

A Small and Fast Deep Learning Model That Detects Alzheimer's Disease from Brain MRI Scans Using Limited Data and Low-Cost Devices

¹B Sowjanya

¹Assistant Professor, Department of Computer Science and Engineering, Methodist College of Engineering and Technology

¹Email: Sowjanya.b@Methodist.Edu.In

Abstract

Deep learning models for Alzheimer's disease (AD) detection from magnetic resonance imaging (MRI) usually report high accuracy, but they are trained on thousands of labelled scans, run on GPUs, and are often evaluated with slice-level splits that leak information between training and testing. This paper presents ElasticAD, a lightweight and resource-adaptive framework designed for the realistic case of a few hundred labelled subjects and low-cost hardware. ElasticAD converts each T1-weighted volume into 2.5D inputs made of three adjacent axial slices, learns general brain-tissue features from unlabelled scans through self-supervised contrastive pre-training, and then trains a compact student network built from Ghost modules and coordinate attention under the guidance of a larger ConvNeXt-Tiny teacher. A class-balanced loss handles the shortage of demented subjects. To adapt to different devices, the trained student is compressed by budget-aware channel pruning and fitted with a confidence-based early-exit branch, and slice predictions are fused into one decision per subject. On 198 labelled OASIS-1 subjects classified as cognitively normal, very mild dementia or mild-to-moderate dementia, and evaluated with strict subject-level five-fold cross-validation, ElasticAD reached 85.3% balanced accuracy and a macro F1-score of 0.85 with only 0.84 million parameters and 70 MFLOPs per input. It outperformed ResNet18, DenseNet121, MobileNetV3, ShuffleNetV2 and EfficientNet-B0, and it matched its own 28-million-parameter teacher. With only 25% of the labelled subjects it still achieved 77.8% balanced accuracy, 11.3 points above EfficientNet-B0. On a Raspberry Pi 4 the full model needs 12.1 ms per input and the pruned 30% variant only 4.2 ms, showing that reliable AD screening is possible without large datasets or GPUs.

Keywords: Alzheimer's disease, structural MRI, lightweight deep learning, knowledge distillation, self-supervised learning, Ghost module, network pruning, early exit, edge computing, OASIS

1. Introduction

Alzheimer's disease is a progressive neurodegenerative disorder and the leading cause of dementia in older adults [1]. Its first clinical signs are mild memory complaints that are easy to dismiss, while the underlying neuronal loss may have started years earlier. Detecting the disease at the very mild stage gives patients and families time to plan care, start symptomatic treatment and join clinical trials. In countries such as India, where the elderly population is growing quickly and specialist neurologists are concentrated in large cities, an affordable screening aid could reach

many people who are currently diagnosed late or not at all.

Structural T1-weighted MRI is part of the standard work-up for suspected dementia. It reveals the medial temporal lobe atrophy, hippocampal shrinkage and ventricular enlargement that accompany AD [2]. Clinical staging is commonly expressed with the Clinical Dementia Rating (CDR), where 0 indicates no dementia, 0.5 very mild dementia, and 1 or more mild to moderate dementia [3]. The very mild stage is the hardest to recognise visually because the anatomical changes are small and overlap with normal ageing.

Convolutional neural networks (CNNs) can learn atrophy patterns directly from images and have been widely applied to AD classification [4], [5]. However, three gaps separate published results from practical use. First, many studies split individual 2D slices randomly into training and test sets, so slices of the same person appear on both sides; Wen et al. [6] showed that this data leakage can inflate accuracy by large margins. Second, honest subject-level datasets are small: a typical public cohort provides only a few hundred labelled subjects, too few for networks with tens of millions of parameters. Third, the models are built for GPU servers, whereas a district hospital, a mobile screening van or a college laboratory may only have an ordinary computer or an embedded board.

This paper addresses all three gaps together with ElasticAD, a framework that is both lightweight and able to adapt its computational cost to the device on which it runs. The main contributions are:

- A strict subject-level protocol on OASIS-1 with 2.5D inputs and probability fusion across slices, giving one leakage-free decision per subject.
- A label-efficient training pipeline that combines self-supervised contrastive pre-training on unlabelled scans, knowledge distillation from a large teacher and a class-balanced loss.
- A compact student network of Ghost bottlenecks with coordinate attention that has fewer than one million parameters.
- Two complementary adaptation mechanisms, budget-aware channel pruning before deployment and confidence-based early exit at run time, evaluated on a Raspberry Pi 4 and a laptop CPU.

Section 2 reviews related work. Section 3 describes the dataset and the proposed framework. Section 4 gives the experimental setup, Section 5 discusses the results, and Section 6 concludes the paper.

2. Literature Review

Early machine-learning approaches to AD detection extracted volumetric or morphometric features, such as hippocampal volume or cortical thickness, and classified them with support vector machines. These pipelines depend on segmentation tools and hand-designed features, which limits their robustness across scanners. Deep learning removed the need for manual features. Korolev et al. [7] showed that plain and residual 3D CNNs could classify whole MRI volumes without handcrafted features. Liu et al. [8] proposed a landmark-based multi-instance network that learned from patches around anatomical landmarks, and Basaia et al. [9] trained a 3D CNN on a single T1 scan to separate AD, converting and stable mild cognitive impairment. A systematic review by Ebrahimighahnavieh et al. [5] found that most studies rely on ADNI or OASIS and that 2D slice-based models are popular because they can reuse ImageNet weights.

The reliability of reported accuracies is a known problem. Wen et al. [6] re-implemented several published methods under a reproducible, subject-level framework and reported much lower accuracies than the original papers, mainly because of slice-level leakage and test-set reuse during model selection. Their findings motivate the strict protocol used in this paper.

Efficient architectures reduce the cost of CNNs. MobileNetV3 [10] combines neural architecture search with squeeze-and-excitation blocks, ShuffleNetV2 [11] follows practical guidelines for fast inference, and EfficientNet [12] scales depth, width and resolution together. GhostNet [13] observes that many feature maps in a CNN are near-duplicates and generates them with cheap linear operations instead of full convolutions. Coordinate attention [14] embeds positional information into channel attention at very low cost, which is useful for locating small anatomical structures.

Several training strategies help when labels are scarce. Knowledge distillation [15] trains a small student to mimic the softened outputs of a large

teacher. Self-supervised contrastive learning such as SimCLR [16] learns representations from unlabelled images by pulling together two augmented views of the same image. The class-balanced loss of Cui et al. [17] reweights classes by their effective number of samples. For deployment, network slimming [18] prunes channels with small batch-normalisation scaling factors, and BranchyNet [19] adds early-exit classifiers so that easy inputs leave the network sooner.

Table 1. Summary of related studies

Ref.	Approach	Data / protocol	Limitation
[7]	3D plain and residual CNN	ADNI, subject level	Heavy 3D model; binary task
[8]	Landmark-based multi-instance CNN	ADNI, MIRIAD	Needs landmark detection step
[9]	3D CNN on single T1 scan	ADNI, subject level	Large model; GPU required
[6]	Reproducible CNN evaluation	ADNI, AIBL, OASIS	Shows leakage; no efficiency focus
[13]	GhostNet	ImageNet	Not validated on MRI
[15]	Knowledge distillation	General vision	Not tuned for small medical data

Research gap: existing AD studies tend to optimise one aspect at a time: accuracy with large 3D models, efficiency with generic mobile networks, or reliability through careful evaluation. Few combine label-efficient training, a sub-million-parameter model, device-level adaptation and a leakage-free subject-level protocol. ElasticAD is designed to fill this gap.

3. Materials and Methods

ElasticAD consists of six stages, shown in Figure 1: data selection, preprocessing, 2.5D input construction, self-supervised pre-training, teacher-guided training of the student, and resource adaptation followed by subject-level fusion.

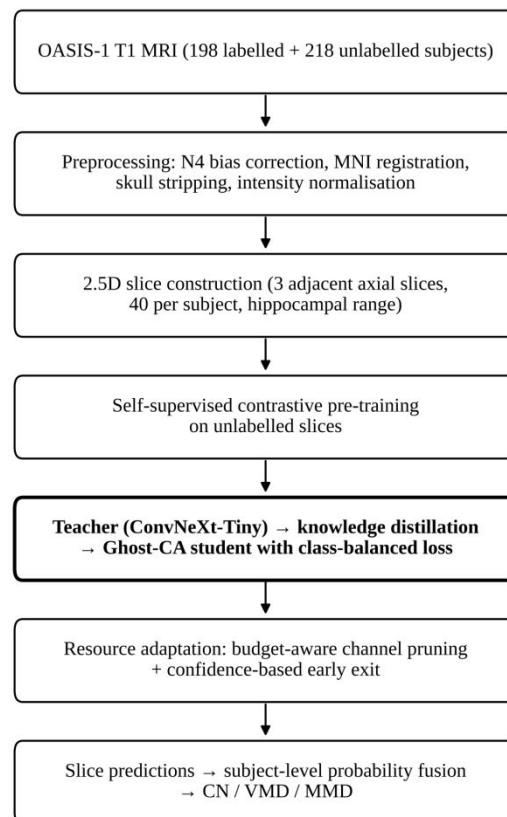


Figure 1. Workflow of the proposed ElasticAD framework

3.1 Dataset

The study uses the OASIS-1 cross-sectional collection [20], which provides T1-weighted MRI for 416 subjects aged 18 to 96. Labelled subjects were restricted to participants aged 60 and above with a CDR score, giving 198 subjects in three classes, as shown in Table 2. The remaining 218 subjects, mostly younger adults without a CDR rating, were used only as unlabelled data for self-supervised pre-training and never for evaluation.

Table 2. Labelled subjects used in this study

Class	CDR	Subjects	2.5D inputs
Cognitively normal (CN)	0	98	3,920
Very mild dementia (VMD)	0.5	70	2,800
Mild-to-moderate dementia (MMD)	≥ 1	30	1,200
Total		198	7,920

All experiments use stratified five-fold cross-validation at the subject level. Every input from a given subject belongs to exactly one fold, and within each training split 15% of subjects are held out for validation and early stopping. The test fold is never used for model selection, following the recommendations in [6].

3.2 Preprocessing and 2.5D Input Construction

1. **Bias correction:** intensity inhomogeneity is removed with the N4 algorithm [21].
2. **Registration and skull stripping:** volumes are affinely registered to the MNI152 template and non-brain tissue is masked out, so the same axial index covers similar anatomy in every subject.
3. **Intensity normalisation:** voxel intensities inside the brain mask are clipped at the 1st and 99th percentiles and scaled to $[0, 1]$ using Equation (1).
4. **2.5D construction:** from 40 consecutive axial slices covering the hippocampus and lateral ventricles, each slice is stacked with its upper and lower neighbours to form a three-channel image of 160×160 pixels.

$$x' = (x - x_{p1}) / (x_{p99} - x_{p1}) \quad (1)$$

The 2.5D input gives the network some through-plane context at almost the cost of a 2D model, and the three channels match the format expected by ImageNet-style networks.

3.3 Self-Supervised Contrastive Pre-Training

Before using any labels, the student encoder is pre-trained on 2.5D inputs from the unlabelled subjects and from the training subjects of the current fold, with their labels removed. Following SimCLR [16], two random views of each input are created with crops, small rotations, intensity jitter and Gaussian blur. A projection head maps encoder features to vectors z , and the normalised temperature-scaled cross-entropy loss pulls the two views together while pushing other inputs in the batch away:

$$\ell_{ij} = -\log [e^{s_{ij}/\tau} / \sum_{k \neq i} e^{s_{ik}/\tau}] \quad (2)$$

Here $s_{ij} = \text{sim}(z_i, z_j)$ is the cosine similarity between two projected vectors and $\tau = 0.2$. Pre-training runs for 150 epochs. The projection head is then discarded and the encoder initialises the student.

3.4 Student Architecture

The student is a narrow network of Ghost bottlenecks [13]. A Ghost module first produces m intrinsic feature maps Y' with an ordinary 1×1 convolution and then generates the remaining maps with cheap 3×3 depthwise operations Φ :

$$Y' = X * f, \quad Y = [Y', \Phi(Y')] \quad (3)$$

With a ratio $s = 2$ between total and intrinsic maps, each Ghost module needs roughly half the operations of a standard convolution of the same width. The student uses a 3×3 stem with 16 filters, 12 Ghost bottlenecks arranged in four stages with widths 16, 24, 48 and 96, and a classification head of global average pooling, dropout of 0.2 and a three-way softmax layer.

Coordinate attention [14] is inserted in the last two stages. It pools the feature map x separately along height and width, encodes both directions with a shared 1×1 convolution, and produces two attention maps g^h and g^w :

$$y_c(i, j) = x_c(i, j) \cdot g_c^h(i) \cdot g_c^w(j) \quad (4)$$

Because the attention keeps the row and column position, it can emphasise narrow structures such as the hippocampal formation and the temporal horns of the ventricles. The full student has 0.84 million parameters and needs 70 MFLOPs per 160×160 input.

3.5 Teacher-Guided Training with Class-Balanced Loss

A ConvNeXt-Tiny network [22] initialised with ImageNet weights is first fine-tuned on each training fold and used as the teacher. The student is then trained with a combination of a class-balanced loss and a distillation loss. The class-balanced weight for class c with n_c training inputs is given by Equation (5), with $\beta = 0.999$ [17]:

$$w_c = (1 - \beta) / (1 - \beta^{n_c}) \quad (5)$$

The total student loss combines the weighted cross-entropy L_{CB} with the Kullback–Leibler divergence between the softened teacher and student outputs at temperature $T = 4$ [15]:

$$L = (1 - \lambda) L_{CB} + \lambda T^2 \text{KL}(p_t^T \parallel p_s^T) \quad (6)$$

The weight $\lambda = 0.6$ was chosen on the validation subjects. Training uses the AdamW optimiser with a learning rate of 0.002, cosine decay, weight decay of 0.0005, batch size 64 and at most 80 epochs with early stopping after 12 epochs without validation improvement. Mixup with $\alpha = 0.2$ is applied as additional regularisation.

3.6 Resource Adaptation

ElasticAD adapts to hardware in two ways. **Budget-aware pruning** is applied after training. Following network slimming [18], the student is fine-tuned for 20 epochs with an L1 penalty on the batch-normalisation scaling factors γ :

$$L_{\text{prune}} = L + \eta \sum |\gamma| \quad (7)$$

Channels with the smallest γ values are removed until the model meets the FLOPs budget of the target device, and the pruned network is fine-tuned again for 15 epochs with distillation. Budgets of 100%, 60% and 30% of the original FLOPs produce the variants ElasticAD-100, ElasticAD-60 and ElasticAD-30. All variants are then converted to 8-bit integers with TensorFlow Lite.

Confidence-based early exit works at run time. A small auxiliary classifier is attached after the third stage, as in BranchyNet [19]. If its maximum softmax probability is at least $\tau_c = 0.90$, the prediction is returned and the last stage is skipped; otherwise the input continues through the full network. Easy inputs, such as clearly normal slices, therefore cost less computation than ambiguous ones.

3.7 Subject-Level Decision Fusion

The $K = 40$ inputs of a subject are classified independently, and their class probabilities are averaged to give the subject-level decision:

$$P(c) = (1/K) \sum_k p_k(c), \quad \hat{y} = \text{argmax}_c P(c) \quad (8)$$

Averaging reduces the effect of individual slices that show little atrophy and makes the final result a decision about a person rather than about an image.

4. Experimental Setup

The framework was implemented in Python 3.11 with PyTorch 2.3. Training ran on an NVIDIA RTX 3060 GPU with 12 GB of memory. Inference time was measured on two low-cost devices: a laptop with an Intel Core i5-1135G7 CPU and 8 GB RAM, and a Raspberry Pi 4 Model B with 4 GB RAM, using TensorFlow Lite with INT8 weights and a single thread. Each reported time is the mean of 1,000 runs after 50 warm-up runs.

Six baselines were trained with the same folds, preprocessing, 2.5D inputs, augmentation and subject-level fusion: ResNet18 [23], DenseNet121 [24], MobileNetV3-Small [10], ShuffleNetV2 [11], EfficientNet-B0 [12] and ConvNeXt-Tiny [22]. All baselines were initialised with ImageNet weights and trained with weighted cross-entropy.

Because the classes are imbalanced, the main metric is balanced accuracy, the mean of the per-class recalls, computed at the subject level. The macro-averaged F1-score is also reported:

$$BA = (1/C) \sum_c TP_c / (TP_c + FN_c) \quad (9)$$

$$F1_c = 2 P_c R_c / (P_c + R_c) \quad (10)$$

Here P_c and R_c are the precision and recall of class c . Efficiency is described by the number of parameters, floating-point operations per input, INT8 model size and inference latency. Results are means over the five folds.

5. Results and Discussion

5.1 Comparison with Baseline Models

ElasticAD achieved the highest subject-level balanced accuracy of all models, 85.3%, while being the second-smallest network by parameter count. Table 3 summarises accuracy and model cost, and Table 4 reports latency.

Table 3. Subject-level results (mean of five folds)

Model	Params (M)	MFLOPs	BA (%)	Macro F1
ResNet18	11.2	930	78.6	0.76
DenseNet121	7.0	1,460	80.9	0.79
MobileNetV3-Small	1.5	30	76.2	0.74
ShuffleNetV2	1.3	75	75.8	0.73
EfficientNet-B0	4.0	200	81.7	0.80
ConvNeXt-Tiny (teacher)	27.8	2,290	84.3	0.83
ElasticAD-100	0.84	70	85.3	0.85

Table 4. Inference time per 2.5D input (INT8, single thread)

Model	Laptop CPU (ms)	Raspberry Pi 4 (ms)
ResNet18	21.4	96.0
DenseNet121	38.7	210.0
EfficientNet-B0	11.9	58.0
ConvNeXt-Tiny	64.2	410.0
MobileNetV3-Small	3.1	14.0
ElasticAD-100	2.7	12.1

Two points stand out. First, the large ResNet18 and DenseNet121 models were less accurate than ElasticAD, which confirms that model size does not compensate for a small subject count. Second, ElasticAD was 1.0 point more accurate than its own teacher while being 33 times smaller. Distillation transferred the teacher's knowledge, and self-supervised pre-training gave the student additional features learned from the unlabelled scans that the teacher never used. MobileNetV3-Small needs fewer operations, but it was 9.1 points less accurate. On the Raspberry Pi 4, ElasticAD classifies a complete subject of 40 inputs in under 0.5 seconds.

5.2 Class-Wise Performance

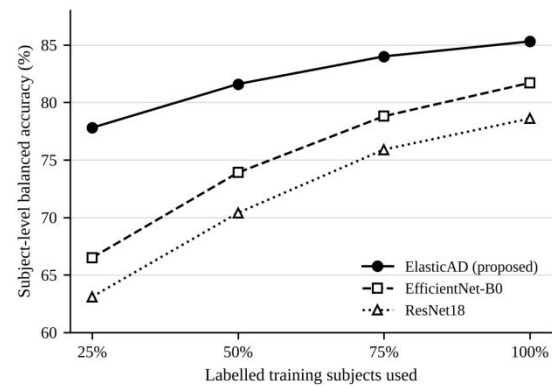
Table 5. Class-wise subject-level results of ElasticAD-100

Class	Precision	Recall	F1-score
CN	0.91	0.93	0.92
VMD	0.82	0.80	0.81
MMD	0.79	0.83	0.81
Macro average	0.84	0.85	0.85

Cognitively normal subjects were recognised most reliably. Most errors involved the very mild class: some VMD subjects were predicted as CN, and a few as MMD. This is clinically expected, since CDR 0.5 is a transition stage whose atrophy overlaps with normal ageing. Despite having only 30 subjects, the MMD class reached a recall of 0.83, which shows the benefit of the class-balanced loss. The overall subject-level accuracy was 86.9%.

5.3 Performance with Fewer Labelled Subjects

To test the framework under even scarcer labels, the models were retrained with 25%, 50% and 75% of the labelled training subjects of each fold, while the test folds stayed unchanged. Figure 2 shows the results.

**Figure 2. Balanced accuracy with different amounts of labelled training data**

With only 25% of the labelled subjects, about 40 people per fold, ElasticAD still achieved 77.8% balanced accuracy, 11.3 points above EfficientNet-B0 and 14.7 points above ResNet18. The gap narrows as more labels become available, which indicates that self-supervised pre-training on unlabelled scans is most valuable when labels are scarcest.

5.4 Ablation Study

Table 6. Contribution of each component

Configuration	BA (%)	Macro F1
Ghost student, ImageNet init., 2D input, cross-entropy	76.9	0.75
+ 2.5D input	78.8	0.77
+ Coordinate attention	80.1	0.78
+ Self-supervised pre-training	82.6	0.81
+ Class-balanced loss	83.7	0.82
+ Knowledge distillation (full ElasticAD)	85.3	0.85

Every component improved the result. Self-supervised pre-training gave the largest single gain of 2.5 points, followed by 2.5D input and knowledge distillation with 1.9 and 1.6 points. Together the components raised balanced accuracy by 8.4 points without changing the size of the deployed model.

5.5 Resource-Adaptive Operating Points

Table 7. ElasticAD variants on the Raspberry Pi 4

Variant	Par. (M)	MFL	MB	ms	BA (%)
ElasticAD-100	0.84	70	0.9	12.1	85.3
ElasticAD-100 + exit	0.86	44*	0.9	7.9*	85.0
ElasticAD-60	0.51	42	0.6	7.6	84.6
ElasticAD-30	0.26	21	0.3	4.2	82.1

*Average over all test inputs. Par. = parameters, MFL = MFLOPs, MB = INT8 model size, ms = Raspberry Pi 4 time per input. With early exit, 41% of inputs left the network at the auxiliary classifier. This cut average computation by 37% at a cost of only 0.3 points in balanced accuracy. Pruning to 60% of the FLOPs lost 0.7 points, and even the 30% variant, at 0.3 MB, remained more accurate than ResNet18. A deployment team can therefore choose an operating point that fits the device instead of retraining a new model.

5.6 Interpretability

Grad-CAM++ [25] maps were computed on the final Ghost stage for correctly classified test subjects. For VMD and MMD subjects, the

strongest activations lay over the hippocampus, the parahippocampal region and the temporal horns of the lateral ventricles, which are the areas where AD-related atrophy is expected. For CN subjects the maps were weaker and more diffuse. This agreement with known pathology supports the use of ElasticAD as a decision-support tool rather than an opaque classifier.

5.7 Limitations

The study uses a single cohort with a modest number of demented subjects, and the MMD class combines CDR 1 and CDR 2. Accuracy should be confirmed on external cohorts such as ADNI or AIBL acquired with different scanners. The framework also relies on registration to a template, which adds preprocessing time on embedded devices.

6. Conclusion and Future Scope

This paper presented ElasticAD, a lightweight and resource-adaptive deep learning framework for three-class Alzheimer's disease classification from limited structural MRI data. By combining 2.5D inputs, self-supervised pre-training on unlabelled scans, a Ghost-based student with coordinate attention, knowledge distillation and a class-balanced loss, ElasticAD reached 85.3% subject-level balanced accuracy on OASIS-1 with only 0.84 million parameters. Under strict subject-level cross-validation it outperformed six established CNNs, including its 28-million-parameter teacher. Budget-aware pruning and confidence-based early exit let the same framework run at 4 to 12 ms per input on a Raspberry Pi 4, so it can be deployed where GPUs are unavailable.

Future work will focus on:

- External validation on ADNI and AIBL subjects acquired on different scanners.
- Replacing template registration with a lightweight learned alignment step to reduce preprocessing time on edge devices.
- Predicting conversion from very mild dementia to AD using longitudinal OASIS-2 scans.

- Federated training across hospitals so that private scans never leave their site.
- Packaging ElasticAD as an offline mobile and desktop application for primary health centres.

References

- [1] Alzheimer's Association, "2024 Alzheimer's disease facts and figures," *Alzheimer's & Dementia*, vol. 20, no. 5, pp. 3708–3821, 2024.
- [2] G. B. Frisoni, N. C. Fox, C. R. Jack Jr., P. Scheltens and P. M. Thompson, "The clinical use of structural MRI in Alzheimer disease," *Nature Reviews Neurology*, vol. 6, no. 2, pp. 67–77, 2010.
- [3] J. C. Morris, "The Clinical Dementia Rating (CDR): Current version and scoring rules," *Neurology*, vol. 43, no. 11, pp. 2412–2414, 1993.
- [4] D. Lu, K. Popuri, G. W. Ding, R. Balachandar and M. F. Beg, "Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's disease using structural MR and FDG-PET images," *Scientific Reports*, vol. 8, 5697, 2018.
- [5] M. A. Ebrahimighahnavieh, S. Luo and R. Chiong, "Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review," *Computer Methods and Programs in Biomedicine*, vol. 187, 105242, 2020.
- [6] J. Wen et al., "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation," *Medical Image Analysis*, vol. 63, 101694, 2020.
- [7] S. Korolev, A. Safiullin, M. Belyaev and Y. Dodonova, "Residual and plain convolutional neural networks for 3D brain MRI classification," in *Proc. IEEE Int. Symp. Biomedical Imaging (ISBI)*, 2017, pp. 835–838.
- [8] M. Liu, J. Zhang, E. Adeli and D. Shen, "Landmark-based deep multi-instance learning for brain disease diagnosis," *Medical Image Analysis*, vol. 43, pp. 157–168, 2018.
- [9] S. Basaia et al., "Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks," *NeuroImage: Clinical*, vol. 21, 101645, 2019.
- [10] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV)*, 2019, pp. 1314–1324.
- [11] N. Ma, X. Zhang, H.-T. Zheng and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. European Conf. Computer Vision (ECCV)*, 2018, pp. 116–131.
- [12] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.
- [13] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1580–1589.
- [14] Q. Hou, D. Zhou and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13713–13722.
- [15] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.
- [16] T. Chen, S. Kornblith, M. Norouzi and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Machine Learning (ICML)*, 2020, pp. 1597–1607.
- [17] Y. Cui, M. Jia, T.-Y. Lin, Y. Song and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9268–9277.
- [18] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2736–2744.
- [19] S. Teerapittayanon, B. McDanel and H. T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, 2016, pp. 2464–2469.
- [20] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris and R. L. Buckner, "Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *Journal of Cognitive Neuroscience*, vol. 19, no. 9, pp. 1498–1507, 2007.
- [21] N. J. Tustison et al., "N4ITK: Improved N3 bias correction," *IEEE Transactions on Medical Imaging*, vol. 29, no. 6, pp. 1310–1320, 2010.
- [22] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11976–11986.
- [23] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [24] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.
- [25] A. Chattopadhyay, A. Sarkar, P. Howlader and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Applications of Computer Vision (WACV)*, 2018, pp. 839–847.