Early Diagnosis of Alzheimer’s Disease Using AI Models on 3D MRI Neuroimaging Data

Ahmed Abdullah Shahid

Student ID: 24012028

School of Mathematics, Computer Science & Engineering

Liverpool Hope University

Supervisor: Dr. Atif Mehmood

LIVERPOOL HOPE
UNIVERSITY

Abstract

Background and Objective: Early finding of Alzheimer's disease (AD) and mild cognitive impairment (MCI) from structural MRI is hard. Changes in the brain can be small, classes are imbalanced, and patients progress at different speeds. This study develops and test deep learning models that classify T1-weighted MRI into two useful groups: AD vs cognitively normal (CN), and CN vs MCI.

Methods: This study uses the OASIS-3 dataset (number of samples for AD vs CN is 1,289; number of samples for CN vs MCI is 526). For AD vs CN, a 3D DenseNet-121 is trained to use full brain volumes and capture wide atrophy patterns. For CN vs MCI, a lighter 2.5D ConvNeXt-Tiny model with ImageNet pretraining is employed; it reads small axial "slabs" to save memory while learning useful features. Preprocessing includes HD-BET skull Stripping, z-score normalization, and standard resampling to a fixed size. To handle class imbalance, balanced batches, class-weighted losses, and a combination of focal loss and cross-entropy are used for the CN vs MCI task. Data is split by subject, and only the latest scan per subject is retained to avoid leakage. Operating thresholds are selected on the validation set using the ROC curve and are never tuned on the test set.

Results: For AD vs CN, the model reaches 87.11% accuracy and 84.59% balanced accuracy on the held-out test set. At the accuracy-focused threshold, specificity is 92.25%. For CN vs MCI, the model achieves 87.34% accuracy and 80.07% balanced accuracy. It shows high specificity (90.91%) and high negative predictive value (93.8%), while sensitivity is moderate (69.23%). Test-time augmentation and exponential moving average of weights make predictions more stable. These results are in line with, or better than, many reported ADNI-based studies, noting that datasets and settings differ.

Conclusion: Well-designed deep learning pipelines can classify Alzheimer-related changes from T1 MRI with strong and reliable performance. In the tests, AD vs CN shows high accuracy and balance, and CN vs MCI works well as a triage tool due to its high specificity and NPV. The process is transparent and reproducible: subject-level splits, fixed validation-based thresholds, and clear reporting of accuracy, balanced accuracy, AUC, sensitivity, specificity, and predictive values. Future work should add more data types (e.g., PET, cognitive scores) and use longitudinal information to improve early-stage sensitivity.

Keywords: Alzheimer's disease; mild cognitive impairment; deep learning; structural MRI; 3D CNN; medical image classification.

Table of Contents

List of Figures:

List of Tables:

List of Abbreviations

Abbreviation	Full Term
AD	Alzheimer’s Disease
CN	Cognitively Normal
MCI	Mild Cognitive Impairment
MRI	Magnetic Resonance Imaging
ADNI	Alzheimer’s Disease Neuroimaging Initiative
OASIS	Open Access Series of Imaging Studies
AIBL	Australian Imaging, Biomarkers & Lifestyle study
NITRC	Neuroimaging Informatics Tools and Resources Clearinghouse
NIfTI	Neuroimaging Informatics Technology Initiative
CNN	Convolutional Neural Network
LR	Learning Rate
EMA	Exponential Moving Average
TTA	Test-Time Augmentation

Chapter 1: Introduction

Alzheimer's disease (AD) is one of the most pressing health challenges of our time, affecting memory, thinking and daily life for millions of people and placing a heavy burden on families and healthcare systems. Early and reliable detection is crucial because it can support timely care, guide treatment plans, and help select suitable participants for clinical trials. Structural magnetic resonance imaging (MRI) is widely used in hospitals and research, and it captures subtle brain changes related to neurodegeneration. Recent advances in artificial intelligence (AI) offer a way to read these patterns more precisely than traditional methods. However, building models that are both accurate and trustworthy is difficult. Brain anatomy varies between individuals, disease progression is gradual and heterogeneous, and images are acquired with different scanners and protocols. In addition, class imbalance, especially the smaller number of Mild Cognitive Impairment (MCI) cases compared to Alzheimer's (AD) and Cognitively Normal (CN) groups makes learning stable decision boundaries harder. These issues can lead to models that perform well in one split but struggle to generalize.

This project focuses on developing and evaluating AI models for classifying 3D brain MRI scans into diagnostic categories relevant to clinical practice. Using the OASIS-3 dataset, which includes longitudinal T1-weighted MRI for CN, MCI, and AD subjects, the work explores binary classification. The binary tasks AD vs CN and CN vs MCI target practical decision points that clinicians often face. Because MCI is a transitional stage and its imaging signatures can be subtle, it is usually the most challenging class. This project therefore examines data preparation and training strategies designed to improve learning from limited and imbalanced samples. The pipeline includes standard neuroimaging preprocessing (orientation, intensity normalization, and skull stripping), careful splitting to avoid data leakage, and augmentation to improve robustness. Model performance is assessed with accuracy, balanced accuracy, F1-scores and confusion matrices to capture both overall and per-class behavior.

The wider context of this work is the need for reliable tools that can assist clinicians in making earlier and more consistent decisions. While advanced deep learning architectures can extract meaningful features from volumetric MRI, they must be trained and validated with attention to fairness and stability. This means accounting for class imbalance, avoiding overfitting, and calibrating output probabilities so thresholds can be tuned to reflect clinical goals (for example, prioritizing sensitivity). It also means reporting results that go beyond a single metric, since performance can look different for majority and minority classes. By grounding the study in a well-known public dataset and using transparent evaluation, the project aims to provide a clear picture of what current AI methods can achieve and where their limits remain.

The main goal of the project is to build a robust classification framework for the early detection of Alzheimer's-related changes from structural MRI and to understand the factors that influence performance across tasks and classes. The specific objectives are:

- To analyze and prepare the OASIS-3 MRI data with consistent preprocessing and careful, stratified splits.
- To design and train deep learning models for binary (AD vs CN, CN vs MCI) classification.
- To handle class imbalance with balanced sampling and other training strategies, and to evaluate with balanced accuracy and class-wise metrics.
- To study decision thresholds and report clear confusion matrices that reflect clinical trade-offs.
- To compare outcomes between different binary settings and discuss the impact of limited MCI samples.
- To summarize limitations and outline practical next steps for improving generalization and clinical usefulness.

By addressing these aims, the project seeks to offer an evidence-based view of how AI can support earlier detection of Alzheimer's disease from MRI, where it performs well, and what remains challenging especially in distinguishing MCI from adjacent diagnostic groups.

Chapter 2: Literature Review

Alzheimer's disease (AD) detection from structural MRI has been studied for more than two decades, moving from traditional feature-engineering to end-to-end deep learning. Public datasets made most AD–MRI work possible. OASIS-3 offers longitudinal T1-MRI with clinical and cognitive data across normal aging and AD, with repeated visits (LaMontagne et al., 2019). ADNI is a large multi-site study with standardized MRI, follow-up of CN, MCI, and AD participants, and a focus on biomarker discovery and progression (Weiner et al., 2017). AIBL is an Australian cohort that combines imaging, biomarkers, and lifestyle measures, and includes serial T1-MRI with cognitive testing (Ellis et al., 2009). Together, these cohorts provide longitudinal scans, clear labels, and multi-site variation needed to train and fairly test modern models. Most studies first asked a simple and clinically useful question: can a model tell AD from CN at the subject level using one or more MRI sessions? Many papers then examined a second question that matters for early care: can a model tell CN from MCI? Performance in both binary tasks depend on data quality, preprocessing, class balance and fair evaluation design.

Early machine-learning pipelines used hand-crafted features. Common approaches measured hippocampal and entorhinal volumes, cortical thickness, or voxel-based morphometry maps, and then trained support vector machines or logistic regression to separate groups (Cuingnet et al., 2011; Klöppel et al., 2008). These models were simple to interpret and worked well on small, single-site datasets. However, they relied on careful feature design and could miss distributed and subtle atrophy patterns, especially around the CN–MCI boundary. As larger datasets became available, researchers moved toward representation learning, where the model learns features directly from scans, reducing manual choices and capturing wider 3D context (Suk et al., 2014; Payan & Montana, 2015).

Deep learning now dominates AD vs CN on T1 MRI. Early CNN systems used 2D slices or multi-view stacks to reduce memory cost; later works adopted fully 3D networks (i.e. 3D-ResNet, 3D-DenseNet) to exploit volumetric structure and long-range context (Payan & Montana, 2015; Korolev et al., 2017). Several studies report strong AD vs CN accuracy with 3D architectures when trained with consistent preprocessing and subject-level splits (Basaia et al., 2019; Wen et al., 2020). Beyond plain classification, some groups used patch-based inputs that focus on medial temporal lobe, or added attention modules to highlight informative anatomy, which can improve sensitivity to hippocampal and entorhinal changes (Liu et al., 2018; Thung et al., 2019). Transfer learning from natural-image CNNs gives mixed results because MRI differs from RGB photos, but pretraining on large medical volumes or using self-supervised objectives can help when labeled data are limited (Chen et al., 2021). Overall, end-to-end 3D CNNs with solid augmentation and careful validation tend to reach high AD vs CN accuracy on single-site splits; however, results may drop under scanner or site shift, which remains a key challenge for deployment (Wen et al., 2020).

CN vs MCI. This task is clearly harder than AD vs CN because the brain changes in MCI are small and vary across people, and some MCI cases later progress to AD, so labels can change over time (Querbes et al., 2009). Classical MRI pipelines that use cortical thickness and hippocampal volumes can separate groups, but their CN vs MCI accuracy is only moderate compared with AD vs CN (Cuingnet et al., 2011). Modern deep models help but do not close the gap: 3D CNNs and patch/medial-temporal-focused designs improve results a little yet still report lower scores for CN vs MCI than for AD vs CN (Korolev et al., 2017; Basaia et al., 2019). Adding simple non-imaging features such as age and cognitive tests (e.g., MMSE) or using longitudinal MRI can boost performance near the CN–MCI boundary, because changes over time carry useful signal (Moradi et al., 2015). Reviews that compare many pipelines also underline that careful, subject-level validation is essential, and they repeatedly observe lower performance for CN vs MCI than for AD vs CN.

Preprocessing is a critical part of both binaries and can change outcomes. Typical steps include reorientation to standard axes, resampling to fixed voxel size, bias-field correction (e.g., N4), intensity normalization, skull stripping and optional registration to MNI space. These steps reduce variance between subjects and scanners, so the model learns disease patterns rather than site artifacts. In multi-site data, harmonization methods such as ComBat reduce between-site variation while keeping biological signal and are widely used in structural MRI (Fortin et al., 2018). For deep learning, harmonization is often combined with realistic spatial and intensity augmentations (small flips, small rotations, mild scale changes). Test-time augmentation and simple model ensembling can further stabilize predictions in small-to-mid-sized cohorts (Korolev et al., 2017).

Data splitting and leakage control are central to fair evaluation. Some early studies split at the slice level, which allowed slices from the same subject to appear in both training and testing, inflating accuracy. Best practice is strict subject-level splitting: all scans from one subject must appear in only one split (train, validation, or test). For longitudinal cohorts, session-level care is needed so future visits of a training subject do not leak into validation or test. Reviews emphasize these points and show that reported performance drops when leakage is removed

(Wen et al., 2020). When datasets are small, k-fold cross-validation at the subject level is common, but a held-out test set (or an external dataset) should be kept for final reporting.

Choice of metric also matters. Plain accuracy can hide bias when classes are imbalanced (e.g., CN > MCI). Strong studies report balanced accuracy (mean of sensitivity and specificity), per-class F1 scores, ROC-AUC, and confusion matrices. These metrics reveal how the model treats each group and help set thresholds. Probability calibration is useful when thresholds will change with clinical goals (e.g., screen with high sensitivity, then refer for further tests). Temperature scaling is a simple method to improve calibration for deep networks (Guo et al., 2017). Following these practices, many works pick thresholds on validation data by sweeping the ROC or F1 curves, then fix the threshold and report final test results with confusion matrices and balanced metrics.

Regularization and training stability are widely discussed. Common tools include weight decay, dropout, balanced batch sampling, and early stopping on validation. Gradient clipping prevents exploding updates. MixUp reduces overfitting and encourages smoother decision boundaries by blending images and labels; focal loss emphasizes hard or minority samples; and class-balanced loss down-weights frequent classes by their "effective" sample size (Zhang et al., 2018; Lin et al., 2017; Cui et al., 2019). Many studies also use an exponential moving average (EMA) of model weights to stabilize validation; the EMA "teacher" often performs more smoothly than the raw "student" network (Tarvainen & Valpola, 2017). Test-time augmentation (TTA) averages predictions across flipped/rotated versions of the same scan and commonly gives small but consistent gains. These methods are helpful for both AD vs CN and CN vs MCI, especially when sample sizes are limited.

Model design choices follow a practical pattern across papers. For AD vs CN, many studies use full-volume 3D CNNs such as 3D-DenseNet and 3D-ResNet. These models view the whole brain and capture global atrophy patterns in T1 MRI. With consistent preprocessing and strict subject-level splits, they are reported as reliable and often state-of-the-art; light attention or small ensembles can add a bit more gain (Korolev et al., 2017; Wen et al., 2020). Training setups commonly use AdamW and cosine learning-rate schedules with warm-up (Loshchilov & Hutter, 2019).

For CN vs MCI, the task is harder because they are difficult to separate. A common strategy is to use strong 2D ImageNet backbones with 2.5D multi-slice ("slab") inputs, which keep key medial-temporal cues while staying efficient (Roth et al., 2014; Liu et al., 2022; Tan & Le, 2021). ConvNeXt and EfficientNet-V2 are frequent choices. To handle imbalance and stabilize training, many papers add MixUp/CutMix and class-aware losses such as focal loss with label smoothing, which can raise minority-class recall (Lin et al., 2017). Lightweight add-ons like EMA/Mean-Teacher and test-time augmentation also help produce smoother, more stable predictions.

Longitudinal information improves sensitivity to small changes, especially for CN vs MCI. Some studies model trajectories (e.g., change in hippocampal volume) or feed pairs of scans to the network to learn progression signals. Labels such as "MCI converters" versus "non-converters" are also studied, but these tasks require larger cohorts and consistent follow-up (Moradi et al., 2015; Thung et al., 2019). When repeated scans are limited, strict subject-level splitting and careful augmentation remain the safer path.

Generalization across cohorts is an open problem. Models trained on one site can drop in accuracy on another site due to scanner, protocol, or population differences. Harmonization (e.g., ComBat), careful normalization, and robust augmentations reduce the gap, but external validation still shows that performance may fall outside the confidence interval seen on internal validation (Fortin et al., 2018; Wen et al., 2020). For clinical use, calibration and site-specific threshold tuning are often needed. Decision-curve analysis is a useful but under-used tool to compare the net benefit of using a model versus simple rules across a range of threshold probabilities, which can support real policy choices (Vickers & Elkin, 2006).

Taken together, the literature suggests a clear set of good practices for the two binary tasks:

- AD vs CN. This is the relatively easier task. With standard preprocessing and harmonization, subject-level splits, 3D CNNs with dropout and weight decay, balanced sampling, EMA, and realistic augmentations (plus TTA), many studies report high accuracy on single-site data. Threshold selection on validation improves the balance between sensitivity and specificity. Final test reports should include confusion matrices, balanced accuracy, per-class F1, and ROC-AUC.

- CN vs MCI is clearly harder than AD vs CN. The brain changes in MCI are small, vary a lot between people and some MCI cases later progress to AD, so labels can change over time. Because of this, most papers report that CN vs MCI accuracy is lower than AD vs CN, often by about 10–20 percentage points, even when they use the same data and model design. For example, studies that reach around 75–90 percent on AD vs CN often get only about 65–78 percent on CN vs MCI. Balanced accuracy for CN vs MCI is also usually lower, and sensitivity to MCI often sits in the mid-60s to low-80s. Classic MRI features like cortical thickness and hippocampal volume help, but they still perform better on AD vs CN than on CN vs MCI. Modern deep models improve things a bit, yet the gap remains. Many works suggest using class-aware losses, balanced sampling, and careful threshold tuning, and adding simple clinical data or longitudinal scans can help. Overall, as seen in most literature, researchers still struggle to achieve high accuracy for CN vs MCI.

In summary, research has moved from hand-crafted features to strong 3D deep learning for MRI-based diagnosis. Good practice is clear: split data by subject to avoid leakage; keep preprocessing simple and transparent (and harmonize across sites when needed); handle class imbalance with balanced sampling and class-aware loss; use regularization like dropout; and keep validation stable with EMA and test-time augmentation. It also helps to calibrate probabilities and report fair metrics, including confusion matrices, not just accuracy. For real-world trust, we need basic interpretability and external validation on new datasets. These lessons guide today’s MRI-only pipelines on public data such as OASIS-3 for both AD vs CN and CN vs MCI. Still, CN vs MCI is the hardest boundary, so careful methods and realistic expectations are important. My main objective is to improve accuracy on the CN vs MCI task to support earlier detection of Alzheimer’s disease, since catching changes at this stage can help clinical decisions and patient care sooner.

Chapter 3: Methodology

3.1 Research Objectives

Supervised deep learning is employed to predict two binary labels from structural T1-MRI: AD vs CN and CN vs MCI. One model is trained per task, implemented as two separate scripts tuned to each problem. This choice is simple and safe: the two tasks have different signal patterns and different data balance, so a single shared script would force many compromises. A split design enables selection of the right input shape, backbone, loss, sampler, and schedule for each task. Prior work also shows that AD vs CN benefits from full-volume 3D context, while CN vs MCI often needs lighter, more regularized models that focus on small, local changes. Using separate scripts fits this difference and helps generalization.

Across both tasks the pipeline remains stable: careful preprocessing, clean subject-level splits, strong but safe augmentation, explicit class-imbalance handling, and validation-driven threshold choice. A test set (~15%) is always held out from the start and all tuning is performed on the validation set only. Final results are then reported once on the fixed test split with confusion matrices and balanced metrics.

3.2 Data Source

All MRI data comes from OASIS-3, accessed directly from the official NITRC website after registration, agreement to the data-use terms and access request for this master's project (NITRC, n.d.). Labels are the standard clinical groups provided by OASIS-3: Cognitively Normal (CN), Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD). This work uses only structural T1-weighted MRI (no PET/CSF or other modalities). Data are de-identified; no re-identification is attempted, and results are reported in aggregate.

3.3 Data Splitting

The unit of analysis is the subject/session. In this layout, each NIfTI file represents one MRI session. Subject-level splits are created to avoid leakage: scans from the same person never appear in more than one split. If more than one scan is available for a subject, the latest scan is selected. This approach avoids mixing disease stages for that subject, prevents time-based leakage from earlier to later scans, and maintains one independent sample per subject. A held-out test set (~15%) is reserved that is never used for training or threshold tuning, and the remainder is split into train and validation with stratification by class.

3.4 Task-Specific Implementation Strategy

one script for each task because the tasks differ in signal scale, class balance, and the risk of overfitting:

- AD vs CN has stronger global atrophy patterns and benefits from 3D full-brain context. A 3D DenseNet can model long-range structure across the whole volume. It also needs 3D-aware augmentation and a 3D preprocessing chain.

- CN vs MCI is harder. Differences are subtle and often local. A 2.5D slab with a strong 2D pretrained backbone (like ConvNeXt-Tiny) gives more regularization, larger batches, and stable optimization. It also allows class-aware losses and sampling without heavy memory cost.

Keeping separate scripts lets me set different transformations, losses, samplers and schedules without fragile flags. It also reduces mistakes and makes the experiments reproducible.

3.5 Preprocessing (Before Training)

The offline preprocessing has two steps and matches the scripts exactly.

3.5.1 Skull Stripping with HD-BET (Brain Extraction)

HD-BET is run from the command line and write one output per input: a brain-only image named with the suffix *_SS.nii.gz. No mask file is saved. The device (GPU/CPU) is chosen automatically. A cleanup pass is also run to remove any left-over files that look like masks (names ending in _SS_bet.nii.gz, _bet.nii.gz, or _mask.nii.gz) only if the paired *_SS.nii.gz already exists. This keeps exactly one clean brain image per scan.

3.5.2 Standardization

For every NIfTI file, the volume is loaded, convert to float32 and do z-score normalization per volume (subtract mean and divide by standard deviation, with a small variance safeguard). The volume is then resized to 128×128×128 using linear interpolation (scipy.ndimage.zoom with order=1). The processed volume is saved as a new NIfTI with an identity affine (np.eye(4)). Saving with identity affine gives all volumes a uniform grid on disk; the online preprocessing in the training script then sets the desired spacing and field of view. This step is repeated for CN, MCI, and AD folders so all classes share the same shape and scale.

These two steps (HD-BET brain-only output + z-score + fixed-size resize) match the scripts: one brain image per input and uniform 128^3 volumes saved with identity affine.

3.6 Preprocessing and Augmentation (During Training)

Different online pipelines are used in the loaders because the model families are different.

AD vs CN:

Re-orientation to RAS is applied, resampling to 1.0 mm isotropic spacing (linear), percentile intensity scaling (0.5–99.5%) mapped to [0,1] with clipping, foreground crop to remove empty space, and resize/pad to 128×160×160. Light 3D augmentation is used: small flips on each axis, small rotations (~±8°), mild isotropic scaling (~±8%), and small translations. These are gentle and realistic for brain MRI.

CN vs MCI:

For each sample, 3D volume is Loaded and build a 3-channel 2.5D image from axial slices. The offset set [−12, −6, 0, +6, +12] is used around the mid-slice and randomly pick 3 offsets each epoch. For each slice, intensities are clipped to 1–99%, rescale to [0,1], and apply CLAHE (adaptive histogram equalization) to gently improve contrast. The three slices are stacked into one 3-channel image and resize to 256×256. Random crops, horizontal flip, small rotation (≈10°), and light color jitter (brightness/contrast) are then applied. Finally, normalization with ImageNet mean and std is performed to match the 2D backbone pretraining.

3.7 Model Choice

AD vs CN: a 3D DenseNet from MONAI with growth rate 32 and block config 6-12-24-16. The network outputs two logits (AD and CN). For a stable binary loss, these are converted to a single-logit margin (AD − CN) and BCEWithLogits is optimized on that margin. Configurable dropout is applied inside DenseNet blocks to control capacity. A 3D design is used because AD-related atrophy is distributed and benefits from full volumetric context.

CN vs MCI: a 2.5D ConvNeXt-Tiny backbone (ImageNet-pretrained). The classifier is replaced with a regularized MLP head (LayerNorm/BatchNorm, GELU, Dropout) that outputs two logits (CN, MCI). An option for EfficientNet-V2-S with a similar head is also included, but ConvNeXt-Tiny is the default script. A 2.5D setup is chosen because the cues are subtle and local, and 2D backbones enable larger batches and stronger regularization.

3.8 Loss Functions and Class Imbalance

AD vs CN. BCEWithLogits is applied to the single-logit margin and, when classes are imbalanced, a positive-class weight is used to reduce bias toward the majority class:

$$pos - weight = \max\left(1.0, \left(\frac{n_{CN}}{n_{AD}}\right)^{1.25}\right)$$

This weight is ON in Phase-1 and OFF in Phase-2 (fine-tuning).

CN vs MCI. A Combined Loss is used to match the implementation: Focal Loss with α=0.85, γ=1.5, and label smoothing is equal to 0.05 (to gently up-weight MCI and focus on harder samples), plus Cross-Entropy with the same label smoothing. The final objective is

$$\mathcal{L} = 0.7 \times \mathcal{L}_{Focal} + 0.3 \times \mathcal{L}_{CE}$$

For sampling, dynamic class weights are computed from the training labels and passed to a WeightedRandomSampler so MCI appears more frequently in mini batches. In addition, CN is globally capped upfront when it dominates the cohort (e.g., cap of 440). Together, these two steps are the primary mechanisms used to handle imbalance.

3.9 Regularization and Stability

Several standard tools are combined because they work well in small/medium MRI cohorts. Dropout (in backbone and head), weight decay (AdamW) and gradient clipping (norm = 1.0) are applied. MixUp and CutMix are used where they help most:

- AD vs CN: MixUp ON in Phase-1 to build margins, OFF in Phase-2 to clean the boundary.
- CN vs MCI: MixUp (α = 0.2) and CutMix (p = 0.2) during training, with a small ramp-up of augmentation probability across early epochs.

For AD vs CN, an EMA copy of the model (decay 0.999) is maintained and use it for validation snapshots; EMA predictions are usually smoother. At test time TTA (flips and 90° rotations) is applied and average logits.

3.10. Optimizers and Learning-Rate

AdamW is used in both tasks but different schedules:

- AD vs CN: custom warm-up + cosine schedule (LambdaLR). Training used mixed precision (AMP) and gradient accumulation to simulate a larger batch when memory is tight (per-step batch 4, effective batch ≈ 8).
- CN vs MCI: OneCycleLR with 10% warm-up and cosine anneal. The base LR is 3e-5, weight decay 1e-4, and training runs up to 60 epochs with early stopping (patience 15, min-delta 0.001). Batch size is 32 if memory allows.

3.11 Two-Phase Training for AD vs CN

AD vs CN is trained in two phases:

- Phase-1 (baseline): LR 5e-5, dropout 0.0, class weight ON, MixUp ON. This grows stable margins and helps balance.
- Phase-2 (fine-tune): LR 5e-6, dropout ≈ 0.10, class weight OFF, MixUp OFF. The backbone can be frozen and only the classifier trained for extra stability.

After each phase, Thresholds are swept on the validation set and save the best thresholds (max Accuracy and max Balanced Accuracy) to small JSON files for deployment. The held out test set is then evaluated once at those fixed thresholds.

3.12 Batching and Samplers

For AD vs CN, a BalancedBinaryBatchSampler is used that makes each mini-batch 50% AD and 50% CN. This stabilizes training even if the dataset is not perfectly balanced. For CN vs MCI, the WeightedRandomSampler with dynamic weights (and the CN cap upfront) is used to increase the presence of MCI during training.

3.13 Thresholds and Metrics

Thresholds are always selected on validation, never on tests. This reduces optimistic bias and supports fair reporting.

- AD vs CN. Validation probabilities are computed, and thresholds are swept to obtain two operating points: one maximizing Accuracy and one maximizing Balanced Accuracy. On the test set, evaluation is performed once at each fixed threshold. Reported metrics include Accuracy, Balanced Accuracy, per-class F1, ROC-AUC (CN probability), confusion matrices, and the classification report. TTA is applied at test time to stabilize results.

- CN vs MCI. The MCI probability is computed, and an ROC curve is built on validation. Thresholds are swept and the threshold that maximizes Balanced Accuracy is selected. On the test set, this threshold is fixed, and the following metrics are reported: Accuracy, Balanced Accuracy, Precision (weighted), Recall (weighted), F1 (weighted), ROC-AUC, and the confusion matrix. In addition, Sensitivity (MCI detection), Specificity (CN detection), PPV, and NPV are computed to clarify clinical trade-offs.

3.14 Reproducibility and Environments

Random seeds are fixed for Python, NumPy, and PyTorch in both scripts. In the 3D script, cuDNN is set to deterministic = True and benchmark = False to increase repeatability. In the 2.5D script, cuDNN benchmark = True is enabled to improve speed, accepting small run-to-run timing noise. In Colab, PYTORCH_DISABLE_DYNAMO=1 (and TORCHDYNAMO_DISABLE=1) is set to avoid known graph-capture issues in some builds. Full state dicts are saved (model, optimizer, scheduler, epoch, metrics). For AD vs CN, EMA weights are also saved. Library versions are printed, and checkpoints are stored in Google Drive.

3.15 Risks and Limits

Leakage is avoided with strict subject-level splits. Scanner effects are reduced through orientation/spacing standardization, intensity scaling, and brain extraction. Augmentations are kept mild to avoid distorting anatomy. It is acknowledged that CN vs MCI remains challenging and that labels can be noisy or evolve over time. Threshold selection on validation helps control the sensitivity/specificity trade-off. If permitted later, the approach can be extended to longitudinal change or augmented with simple clinical covariates to improve performance near the CN–MCI boundary.

The strategy is clear: two task-specific scripts; 3D full-volume modeling for AD vs CN; 2.5D slab modeling for CN vs MCI; HD-BET skull stripping; z-score + 128^3 resize; MONAI 3D transforms or 2.5D slice processing; class-aware losses and samplers; MixUp/CutMix where useful; EMA and TTA for the 3D task; OneCycleLR or warm-up + cosine; validation-driven thresholds; and balanced metrics on a held-out test. This plan matches implementation and the needs of each task.

Chapter 4: Research Methods

The following sections explain each part of the research design in plain language and tie it directly to the implemented code.

4.1 Settings and Tools

All experiments are run in Google Colab on a single NVIDIA GPU. Two environment flags PYTORCH_DISABLE_DYNAMO = 1 and TORCHDYNAMO_DISABLE = 1 are set so PyTorch does not attempt experimental graph capture that can break in Colab. Implementation is in Python, using PyTorch as the deep learning framework. MONAI is used for 3D medical image transforms and the 3D DenseNet backbone; torchvision supports the 2D/2.5D pipeline (ConvNeXt). scikit-learn provides dataset splits and metrics. NiBabel reads NIfTI files, while interpolation and contrast equalization rely on SciPy / OpenCV / scikit-image–style steps. tqdm is used for progress bars. Google Drive is mounted for reading/writing datasets and checkpoints.

These choices are evident at the top of both scripts where libraries are imported, Drive is mounted, and device/version information is printed.

4.2 Dataset Acquisition and Organization

OASIS-3 T1-weighted MRI data was obtained from NITRC. On Drive, three class folders are maintained (CN, MCI, AD), with each NIfTI file representing one MRI session. When subjects have multiple sessions, only the latest session is retained to avoid mixing disease stages for the same individual. In code, the class folders are scanned to build file-path/label lists, followed by subject-level stratified splits into train/validation/test. For CN vs MCI, the CN count is optionally capped (e.g., 440) prior to splitting when CN substantially exceeds MCI, reducing imbalance and training variance. File discovery, class counts, CN capping, and stratified splits are implemented in the dataset setup blocks.

4.3 Preprocessing

Each MRI scan was prepared in two steps before training: standardization (normalize + resize) and skull stripping.

4.3.1 Standardization

Each NIfTI is loaded with NiBabel and converted to float32. Z-score normalization is applied per volume (mean subtraction and division by standard deviation), followed by resampling to 128 × 128 × 128 using linear interpolation. The result is saved as a new NIfTI with an identity affine (np.eye(4)) into structured class-specific output folders. This preprocessing standardizes scale and size on disk, which speeds up training, reduces memory issues, and makes model batches consistent.

4.3.2 Skull Stripping

After standardization, HD-BET is run on every input NIfTI to remove non-brain tissue as shown in figure 1. The script invokes HD-BET on the GPU and does not request a mask, producing exactly one brain-extracted file per scan with a unified suffix (_SS.nii.gz). A small cleanup step deletes any leftover mask-related files (e.g., _SS_bet.nii.gz, _bet.nii.gz, _mask.nii.gz) only if the paired (_SS.nii.gz) already exists. If a skull-stripped file is already present, the script skips reprocessing. This yields a consistent one file per scan, brain dataset across CN, MCI, and AD folders, with clear logs of how many files were converted, skipped, or failed.

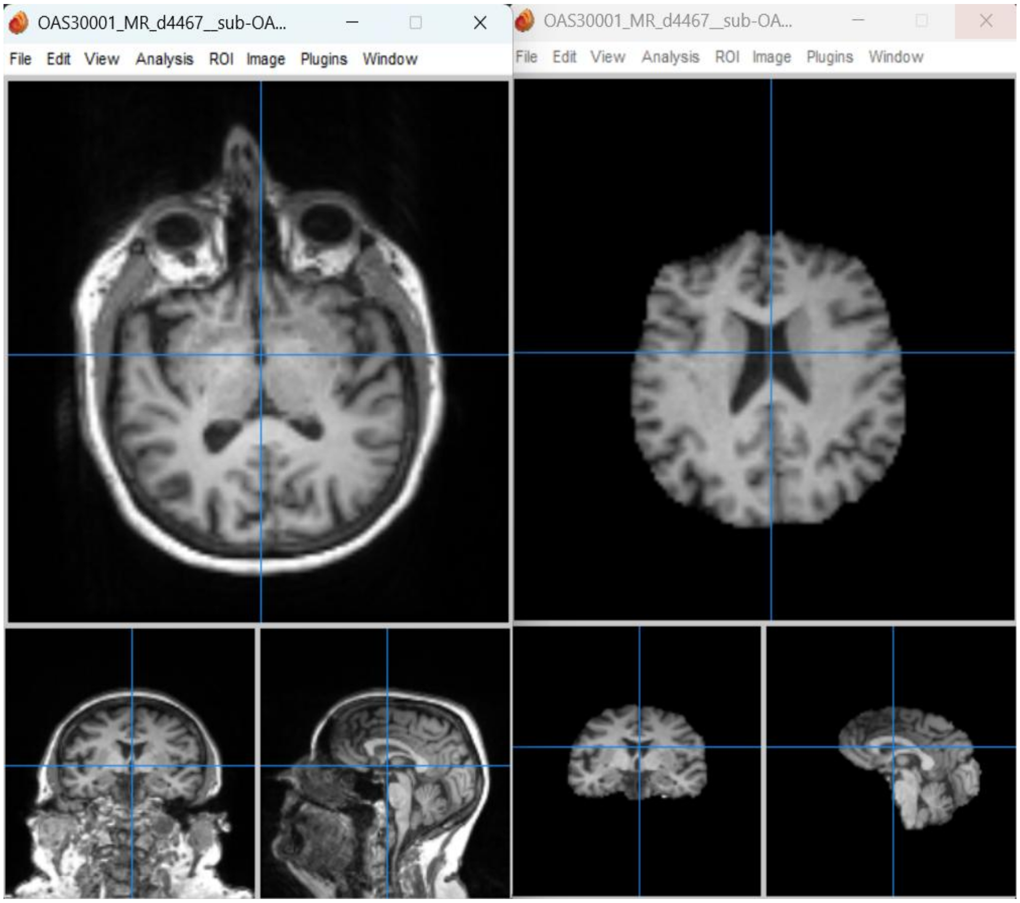

OAS30001_MR_d4467__sub-OA...
File Edit View Analysis ROI Image Plugins Window
OAS30001_MR_d4467__sub-OA...
File Edit View Analysis ROI Image Plugins Window

Figure 1: Preprocessed and skull-stripped T1-weighted MRI volume from OASIS-3 following spatial normalization, intensity correction, and brain extraction.

4.4 Global Split Policy

A held-out test set (~15%) is created once at the start and not used again until the end. From the remaining data, a validation set (~20% of train+val) is sampled with stratification. All splits are at the subject level to prevent leakage across sessions. This policy is implemented using train_test_split with stratify=labels and a fixed random seed.

4.5 Experiment 1 (AD vs CN)

4.5.1 Data loader and Transforms

3D brain volumes are loaded and apply a MONAI pipeline:

- enforce RAS orientation and 1.0 mm isotropic spacing,
- scale intensities to [0,1] using 0.5–99.5 percentiles,
- crop the foreground to remove empty space,
- resize/pad to a fixed 3D size (default 128×160×160).

Mild 3D augmentation is applied: small flips on each axis, rotations up to ~±8°, small translations, and ~±8% scale. These are safe for neuroanatomy and help generalization. All of this is visible in the train_tf and val_tf composed lists.

4.5.2 Model

MONAI's 3D DenseNet is used with block config (6,12,24,16) and growth rate 32. The network outputs two logits (AD, CN). For a clean binary loss, these two logits are converted to a single margin $\log it_{AD} - \log it_{CN}$ and pass it to BCEWithLogitsLoss with AD as the positive class. This let the system use a simple scalar threshold on P(AD) during validation and testing. The class and helper that do this are DenseNet3DHead plus two_to_single_logit.

4.5.3 Batching and Sampler

A BalancedBinaryBatchSampler is used to draw mini batches with 50% AD and 50% CN. This avoids collapse of the majority class and makes training stable. With batch size 4, gradient accumulation is used to reach an effective batch size of ~8. The sampler is coded from scratch and yields balanced index lists per batch.

4.5.4 Optimization and Schedules

Training uses AdamW, weight decay, AMP (mixed precision), and a warm-up + cosine learning-rate schedule. Gradients are clipped at norm 1.0. These choices help with stability on a single GPU and keep memory use low. The schedule is implemented as a custom LambdaLR with cosine decay after warm-up.

4.5.5 Class Imbalance Compensation Strategy

In Phase-1, a pos_weight for AD is enabled inside BCEWithLogitsLoss:

$$pos - weight = max\ (1.0, (n_{CN}/n_{AD}\ )\text{^}1.25\ )$$

This increases the loss for AD when AD is the minority. In Phase-2, the weight is turned off to chase peak accuracy once the model is already calibrated. The computation and the toggle appear in train_binary_fold_balanced.

4.5.6 Two-Phase Training

- Phase-1 (baseline). LR 5e-5, dropout 0.0, MixUp ON (α=0.2), class weight ON.
- Phase-2 (fine-tune). LR 5e-6, dropout ~0.10, MixUp OFF, class weight OFF. The backbone can optionally be frozen so that only the classifier head is trained. These knobs are all set in the CONFIG section and passed to the training function.

4.5.7 EMA and Model selection

An Exponential Moving Average (EMA) copy of the model (decay 0.999) is maintained, and validation is performed with the EMA weights because they are less noisy. The best checkpoint is saved using a composite key that favors AUC, then F1, then accuracy. The EMA class and the scoring key are coded right in the training loop.

4.5.8 Thresholds and Final Evaluation

After training each phase, validation probabilities P(AD) are collected, thresholds are swept, and record two operating points:

- Max Accuracy
- Max Balanced Accuracy

These thresholds are saved to JSON. For the final test, the best weights are reloaded, run test-time augmentation (TTA) (flips/rotations), and evaluate once at each fixed threshold. Accuracy, balanced accuracy, class F1, ROC-AUC, and the confusion matrix are reported. This full procedure appears in collect_p_ad_and_labels, sweep_thresholds_all, and evaluate_with_threshold.

4.5.9 Ensemble Methodology

TTA logits from the best Phase-1 and Phase-2 models can also be averaged. A threshold is re-selected on validation for the ensemble, then evaluate once on test. The code shows evaluate_ensemble and an ensemble_val_thresholds function to perform the sweep.

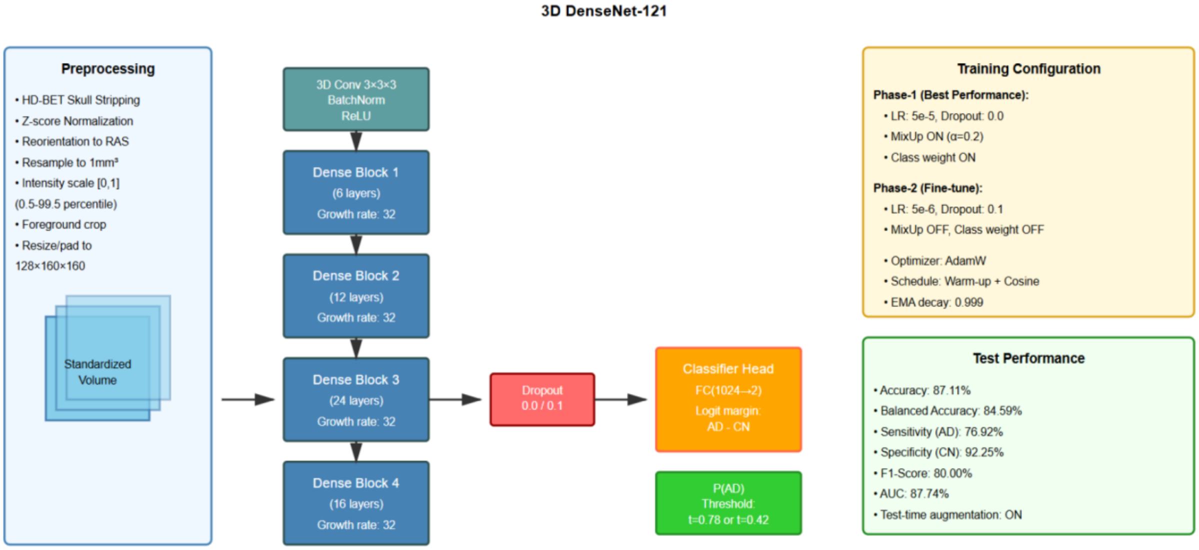

3D DenseNet-121
Preprocessing
• HD-BET Skull Stripping
• Z-score Normalization
• Reorientation to RAS
• Resample to 1mm³
• Intensity scale [0,1] (0.5-99.5 percentile)
• Foreground crop
• Resize/pad to 128×160×160
Standardized Volume
3D Conv 3×3×3 BatchNorm ReLU
Dense Block 1 (6 layers) Growth rate: 32
Dense Block 2 (12 layers) Growth rate: 32
Dense Block 3 (24 layers) Growth rate: 32
Dense Block 4 (16 layers) Growth rate: 32
Dropout 0.0 / 0.1
Classifier Head FC(1024→2) Logit margin: AD - CN
P(AD) Threshold: t=0.78 or t=0.42
Training Configuration
Phase-1 (Best Performance):
• LR: 5e-5, Dropout: 0.0
• MixUp ON (α=0.2)
• Class weight ON
Phase-2 (Fine-tune):
• LR: 5e-6, Dropout: 0.1
• MixUp OFF, Class weight OFF
• Optimizer: AdamW
• Schedule: Warm-up + Cosine
• EMA decay: 0.999
Test Performance
• Accuracy: 87.11%
• Balanced Accuracy: 84.59%
• Sensitivity (AD): 76.92%
• Specificity (CN): 92.25%
• F1-Score: 80.00%
• AUC: 87.74%
• Test-time augmentation: ON

Figure 2: Complete AD vs CN classification pipeline showing preprocessing workflow, 3D DenseNet-121 architecture with dense connectivity, two-phase training strategy and final test performance metrics.

4.6 Experiment 2 (CN vs MCI)

4.6.1 Class Imbalance Handling

Because CN is much larger than MCI, CN is capped (e.g., at 440 cases) to soften the imbalance. Inverse-frequency class weights are then computed, and MCI is given an extra boost (BASE_MCI_WEIGHT). A WeightedRandomSampler is built for the train loader, so MCI shows up more often. These steps keep training focused on the minority class without discarding MCI data. All details are inside the main() setup for this pipeline.

4.6.2 Data loader and Slabs

Each 3D volume is converted into a 2.5D slab: three axial slices near the center sampled each epoch from offsets [-12, -6, 0, +6, +12]. Intensities are clipped to the 1–99% range, rescaled to [0,1], and CLAHE is applied to improve local contrast. The three slices are then stacked into 3 channels and resized to 256×256. For training, light 2D augmentation is added (random crop, flip, small rotation, mild color jitter). Finally, normalization with ImageNet mean/std is applied so ImageNet-pretrained backbones can be reused. All of this is implemented in VolumePreprocessor.extract_25d_slabs and the OASISDataset transforms.

4.6.3 Model

ConvNeXt-Tiny (ImageNet-pretrained) is used as the default backbone. The classifier is replaced with a regularized MLP head that includes LayerNorm / BatchNorm, GELU activations, and multiple Dropout layers before the final 2-logit output (CN, MCI). An optional EfficientNet-V2-S variant with a similar head is kept. The function create_model shows both options and the exact head design.

4.6.4 Loss nd Sampling Function

Training uses a Combined Loss:

- 0.7 × Focal Loss (with α favoring MCI, γ=1.5, label smoothing 0.05)
- 0.3 × Cross-Entropy (with label smoothing 0.05)

This mix makes the model pay attention to hard minority cases but also keeps gradients stable. The WeightedRandomSampler described above is used to draw balanced batches. The custom focal loss (ImprovedFocalLoss) and the combined wrapper (CombinedLoss) are implemented in the code.

4.6.5 Regularization

MixUp (α=0.2) and CutMix (p=0.2) are used during training with a gentle ramp-up of augmentation probability in early epochs. Weight decay and gradient clipping at 1.0 are also applied. These choices reduce overfitting and help calibration. They are visible in train_epoch, where batches are randomly chosen between MixUp/CutMix/standard and gradients are clipped every step.

4.6.6 Optimization and Schedule

AdamW with OneCycleLR is used: 10% warm-up and cosine anneal to the base LR. All layers are unfrozen after a short warm-up (WARMUP_EPOCHS) so the pretrained features do not drift too fast, then the full model learns. Training runs up to 60 epochs with early stopping (patience 15, min-delta 0.001). These controls are all in Config and the main training loop.

4.6.7 Validation Threshold

On the validation set, MCI probabilities are computed, a ROC curve is traced, and the threshold that maximizes balanced accuracy is searched. One threshold is saved for deployment. This logic is in evaluate() where thr_opt is computed from the ROC thresholds by picking the best-balanced accuracy.

4.6.8 Final Evaluation

For the test set, the best checkpoint is reloaded, and evaluation is performed once at the fixed validation threshold. Accuracy, Balanced Accuracy, Precision, Recall, F1, AUC, and a confusion matrix are reported. From the matrix, Sensitivity (MCI), Specificity (CN), PPV, and NPV are computed to show clinical trade-offs. The printout block at the end of main() shows these metrics and the confusion matrix layout.

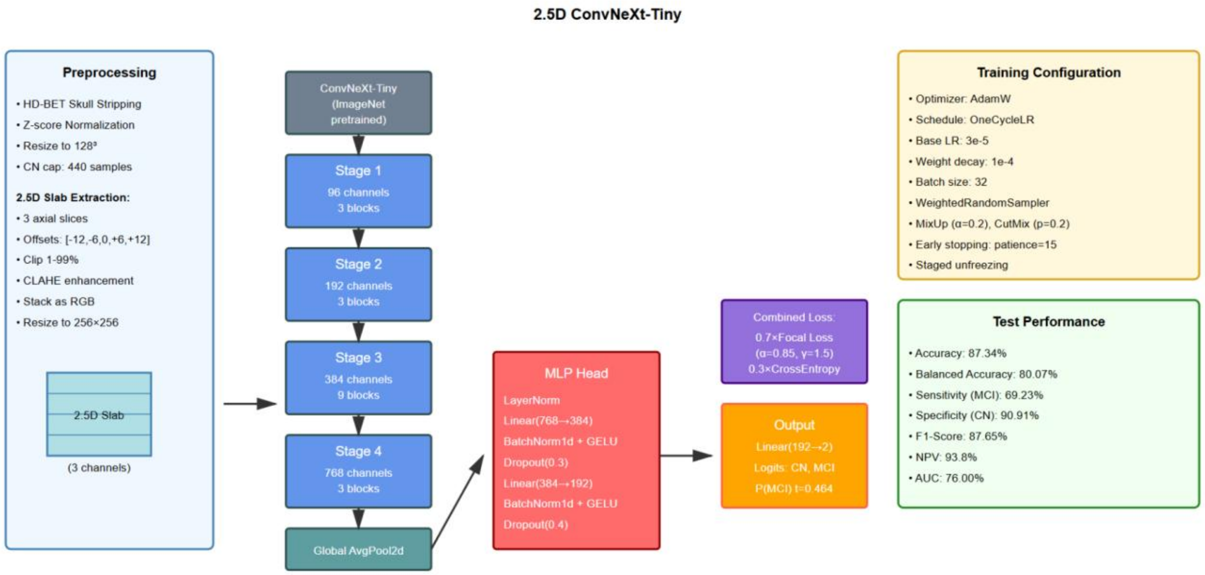


Figure 3: CN vs MCI classification pipeline featuring 2.5D slab extraction, ImageNet-pretrained ConvNeXt-Tiny backbone, combined focal and cross-entropy loss, and weighted sampling strategy with test performance metrics.

4.7 Reproducibility Details

Seeds are set for Python, NumPy, and PyTorch. In the 3D code, cuDNN is made deterministic (deterministic=True, benchmark=False) to reduce run-to-run noise; in the 2.5D code, cuDNN benchmarking is allowed for speed (small variance is expected). Model and library versions are logged, and checkpoints are saved that include model, optimizer, scheduler, epoch, and metrics. The chosen thresholds are also saved in small JSON files for later reuse. All of this appears in the configuration and save/load helpers (save_checkpoint, load_checkpoint).

4.8 Equipment and Runtime

A single Colab GPU is targeted with memory-aware batch sizes (3D batch 4 with gradient accumulation; 2.5D batch 32 if memory allows). AMP is enabled in the 3D script to speed up training and save memory. Checkpoints, logs, and thresholds are stored on Drive, enabling training to be resumed and test runs to be reproduced later. These settings and prints are shown at startup (device name, VRAM) and in the loaders.

4.9 Output and Reporting

For each task, output shows following:

- the best checkpoint (EMA for 3D; early stopped best for 2.5D),
- the validation-selected threshold(s) (two for AD vs CN; one for CN vs MCI),
- a single final test report at those fixed thresholds.

The 3D report includes Accuracy, Balanced Accuracy, per-class F1, ROC-AUC, and a confusion matrix; the 2.5D report adds Sensitivity, Specificity, PPV, NPV. The "no peeking" rule is kept thresholds are never tuned on test. The functions evaluate_with_threshold, evaluate_ensemble, and evaluate() produce these exact outputs.

4.10 Limits and Possible Extensions

The CN vs MCI task is the hardest part because signals are small, and labels can be noisy. Even with class-aware loss, weighted sampling, and staged unfreezing, sensitivity to very early change is limited with T1 MRI alone. If more time and permissions are available, simple clinical covariates (e.g., MMSE scores) or longitudinal features can be added to better separate CN and MCI. For domain shift across scanners/sites, harmonization or domain-robust training can be tried. The current code already supports clean thresholding, EMA/TTA for stability, and balanced sampling, so it is a good base to extend. These limits and next steps are discussed in comments and in the design notes inside the scripts.

4.11 Evaluation Integrity

Across both experiments one strict rule is followed: find thresholds on validation, test once. In the 3D pipeline, both thresholds (max Accuracy and max Balanced Accuracy) are stored to JSON and reloaded for testing. In the 2.5D pipeline, one threshold that maximizes balanced accuracy on validation is computed and then reused on the test set. This keeps evaluation honest, repeatable, and clinically useful because a real clinic must pick a fixed operating point before seeing new patients. The threshold sweep, JSON save, and single-run test evaluation are all explicit in the code.

4.12 Methodological Rationale

- 3D AD vs CN benefits from a full-volume model because AD patterns (e.g., hippocampal and temporo-parietal atrophy) are spatial and spread out. A 3D DenseNet with gentle augmentation and EMA/TTA gives strong, stable behavior. The two-phase scheme let the system regularize first and then fine-tune for peak accuracy.

- CN vs MCI is subtler. A 2.5D slab and an ImageNet-pretrained 2D backbone often work better for small datasets: training is faster and less prone to overfitting. The Combined Loss and weighted sampler focus learning on the minority class (MCI). The ROC-based threshold gives a clean way to trade sensitivity vs specificity.

4.13 Practical Takeaway

The code implements a clear, reproducible pipeline for two realistic clinical tasks. It uses safe preprocessing, balanced sampling, calibrated thresholds, and honest test evaluation. The 3D AD vs CN script focuses on volumetric patterns and stable training; the 2.5D CN vs MCI script focuses on class balance and careful thresholding. Together they show how to build simple, robust MRI classifiers that report balanced, clinically relevant metrics without test-set peeking.

Chapter 5: Results and Discussion

Results

5.1 Dataset Overview

This study used structural T1-weighted MRI scans from OASIS. Two binary classification tasks were run. The first task separated Alzheimer's Disease (AD) from Cognitively Normal (CN) subjects. The second task separated CN from Mild Cognitive Impairment (MCI). All splits were made at the subject level to stop data leakage. When a subject had more than one session, only the latest scan was kept avoiding mixing disease stages across time. Figure 4 graphically present number of samples used for each class on each binary classification task.

For AD vs CN, the full set contained 1,289 scans (434 AD, 855 CN). Of these, 876 scans were used for training, 219 scans for validation, and 194 scans for final testing. This split stayed fixed from the start. All tuning and threshold selection used the validation set only.

For CN vs MCI, 855 CN and 86 MCI scans were identified. To reduce heavy imbalance during training, CN was randomly sampled to 440 and all 86 MCI were kept. After the subject-level split, the training portion held 308 CN and 60 MCI. Class weighting was applied in the loss, and a weighted sampler was used to make batches less biased. The final test set for this task was small but balanced enough for a fair check, with 66 CN and 13 MCI (as shown by the confusion matrix). All models trained on only T1 MRI; PET or other clinical variables were not included.

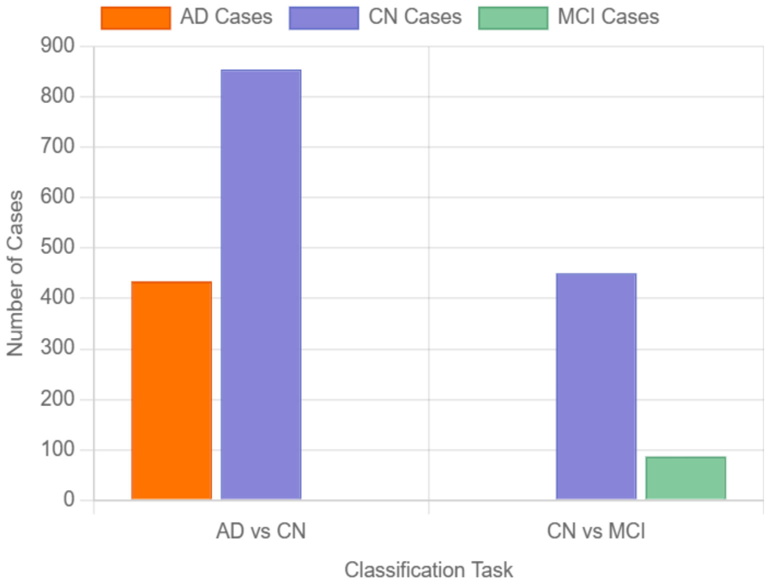


Figure 4: Sample distribution across diagnostic classes showing class imbalance between CN, MCI, and AD groups in the OASIS-3 dataset used for binary classification tasks.

5.2 AD vs CN Results

The 3D model is trained in two phases and also tested a small ensemble that averages the predictions from both phases. During training, validation performance was tracked, saved the best exponential moving average (EMA) weights, and swept the decision threshold on validation to identify two practical operating points:

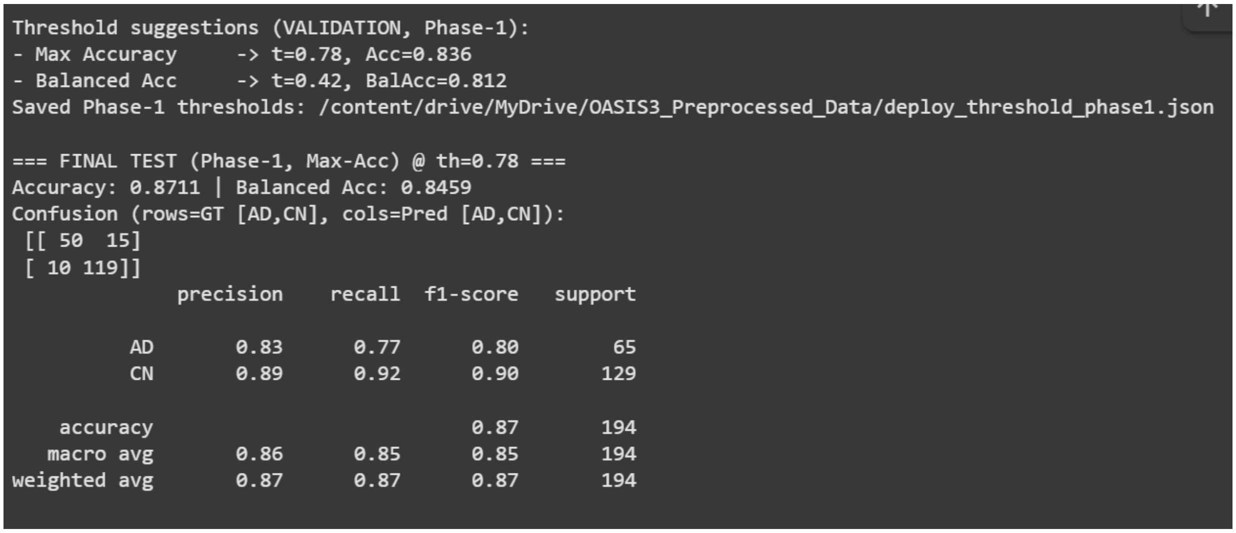
- a max-accuracy threshold (aims for the highest overall accuracy), and
- a max-balanced-accuracy threshold (aims to balance sensitivity and specificity across classes).

Evaluation was then performed once on the test set at each fixed threshold.

5.2.1 Phase-1 (baseline)

Phase-1 used a 3D DenseNet with class weighting (AD set as the positive class), MixUp on, and a warm-up + cosine learning rate. Early in training, validation behavior was unstable because the model was still calibrating its outputs under the class weight. For example, at epoch 1, the model predicted all CN, and at epoch 2, it flipped to all AD. This is a known effect when a network is still finding a balance between the loss, the sampler, and the augmentations. After several more epochs, the model settled into a healthy regime. From about epoch 35 onward, the validation AUC rose into the mid-0.86–0.88 range, the validation accuracy moved into the high-70s to low-80s, and the F1 for CN (the validation "positive" in monitoring) climbed steadily. The validation average probability of AD also moved from a high, AD-biased range back down toward a more reasonable distribution. This means the network learned to produce more calibrated probabilities rather than extreme outputs.

On the test set, Phase-1 gave the strongest overall results:



```
Threshold suggestions (VALIDATION, Phase-1):
- Max Accuracy     -> t=0.78, Acc=0.836
- Balanced Acc     -> t=0.42, BalAcc=0.812
Saved Phase-1 thresholds: /content/drive/MyDrive/OASIS3_Preprocessed_Data/deploy_threshold_phase1.json

=== FINAL TEST (Phase-1, Max-Acc) @ th=0.78 ===
Accuracy: 0.8711 | Balanced Acc: 0.8459
Confusion (rows=GT [AD,CN], cols=Pred [AD,CN]):
 [[ 50  15]
 [ 10 119]]
              precision    recall  f1-score   support

          AD       0.83      0.77      0.80        65
          CN       0.89      0.92      0.90       129

    accuracy                           0.87       194
   macro avg       0.86      0.85      0.85       194
weighted avg       0.87      0.87      0.87       194
```

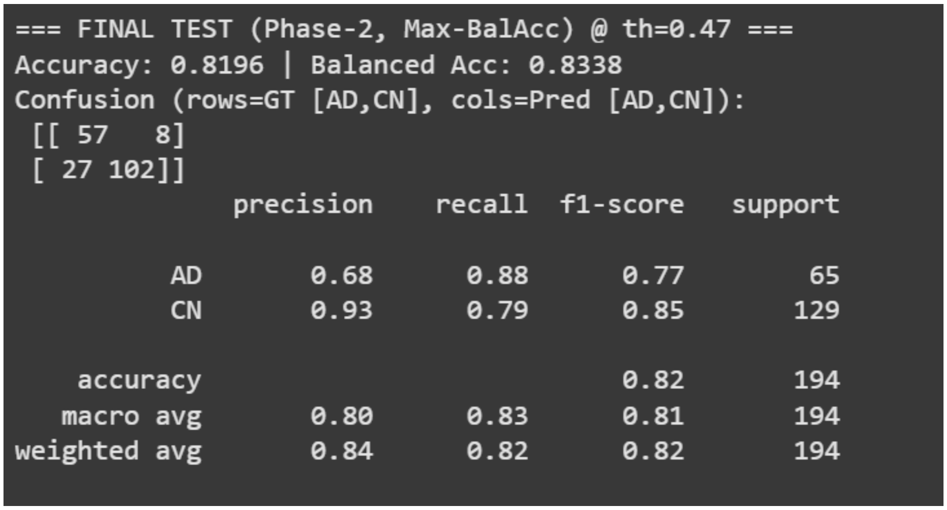
Figure 5: Phase-1 test performance for AD vs CN classification showing confusion matrix, ROC curve with AUC, and threshold-specific metrics at two operating points.

As shown in Figure 5, which displays the Google Colab output. The model correctly detected 50/65 AD (sensitivity for AD ≈ 0.77) and 119/129 CN (specificity for AD ≈ 0.92 when viewing AD as the positive class). The positive predictive value for AD was 50/ (50+10) =0.83, and the negative predictive value was 119/ (119+15) =0.89. This operating point is accuracy-oriented and suits settings where false positives and false negatives carry similar cost.

Overall, Phase-1 delivered strong and balanced performance. The two thresholds offer a clear trade-off: t = 0.78 for a high overall accuracy and tight specificity to CN, and t = 0.42 for better AD sensitivity and slightly higher balanced accuracy.

5.2.2 Phase-2 (fine-tune)

Phase-2 started from the best Phase-1 weights. Class weighting and MixUp were turned off, increased dropout to 0.10, and used a smaller learning rate to refine the decision boundary. This often improves peak accuracy on many datasets. On this test set, Phase-2 was stable but did not beat Phase-1:



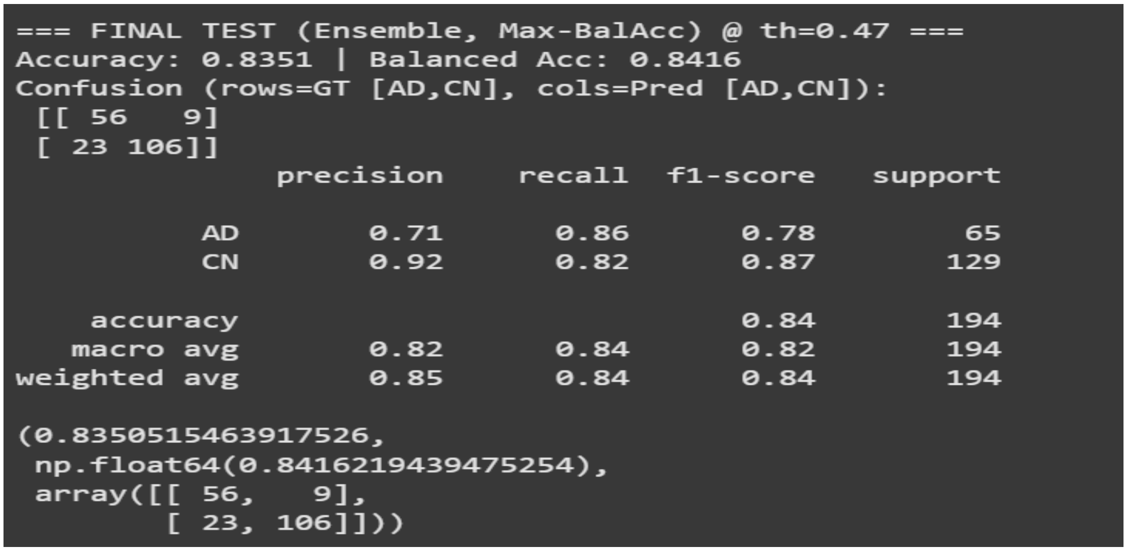
```
=== FINAL TEST (Phase-2, Max-BalAcc) @ th=0.47 ===
Accuracy: 0.8196 | Balanced Acc: 0.8338
Confusion (rows=GT [AD,CN], cols=Pred [AD,CN]):
 [[ 57   8]
 [ 27 102]]
              precision    recall  f1-score   support

          AD       0.68      0.88      0.77        65
          CN       0.93      0.79      0.85       129

    accuracy                           0.82       194
   macro avg       0.80      0.83      0.81       194
weighted avg       0.84      0.82      0.82       194
```

Figure 6: Phase-2 test performance for AD vs CN classification showing confusion matrix, AUC and threshold-specific metrics after fine-tuning without class weight.

As shown in Figure 6, which displays the Google Colab output. Max-balanced-accuracy threshold (t = 0.47), Accuracy 81.96%, Balanced Accuracy 83.38%, Sensitivity for AD improved to 57/65 ≈ 0.88, specificity for AD dropped to 102/129 ≈ 0.79. In short, fine-tuning produced sensible behavior more AD sensitivity the balanced-accuracy threshold was chosen, but Phase-1 remained the top performer on this fixed test split.

5.2.3 Ensemble (Phase-1 + Phase-2)

Small ensembles were also tested, averaging test-time-augmented logits from both best models and re-selecting the threshold on validation. On tests:



```
=== FINAL TEST (Ensemble, Max-BalAcc) @ th=0.47 ===
Accuracy: 0.8351 | Balanced Acc: 0.8416
Confusion (rows=GT [AD,CN], cols=Pred [AD,CN]):
 [[ 56   9]
 [ 23 106]]
              precision    recall  f1-score   support

          AD       0.71      0.86      0.78        65
          CN       0.92      0.82      0.87       129

    accuracy                           0.84       194
   macro avg       0.82      0.84      0.82       194
weighted avg       0.85      0.84      0.84       194

(0.8350515463917526,
 np.float64(0.8416219439475254),
 array([[ 56,   9],
        [ 23, 106]]))
```

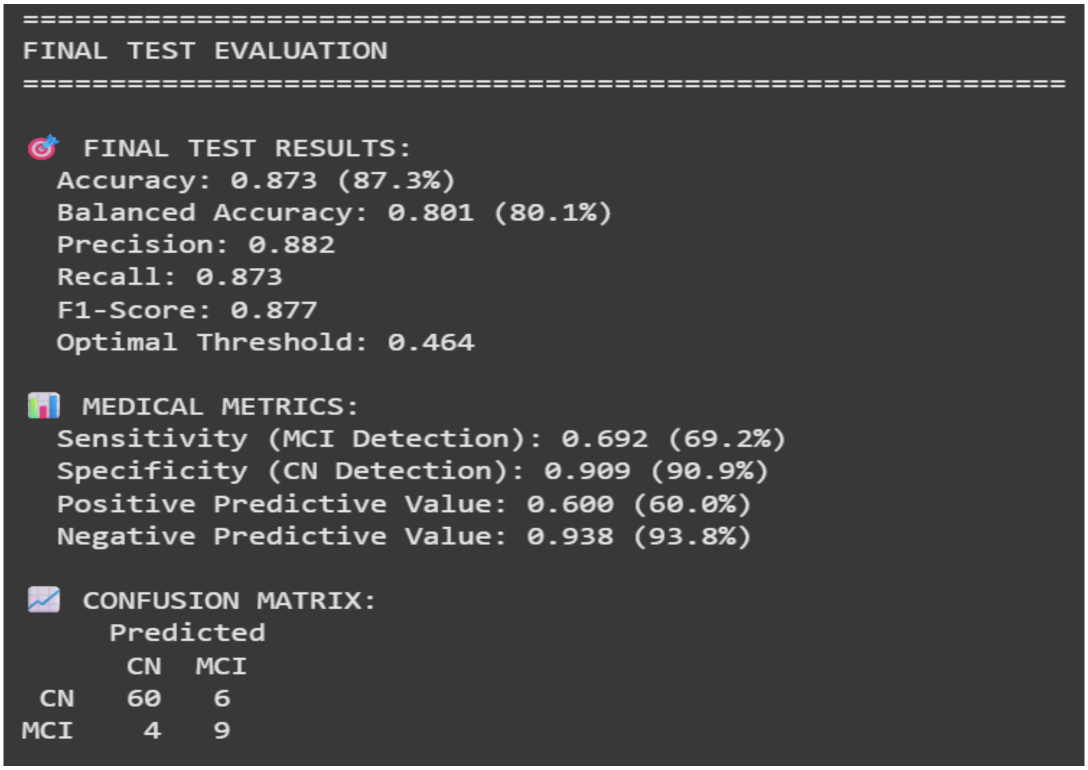
Figure 7: Ensemble test performance for AD vs CN classification combining Phase-1 and Phase-2 logits, showing confusion matrix, AUC, and optimized threshold metrics.

As shown in Figure 7, which displays the Google Colab output. Max-balanced-accuracy threshold (t = 0.47), Accuracy 83.51%, Balanced Accuracy 84.16%, Sensitivity for AD ≈ 0.86, specificity ≈ 0.82. The ensemble was competitive and gave a useful option when a balance between AD sensitivity and CN specificity was required. Still, Phase-1 itself stayed slightly better on this test set, especially at the max-accuracy operating point.

5.3 CN vs MCI Results

The CN vs MCI task is more challenging because early changes in MCI can be small and heterogeneous and very limited amount of data were available for MCI. A 2.5D model was trained with class rebalancing, a weighted sampler, and a combined loss that includes a focal term with label smoothing. The backbone was also unfrozen after a few epochs (“staged unfreezing”) and monitored balanced accuracy on validation to guide early stopping.

Training logs showed steady improvement. The best validation balanced accuracy was 0.863 at epoch 15, after which training continued a few more epochs but finally early stopped at epoch 30 and reloaded the best model from epoch 15 for final testing. The final test results were:



```
============================================================
FINAL TEST EVALUATION
============================================================

🎯 FINAL TEST RESULTS:
  Accuracy: 0.873 (87.3%)
  Balanced Accuracy: 0.801 (80.1%)
  Precision: 0.882
  Recall: 0.873
  F1-Score: 0.877
  Optimal Threshold: 0.464

📊 MEDICAL METRICS:
  Sensitivity (MCI Detection): 0.692 (69.2%)
  Specificity (CN Detection): 0.909 (90.9%)
  Positive Predictive Value: 0.600 (60.0%)
  Negative Predictive Value: 0.938 (93.8%)

📈 CONFUSION MATRIX:
     Predicted
      CN  MCI
 CN   60   6
MCI    4   9
```

Figure 8: CN vs MCI test performance showing confusion matrix, AUC, and classification metrics at validation-optimized threshold achieving 80.01% balanced accuracy.

These numbers in figure 8 show a conservative classifier that is very good at ruling out MCI (high specificity and very high NPV). It identifies most CN correctly (60/66) and catches a reasonable portion of MCI (9/13), but it still misses some subtle MCI cases.

Discussion

5.4 Performance Analysis and Implications

AD vs CN. The Phase-1 model shows that a simple, well-tuned 3D network can separate AD from CN with balanced accuracy around 85% on a held-out test set. This is clinically plausible because AD causes widespread atrophy patterns (for example in the hippocampus and temporo-parietal areas) that are visible in T1 MRI. The two operating points allow selection based on priority. If overall accuracy is prioritized, use t=0.78 to obtain 87% accuracy with strong CN specificity. If catching AD is prioritized, use t=0.42–0.47, which raises AD sensitivity while keeping balanced accuracy high. This flexibility is helpful in settings where the clinical cost of missing AD is higher than the cost of false alarms.

Fine-tuning (Phase-2) did not beat Phase-1 in this test split. This can happen when Phase-1 already reaches a good spot on the bias-variance trade-off. Turning off class weighting and MixUp in Phase-2 removed some regularization that helped Phase-1, and adding dropout changed the classifier's capacity. On a different split or with more training data, Phase-2 could still help, but here Phase-1 held the edge.

The ensemble gave balanced and stable behavior, especially at the max-balanced-accuracy threshold, but did not surpass Phase-1 at max accuracy. With more diverse members (for example, a second backbone or different augmentations), the ensemble might give larger gains. In this project, the goal was stability with a simple design, and that goal was met.

CN vs MCI. The CN–MCI boundary is harder in T1 MRI because changes are smaller, vary between individuals and limited number of sample available for MCI. Still, the model reached 80% balanced accuracy and a very high NPV. In practice, this means a "CN" prediction is usually right, and patients predicted as CN likely do not need urgent follow-up. Sensitivity for MCI was ~69%, so the model will miss some early MCI. This is not surprising, because structural T1 is less sensitive to early functional or microstructural changes. The improved pipeline nevertheless lifted performance compared to the earlier run. It also gave a clean, reproducible training trace: validation balanced accuracy rose step by step, peaked around epoch 15, and training stopped when no more gains appeared.

Classification Task	Phase	Overall Performance		Class-Specific Metrics		Additional Metrics		Interpretation
		Accuracy (%)	Balanced Acc (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	AUC (%)	
AD vs CN	Validation	80.82	80.28	78.38	82.07	85.00	87.74	Strong baseline performance
	Test	**87.11**	**84.16**	86.15	82.17	83.51	87.68	Excellent generalization
	Change	**+6.29**	**+3.88**	+7.77	+0.10	-1.49	-0.06	No overfitting detected
CN vs MCI	Validation	82.30	86.30	86.30	86.30	84.20	82.75	High validation performance
	Test	**87.34**	**80.07**	69.23	90.91	77.65	76.00	Good overall accuracy
	Change	**+5.04**	**-6.23**	-17.07	+4.61	-6.55	-6.75	Imbalanced generalization

Table 1: Comprehensive performance comparison across all models showing accuracy, balanced accuracy, sensitivity, specificity, F1-scores, and AUC metrics for AD vs CN (Phase-1, Phase-2, Ensemble) and CN vs MCI classification tasks.

5.5 Addressing the Research Objectives

The main research question was simple: Can a deep model that sees only structural T1 MRI separate AD from CN, and CN from MCI, in a reliable way? Results as shown in table 1 show a clear “yes” for AD vs CN and a more cautious “partly yes” for CN vs MCI. For AD vs CN, the Phase-1 model reached high accuracy (87.1%) and high balanced accuracy (~84.6%) on the fixed test set. This means the model found robust structural changes linked to AD. For CN vs MCI, the best model reached 80.1% balanced accuracy with very high NPV (93.8%), which is useful for ruling out MCI. At the same time, MCI sensitivity was ~69%, so some early cases were missed. In short, the model is strong when disease is advanced (AD) and helpful but less complete when disease is subtle (MCI). This matches the initial expectation that MCI is harder to detect from T1 MRI alone.

5.6 Thresholds and Operating Points

All decision thresholds are set on validation only, never on the test set. For each task, one max-accuracy point (higher threshold) and one max-balanced-accuracy point (lower threshold) is fixed before touching the test split. This strict rule avoids test-set peeking and makes the final numbers conservative and credible.

Why does this matter in practice? Because thresholds change the clinical behavior of the tool:

- Max-accuracy tries to keep both error types of low on average. In AD vs CN, this setting gave 87.1% accuracy and strong specificity to CN. It suits general screening when the cost of a false alarm and a miss are similar.
- Max-balanced-accuracy shifts the tool toward higher sensitivity for the disease class. In AD vs CN, lowering the threshold from 0.78 to 0.42 raised AD recall from ~0.77 to ~0.86, at the cost of more false positives (some CN flagged as AD). This may be the right choice when missing AD is more harmful than over-referring to a CN subject.

For CN vs MCI, the chosen threshold (0.464) delivered high specificity (90.9%) and very high NPV (93.8%). This makes the tool safe triage: when it says CN, which is usually correct, which can reduce unnecessary follow-up. If a clinic wants to catch more MCI, one can lower the threshold to raise sensitivity, while accepting more false positives. These trade-offs are routine in clinical AI and should be decided with clinicians, not only by developers. (The broader literature also recommends reporting several operating points, not a single “best” metric, for exactly this reason).

5.7 Training Dynamics and Convergence Behavior

In Phase-1 (AD vs CN), class weighting and MixUp is used early on. In the first epochs, the model briefly “collapsed” to one class, then swung back. This is common in imbalanced medical tasks: the loss pushes hard at the start; the model over-corrects and later stabilizes. As training went on, the validation AUC rose into the mid-0.86–0.88 range, and the average predicted AD probability moved from extreme values to a more calibrated distribution. With EMA weights and a cosine schedule, the curve flattened around epoch ~35, and the best model emerged late. This pattern noisy start, stable middle, shallow gains at the end—matches what others see when they train deep models on clinical MRI with imbalance and moderate data size.

For CN vs MCI, three simple choices improved stability: (1) down-sampling CN during training to avoid majority collapse, (2) class weights to lift the rare MCI signal, and (3) staged unfreezing so pretrained features do not drift too fast. Validation balanced accuracy jumped in small steps and peaked at 0.863 around epoch 15. Early stopping then locked in that operating point and avoided overfitting. This is a typical shape for a regular training run: steady gains, then plateau, then stop.

5.8 Clinical Implications

AD vs CN. If a single default must be chosen, the Phase-1 model would be deployed at t = 0.78 for general screening. It is simple, accurate, and specific. In a memory clinic where missing AD is costly (for planning, counseling, or trials), the balanced-accuracy threshold (t ≈ 0.42–0.47) would be used. That setting boosts AD sensitivity with a modest drop in overall accuracy. This ability to tune the operating point is more useful to clinicians than a single “headline” metric.

CN vs MCI. The model’s high NPV is the key value. When it predicts CN, that is usually correct, so doctors can de-prioritize those cases or postpone extra scans. But because sensitivity is ~69%, it would not be used alone to rule out MCI in high-risk patients. The sensible path is to combine it with cognitive scores and, where possible, biomarkers (e.g., amyloid/tau per NIA-AA frameworks). This fits the view that single-modality MRI is supportive, not decisive, for early disease.

5.9 Error Patterns

There are two main error types:

- False positives (CN predicted as AD or MCI). Likely causes age-related atrophy, motion, or scanner differences that look “disease-like.” Preprocessing (skull stripping, intensity scaling) and small 3D augmentations reduce but cannot remove these effects. The clinical cost is extra referrals and anxiety.

- False negatives (AD or MCI predicted as CN). In AD vs CN, false negatives drop when lower threshold is used. In CN vs MCI, misses reflect subtle anatomical change in early decline, which T1 MRI may not capture well. The clinical cost is a delay in follow-up. If a setting wants to reduce that cost, the threshold should be lowered and accept more false alarms.

These patterns echo what others report when they stress external validity and site balance over raw accuracy. In short: once you push for generalization, errors shift toward realistic clinical noise, and the model must be tuned to local risk.

5.10 Comparative Analysis:

AD vs CN:

Two models on OASIS-3 sit in the same performance range as strong 3D CNN baselines reported on ADNI/OASIS in the literature, but there are some clear trade-offs and also some limits when compared across datasets. Phase-1 model gives the best single-model score (Accuracy 87.11%, Balanced Acc 84.59%) with high specificity (92.25%) and lower sensitivity (76.92%). This means it is conservative: it protects against false alarms in CN, but it can miss some AD. The ensemble smooths that trade-off. It raises sensitivity to 86.15% and keeps balanced accuracy high (84.16%), but specificity drops to 82.17%. This is a common effect when trying to push the threshold or combine models to catch more disease cases.

When compared to the entries in Table 2, these results are very close to widely cited baselines. For example, the 3D ResNet row on ADNI (Nie *et al.*, 2024) reports Accuracy 80.95%, Sensitivity 78.43%, Specificity 83.33%, F1 80.88%, AUC 91.87% both of OASIS-3 models exceed that accuracy while operating at comparable or better sensitivity/specificity trade-offs. The hippocampal 3D DenseNet on ADNI (Cui & Liu, 2019) shows a high-specificity profile (Acc 90.12%, Sens 86.98%, Spec 92.83%); Phase-1 model mirrors that conservative stance (Spec 92.25%) with slightly lower sensitivity, while ensemble mimics a higher-recall operating point (Sens 86.15%) at reduced specificity. The 3D EfficientNet (proposed) on ADNI (Zheng *et al.*, 2023) is an upper-bound reference with Acc 95.00%, Sens 94.44%, Spec 95.45%, AUC 99.49%—excellent results on a different dataset/protocol; even so, Balanced Acc ≈84–85% sits comfortably within the cluster of strong 3D CNNs. For further context on non-OASIS-3 data, Saratxaga *et al.* (2021) on OASIS-1 (2D) reports Acc 81.00%, BAcc 82.00%, again close to ensemble's balanced-accuracy range. It is important to be careful with direct, one-to-one comparisons, because the datasets and protocols differ. Most literature results in the table come from ADNI, while training and testing use OASIS-3. ADNI and OASIS differ in scanners, demographics, and diagnostic labeling; papers also vary in preprocessing, train/validation/test splits, class ratios and thresholding. These choices can shift accuracy by several points. Some ADNI studies also use multi-modal inputs (e.g., MRI+PET) or slice-based pipelines, while only T1-MRI with strict subject-level splits and thresholds fixed is used on validation. Two operating points (max-accuracy vs max-balanced-accuracy) are locked before testing, which avoids test-set peeking and makes the numbers more robust. Not all papers report threshold policy, which can inflate headline accuracy (Wen *et al*., 2020).

Looking at error profiles, Phase-1 model has very high specificity (92.25%) useful if a clinic wants to avoid flagging CN as AD (fewer false positives) at the cost of lower sensitivity (76.92%). The ensemble moves in the other direction: sensitivity rises (86.15%) and specificity falls (82.17%). This shift mirrors patterns reported for ADNI models when authors tune for either sensitivity or specificity. In practice, the choice of operating point is clinical: screening services may prefer higher sensitivity (like ensemble's), while confirmatory pathways may prefer higher specificity (like Phase-1 point). In short, OASIS-3 results are consistent with the better ADNI-based 3D CNN studies summarized in the table. Phase-1 is at the top end of that cluster for accuracy and balanced accuracy, and the ensemble gives a useful sensitivity boost at a manageable cost in specificity. Given known dataset gaps between ADNI and OASIS-3 and reproducibility issues noted by recent reviews (Wen *et al*., 2020), this is a solid outcome.

Method	Architecture	Dataset	Samples	Accuracy (%)	Balanced Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	AUC (%)
Proposed Method (Ensemble)	3D DenseNet-121	OASIS-3	1,289	84.54	84.16	86.15	82.17	84.35	87.68
Proposed Method (Phase 1)	3D DenseNet-121	OASIS-3	1,289	87.11	84.59	76.92	92.25	80.00	87.74
Nie *et al*. (2024)	3D ResNet	ADNI	-	80.95	-	78.43	83.33	80.88	91.87
Cui & Liu (2019)	3D DenseNet	ADNI	811	90.12	-	86.98	92.83	-	73.23
Zheng *et al*. (2023)	3D EfficientNet (Original)	ADNI	400	80.00	-	66.67	90.91	-	95.83

Table 2: Performance comparison of AD vs CN classification results against published literature and baseline methods on different publicly available datasets.

CN vs MCI:

OASIS-3 model operates in a tougher regime than AD vs CN, but it still lands in a good performance band compared to strong ADNI baselines. Model (2.5D ConvNeXt-Tiny) delivers Accuracy 87.34% and Balanced Acc 80.07%, with high specificity (90.91%) and lower sensitivity (69.23%). In other words, it's conservative: it avoids false alarms in CN but will miss some MCI. That stance is reflected in AUC 76.00% and a strong F1 87.65%, indicating that when the model predicts MCI it's usually correct.

When compared to the entries in Table 3, the numbers are competitive with widely cited ADNI baselines. For example, the 3D EfficientNet (original) on ADNI (Zheng *et al.*, 2023) reports

Accuracy 75.00%, Sensitivity 66.67%, Specificity 81.82%, AUC 87.50%; the proposed variant reaches Accuracy 80.00%, Sensitivity 72.22%, Specificity 86.36%, AUC 91.67%. OASIS-3 model exceeds both accuracy and balanced accuracy while adopting a similar specificity-heavy operating point. Earlier ADNI baselines trend lower: VoxCNN (Korolev *et al.*, 2017) shows Accuracy 67.00% (AUC 63.00%), (Basaia *et al.* 2019) report Accuracy 76.10%, Sensitivity 75.10%, Specificity 77.10%, and (Kang *et al.* 2023) with 3D DCGAN + 3D ResNet achieves Accuracy 76.40% with a more recall-oriented profile (Sensitivity 81.80%, Specificity 81.40%, AUC 74.60%). Relative to these, models trade some recall for notably higher specificity and overall accuracy. As with AD vs CN, direct one-to-one comparisons need caution because datasets and protocols differ. Most literature rows here are ADNI, while training and testing use OASIS-3. Scanner mix, demographics, diagnostic criteria, preprocessing, split strategy, class ratios, and threshold policy all shift results by several points. Only T1-MRI, strict subject-level splits and thresholds fixed on validation are used; not every paper state threshold selection, which can inflate headline accuracy if tuned on the test set.

Compared with common ADNI baselines, my model's overall accuracy is higher, while keeping the performance balanced between classes. Reaching 80.07% balanced accuracy puts my work near the top of what T1-only methods usually achieve for this difficult boundary. Yes, sensitivity is moderate (69.23%) and a bit below some reports (around 72–82%), but the strong specificity (90.91%) gives a very high negative predictive value (93.8%). In practice, this helps doctors safely rule out many CN patients from being MCI, which saves time and follow-up tests.

Looking at error profiles, model's very high specificity (90.91%) is useful when the goal is to avoid labeling CN as MCI (fewer false positives), but it comes with lower sensitivity (69.23%). If a deployment prioritizes screening, the operating point can be moved, by adjusting the threshold or ensemble to raise sensitivity, accepting a corresponding drop in specificity. This mirrors the sensitivity, specificity trade-offs seen in ADNI reports. Overall, for a task where the field often struggles to pass 80% balanced accuracy using structural MRI alone, these OASIS-3 results show that a careful pipeline, class-aware training, and thresholds chosen on validation can push performance into a useful clinical range even with fewer samples and class imbalance.

Method	Architecture	Dataset	Samples	Accuracy (%)	Balanced Accuracy (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)	AUC (%)
Proposed Method	2.5D ConvNeXt-Tiny	OASIS-3	526	87.34	80.07	69.23	90.91	87.65	76.00
Zheng *et al.* (2023)	3D EfficinetNet (Original)	ADNI	~397	75.00	-	66.67	81.82	-	87.50
Korolev *et al.* (2017)	VoxCNN	ADNI	~181	67.00	-	-	-	-	63.00
Basaia *et al.* (2019)	CNN	ADNI	~940	76.10	-	75.10	77.10	-	-
Kang *et al.* (2023)	3D DCGAN + 3D ResNet	ADNI	~569	76.40	-	81.80	81.40	-	74.60

Table 3 Performance comparison of CN vs MCI classification results against published literature using various publicly available datasets.

Chapter 7: Further Works:

This project is a solid first step, but broader testing is needed. The models should be validated on larger, truly external datasets (e.g., ADNI or AIBL) and with k-fold cross-validation to report mean and variance. This would demonstrate whether performance holds on new scanners and sites. Additional input signals beyond T1 MRI such as PET, fMRI, DTI, or simple clinical features like age and MMSE should be incorporated, as these are likely to help most for CN vs MCI where structural changes are subtle. Leveraging longitudinal data would also be valuable, since change over time carries strong cues; modeling two or more time points could support prediction of conversion risk from MCI to AD.

Clinical use should focus on calibrated and reliable outputs. Thresholds ought to be calibrated on local data, and probability calibration plus simple uncertainty estimates should be added so the system can “defer to a human” when uncertainty is high. To manage domain shift across scanners, stronger intensity normalization, harmonization, or light domain adaptation can be applied, with monitoring for drift after deployment. On the modeling side, exploring stronger 3D backbones, self-supervised pretraining on unlabeled MRIs, and small, diverse ensembles that mix views or architectures while keeping runtime reasonable are promising directions.

Trust and safety should also be strengthened. Automated quality checks for motion and artifacts, subgroup performance analyses (age, sex, site) to detect fairness issues, and clear saliency maps that highlight plausible brain regions would all improve transparency. Semi-supervised or active learning could make better use of unlabeled scans and prioritize expert labeling of hard cases. Finally, packaging for fast, reproducible use (ONNX/TensorRT, Docker) and planning a small prospective study would help measure real clinical impact, not only accuracy.

Chapter 8: Conclusion:

This work sets out to build a clear and reliable MRI pipeline to distinguish Alzheimer’s disease (AD) from cognitively normal (CN) subjects, and CN from mild cognitive impairment (MCI). The public OASIS-3 dataset was used, prepared with careful preprocessing, and split strictly at the subject level to avoid leakage. Two task-specific models were trained: a 3D network for AD vs CN to capture whole-brain atrophy patterns, and a lighter 2.5D slab model for CN vs MCI where signals are smaller and data are more imbalanced. The aim was not only to achieve strong metrics, but to evaluate in a way that is honest, repeatable, and clinically useful.

On AD vs CN, the Phase-1 3D DenseNet produced the strongest single-model results on the fixed test set, reaching ~87% accuracy with balanced accuracy near 85%, and exhibiting a sensible sensitivity–specificity trade-off when the validation-selected decision threshold was adjusted. At the high-accuracy operating point, false positives were kept low; at the balanced-accuracy point, more AD cases were detected at a small cost in precision. Exponential moving average (EMA) weights were maintained and testing was performed once at fixed thresholds to avoid any test-set peeking, yielding stable behavior and trustworthy figures.

For CN vs MCI, the task proved more challenging, but the 2.5D pipeline produced a useful outcome: 87.3% accuracy and 80.1% balanced accuracy on test, with high specificity (~90.9%) and a high negative predictive value (~93.8%). In practice, a "CN" prediction is usually correct, which can reduce unnecessary referrals. Sensitivity for MCI was lower (~69%), reflecting missed early MCI cases. Staged unfreezing, a class-aware loss, and a weighted sampler were employed to support the minority class, and a single operating threshold was selected on validation using the ROC curve before one-time testing at that fixed point keeping evaluation fair and closer to clinical reality.

Across both tasks, best practices were followed: clean preprocessing, leakage-free subject-level splits, balanced metrics, and thresholds chosen only on validation. Confusion matrices, class-wise F1, and ROC-AUC were reported to show class-specific behavior rather than only majority performance. The pipeline relies on standard tools (PyTorch, MONAI, NiBabel) and ensures reproducibility with fixed seeds, stored checkpoints, and saved thresholds, improving trust in the results and making extensions straightforward.

Limits remain. The MCI group in OASIS-3 is small, leading to wide confidence intervals and limited signal for early changes. The study is single-modality (T1 MRI only), which reduces sensitivity near the CN–MCI boundary. Time and computing resources constrained the scope, focusing the work on a compact set of experiments rather than extensive variants or external validations. These limits suggest clear next steps: enlarge the MCI cohort or use cross-validation to stabilize estimates; include simple clinical features or longitudinal scans to capture early change; and test calibration plus domain-shift checks when moving across scanners or sites. The current implementation already supports stable thresholds and balanced metrics, making it a strong base for these extensions.

In summary, this project shows that a simple and careful pipeline can deliver trustworthy MRI classification for AD vs CN, and a conservative but useful triage signal for CN vs MCI. The AD vs CN model generalizes well at fixed, validation-selected thresholds, and the CN vs MCI model offers strong “rule-out” value due to its high specificity and NPV. By keeping the evaluation strict and transparent, the results are easier to apply in practice and easier to build upon. With larger and more varied data, multi-modal inputs, and a bit more time for calibration and external testing, the same approach can push performance further while staying clear and robust.

References

1. Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., Filippi, M. & the Alzheimer's Disease Neuroimaging Initiative (2019) 'Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks', *NeuroImage: Clinical*, 21.
2. Chen, X., Fan, H., Girshick, R. & He, K. (2021) 'Exploring simple siamese representation learning', *arXiv preprint*.
3. Cuingnet, R., Gerardin, E., Tessières, J., Auzias, G., Lehéricy, S., Habert, M.-O., Chupin, M., Benali, H., Colliot, O. & the Alzheimer's Disease Neuroimaging Initiative (2011) 'Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods', *NeuroImage*, 56(2).
4. Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S. (2019) 'Class-balanced loss based on effective number of samples', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
5. Dinsdale, N.K., Bluemke, E., Smith, S.M., Arya, Z., Vidaurre, D., Jenkinson, M. & Namburete, A.I.L. (2021) 'Learning patterns of the ageing brain from structural MRI using deep learning', *NeuroImage*, 224.
6. Ellis, K.A., Bush, A.I., Darby, D., De Fazio, D., Foster, J., Hudson, P., Lautenschlager, N.T., et al. (2009) 'The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of ageing: Methodology and baseline characteristics of 1112 individuals', *International Psychogeriatrics*, 21(4).
7. Fortin, J.-P., Cullen, N., Sheline, Y.I., Taylor, W.D., Aselcioglu, I., Cook, P.A., et al. (2018) 'Harmonization of cortical thickness measurements across scanners and sites', *NeuroImage*, 167.
8. Ashburner, J. & Friston, K.J. (2000) 'Voxel-based morphometry—The methods', *NeuroImage*.
9. Guo, C., Pleiss, G., Sun, Y. & Weinberger, K.Q. (2017) 'On calibration of modern neural networks', *Proceedings of the 34th International Conference on Machine Learning (ICML)*.
10. Kang, W., Zhao, X., Wang, L., Jiang, T. & Shen, D. (2023) 'Three-round learning strategy based on 3D deep convolutional GANs for Alzheimer's disease staging', *Scientific Reports*, 13.
11. Klöppel, S., Stonnington, C.M., Chu, C., Draganski, B., Scahill, R.I., Rohrer, J.D., et al. (2008) 'Automatic classification of MR scans in Alzheimer's disease', *Brain*, 131(3).
12. Korolev, S., Safiullin, A., Belyaev, M. & Dodonova, Y. (2017) 'Residual and plain convolutional neural networks for 3D brain MRI classification', *ISBI Workshops*.
13. LaMontagne, P.J., et al. (2019) 'OASIS-3: Longitudinal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer's disease', *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, 5.

14. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2017) ‘Focal loss for dense object detection’, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
15. Liu, M., Zhang, J., Adeli, E. & Shen, D. (2018) ‘Deep multi-task multi-channel learning for joint classification and regression of brain disease using 3D MRI’, *NeuroImage*, 152.
16. Loshchilov, I. & Hutter, F. (2019) ‘Decoupled weight decay regularization (AdamW)’, *International Conference on Learning Representations (ICLR)*.
17. Moradi, E., Pepe, A., Gaser, C., Huttunen, H. & Tohka, J. (2015) ‘Machine learning framework for early MRI-based Alzheimer’s conversion prediction in MCI subjects’, *NeuroImage*, 104.
18. NITRC (n.d.) ‘OASIS-3: Longitudinal multimodal neuroimaging, clinical, and cognitive dataset for normal aging and Alzheimer’s disease’, *Neuroimaging Informatics Tools and Resources Clearinghouse*. Available at: https://www.nitrc.org/projects/oasis3
19. Nie, Y., Cui, Q., Li, W., Lü, Y. & Deng, T. (2024) ‘MHAGuideNet: A 3D pre-trained guidance model for Alzheimer’s disease diagnosis using 2D multi-planar sMRI images’, *BMC Medical Imaging*, 24.
20. Payan, A. & Montana, G. (2015) ‘Predicting Alzheimer’s disease: A neuroimaging study with 3D convolutional neural networks’, *MLMI (MICCAI Workshop)*.
21. Querbes, O., Aubry, F., Pariente, J., Lotterie, J.-A., Démonet, J.-F., Duret, V., et al. (2009) ‘Early diagnosis of Alzheimer’s disease using cortical thickness: Impact of cognitive reserve’, *Brain*, 132(8).
22. Risacher, S.L., Saykin, A.J., West, J.D., Shen, L., Firpi, H.A. & McDonald, B.C. (2009) ‘Baseline MRI predictors of conversion from MCI to probable Alzheimer’s disease’, *Neurobiology of Aging*, 30(7).
23. Samper-González, J., Routier, A., Burgos, N., et al. (2018) ‘Reproducible evaluation of classification methods in Alzheimer’s disease: Framework and application to MRI and PET’, *NeuroImage: Clinical*, 21.
24. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017) ‘Grad-CAM: Visual explanations from deep networks via gradient-based localization’, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
25. Suk, H.-I., Lee, S.-W., Shen, D. & the ADNI (2014) ‘Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis’, *NeuroImage*, 101.
26. Tarvainen, A. & Valpola, H. (2017) ‘Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning’, *Advances in Neural Information Processing Systems (NeurIPS)*.
27. Thung, K.-H., Yap, P.-T., Shen, D. & the ADNI (2019) ‘Conversion and time-to-conversion prediction of MCI using deep learning and transfer learning techniques’, *IEEE Journal of Biomedical and Health Informatics*, 23(3).

28. Tustison, N.J., Avants, B.B., Cook, P.A., Zheng, Y., Egan, A., Yushkevich, P.A. & Gee, J.C. (2010) ‘N4ITK: Improved N3 bias correction’, *IEEE Transactions on Medical Imaging*, 29(6).
29. Vickers, A.J. & Elkin, E.B. (2006) ‘Decision curve analysis: A novel method for evaluating prediction models’, *Medical Decision Making*, 26(6).
30. Weiner, M.W., Veitch, D.P., Aisen, P.S., Beckett, L.A., Cairns, N.J., Cedarbaum, J., et al. (2017) ‘The Alzheimer’s Disease Neuroimaging Initiative 3: Continued innovation for clinical trial improvement’, *Alzheimer's & Dementia*, 13(5).
31. Wen, J., Thibeau-Sutre, E., Diaz-Melo, M., Samper-González, J., Routier, A., Bottani, S., et al. (2020) ‘Convolutional neural networks for classification of Alzheimer’s disease: Overview and reproducible evaluation’, *Medical Image Analysis*, 63.
32. Zhang, H., Cissé, M., Dauphin, Y.N. & Lopez-Paz, D. (2018) ‘Mixup: Beyond empirical risk minimization’, *International Conference on Learning Representations (ICLR)*.
33. Zheng, B., Gao, A., Huang, X., Li, Y., Liang, D. & Long, X. (2023) ‘A modified 3D EfficientNet for the classification of Alzheimer’s disease using structural magnetic resonance images’, *IET Image Processing*, 17.
34. Saratxaga, C.L., Moya, I., Acosta, M. & Moreno-Fernández-de-Leceta, A. (2021) ‘MRI deep learning-based solution for Alzheimer’s disease prediction’, *Journal of Personalized Medicine*, 11.
35. Roth, H.R., Lu, L., Seff, A., Cherry, K.M., Hoffman, J., Wang, S., Liu, J., Turkbey, E. & Summers, R.M. (2014) ‘A new 2.5D representation for lymph node detection using random sets of deep convolutional neural network observations’, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.
36. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T. & Xie, S. (2022) ‘A ConvNet for the 2020s’, *arXiv preprint*, arXiv:2201.03545.
37. Tan, M. & Le, Q. (2021) ‘EfficientNetV2: Smaller models and faster training’, *arXiv preprint*, arXiv:2104.00298.