1.0
weight_ad = total / (2 * num_ad) if num_ad > 0 else 1.0

class_weights = torch.tensor([weight_cn, weight_ad], device=device)
criterion = nn.CrossEntropyLoss(weight=class_weights)
```

This weighting strategy ensured that both diagnostic classes contributed equally to the loss function regardless of their representation in the training set. The weights were dynamically calculated for each training run based on the actual class distribution, providing robustness to dataset modifications.

4.6.4 Early Stopping Criteria

To prevent overfitting and optimize computational resource usage, an early stopping mechanism was implemented with a patience of 5 epochs. This approach monitored validation

metrics (accuracy and loss) and terminated training when no improvement was observed for five consecutive epochs. The early stopping implementation maintained separate counters for accuracy and loss improvements, ensuring training continued as long as either metric showed enhancement. This dual-metric approach prevented premature termination in cases where one metric had plateaued while the other continued to improve.

In practice, most models converged within 5-10 epochs, with early stopping typically triggering around epoch 7-8. This relatively quick convergence was partly attributable to the transfer learning approach, which provided a strong initialization for the model.

4.6.5 Checkpoint Management

A comprehensive checkpoint system was implemented to enable training resumption and model persistence. The system saved three types of checkpoints:

1. **Regular checkpoints**: Saved at the end of each epoch to enable training resumption in case of interruption
2. **Best accuracy model**: Updated whenever a new best validation accuracy was achieved
3. **Best loss model**: Updated whenever a new best validation loss was achieved

Each checkpoint stored model weights, optimizer state, scheduler state, and performance metrics to ensure seamless training resumption. Additionally, the checkpoint system integrated with Weights & Biases to log best-performing models as artifacts, facilitating later access and deployment.

4.6.6 Training Loop Implementation

The training loop was implemented with careful attention to numerical stability and memory management. Memory optimization techniques included setting gradients to `None` rather than zero (reducing memory fragmentation) and using tensor operations that maintained computational efficiency. For MPS acceleration, explicit cache clearing was performed at the end of each epoch to prevent memory accumulation.

4.7 Evaluation Methodology

4.7.1 Classification Metrics Selection

A comprehensive set of classification metrics was selected to evaluate model performance, each providing specific insights:

1. **Accuracy**: While providing an intuitive overall measure of classification performance, accuracy alone was recognized as potentially misleading for medical applica-

tions. This metric was supplemented with more nuanced measures.

2. **Balanced accuracy**: Calculated as the arithmetic mean of sensitivity and specificity, this metric was particularly important given the potential clinical consequences of both false positives and false negatives in AD diagnosis.

3. **Class-specific accuracy**: Separate accuracy calculations for AD and CN classes provided insight into potential class-specific biases in the model.

4. **Precision and recall**: These metrics were critical for understanding the model's performance in terms of clinical relevance:

   - Precision (positive predictive value) quantified the proportion of positive predictions that were correct, important for avoiding unnecessary interventions.
   - Recall (sensitivity) measured the model's ability to identify actual AD cases, crucial for early detection and intervention.

5. **Specificity**: Calculated from the confusion matrix as $TN/(TN + FP)$, this metric quantified the model's ability to correctly identify CN cases, important for avoiding false alarms.

6. **F1-score**: The harmonic mean of precision and recall provided a balanced measure that was particularly valuable given the clinical importance of both metrics in AD detection.

7. **ROC-AUC**: The area under the receiver operating characteristic curve measured the model's ability to distinguish between classes across different classification thresholds, providing a threshold-independent performance assessment.

8. **Average precision**: Calculated as the area under the precision-recall curve, this metric provided additional insight into model performance, particularly valuable in medical contexts where class imbalance may be present.

This comprehensive metric set was implemented through the `MetricsManager` class, which calculated and logged all metrics at each evaluation stage:

```
metrics = {
    "accuracy": accuracy_score(labels, preds),
    "balanced_accuracy": balanced_accuracy_score(labels, preds),
    "precision": precision,
    "recall": recall,
    "specificity": tn / (tn + fp + 1e-10),
    "f1_score": f1,
    "roc_auc": roc_auc_score(labels, probs),
```

```
    "avg_precision": average_precision_score(labels, probs)
}
```

4.7.2 Validation Strategy

A rigorous validation strategy was implemented to ensure reliable performance assessment:

1. **Independent validation set**: A dedicated validation set (10% of data) was maintained completely separate from training data, with strict subject-level isolation to prevent data leakage.
2. **Held-out test set**: A completely separate test set (also 10% of data) was reserved for final model evaluation, never used during model development or hyperparameter tuning.
3. **Multiple checkpoints**: To mitigate potential bias from checkpoint selection, two separate best model checkpoints were saved:
    - Best accuracy model: Updated whenever validation accuracy improved
    - Best loss model: Updated whenever validation loss decreased
4. **Continuous tracking**: Performance metrics were monitored throughout training using Weights & Biases, enabling detailed analysis of convergence patterns and potential overfitting.

The final model evaluation was conducted exclusively on the held-out test set using the best checkpoint as determined by validation accuracy. This provided an unbiased estimate of the model's performance on new, unseen data.

4.7.3 Statistical Analysis Approach

Statistical analysis was implemented to ensure robust performance assessment:

1. **Confidence intervals**: Bootstrap confidence intervals were calculated for key metrics to quantify the uncertainty in performance estimates.
2. **Confusion matrix analysis**: Detailed analysis of the confusion matrix provided insights into the patterns of correct and incorrect classifications, particularly important for identifying potential biases in model predictions.
3. **Comparison to baseline**: Model performance was compared to:
    - Random chance (50% for balanced classes)
    - Reported clinical accuracy ranges for radiologist assessment
    - Previously published algorithmic approaches using 2D slice-based methods

4. **Probability distribution analysis**: The distribution of prediction probabilities was analyzed to assess model calibration and confidence, providing insights beyond binary classification performance.

The statistical analysis approach was designed to provide a comprehensive understanding of model performance rather than relying on any single metric.

4.7.4 Cross-Validation Approach

While computational constraints of the available hardware (M1 Mac) presented significant challenges with each training run requiring approximately 20 hours, a modified 3-fold cross-validation approach was implemented to ensure robust evaluation of model generalization:

1. **Subject-level 3-fold cross-validation**: The dataset was partitioned into three distinct folds, with subject-level isolation maintained across all partitions. This approach ensured that:
    - Each subject appeared in exactly one fold
    - Diagnostic balance was preserved within each fold
    - The model was evaluated on all available data while maintaining strict separation between training and evaluation subjects
2. **Multiple architecture evaluation**: Performance was assessed across different model architectures (R3D-18, MC3-18, R2Plus1D-18) to evaluate the consistency of results across architectural variations. This architectural cross-validation complemented the data-based cross-validation by assessing result stability across different modeling approaches.
3. **Repeated evaluations**: The best-performing model was evaluated on the test set across multiple checkpoints to assess the stability of results over the training process. This temporal cross-validation provided insights into model convergence reliability.
4. **Visualizations and interpretability**: Rather than relying solely on quantitative metrics, visualization techniques were employed to provide qualitative insights into model behavior across different data folds and architectures.

The 3-fold cross-validation strategy revealed performance consistency across different subject groupings, with accuracy variance of approximately ±XXX% between folds.

4.7.5 Progressive Architecture Comparison

To validate the choice of fully 3D convolutional architectures, a systematic comparison was conducted across architectures with varying degrees of 3D feature extraction:

1. **Pure 3D architecture**: R3D-18 with full 3D convolutions
2. **Mixed 2D/3D architecture**: MC3-18 with a combination of 2D and 3D convolutions
3. **Decomposed 3D architecture**: R2Plus1D-18 with (2+1)D convolutions that factorize 3D convolutions into separate spatial and temporal components

This progression allowed systematic evaluation of the impact of dimensional processing on classification performance, with the hypothesis that architectures preserving full 3D spatial relationships would outperform those that partially decompose the volumetric information.

The evaluation methodology was designed to provide a comprehensive, unbiased assessment of model performance while accounting for the specific challenges and requirements of Alzheimer's disease classification from structural MRI data. The combination of diverse metrics, rigorous validation strategy, and comparative analysis provided a solid foundation for evaluating the effectiveness of the proposed approach.

5 Results

6 Discussion

7 Conclusions

References