SN Computer Science (2024) 5:252
https://doi.org/10.1007/s42979-023-02581-8

ORIGINAL RESEARCH

An Optimal Weighted Ensemble of 3D CNNs for Early Diagnosis of Alzheimer's Disease

Sriram Dharwada[1] · Jitendra Tembhurne[2] · Tausif Diwan[2]

Received: 28 December 2022 / Accepted: 26 December 2023

Abstract

Alzheimer disease (AD) is a chronic neurological disorder in which the loss of brain cells causes dementia. Early and accurate diagnosis of AD will lead to better treatment of the disease before irreversible brain damage has been occurred. This paper proposes the classification of Alzheimer's disease using 3D structural Magnetic Resonance Imaging (sMRI) images through 3D convolutional neural networks (CNNs). Most existing methods utilizing 3D subject-level CNNs for Alzheimer's disease classification design a single model which relies on a very large training dataset for improved generalization. Herein, we address this issue through 3D transfer learning which makes use of knowledge gained from a pre-trained task. We train 3D versions of five classical 2D image classification architectures—ResNet, ResNeXt, SE-ResNet, SE-ResNeXt, and SE-Net—by initializing each model with pre-trained weights from their 2D counterparts, and combine their predictions through a weighted average method. The weights assigned to each model of the ensemble are optimized to achieve a performance better than any single 3D CNN model. With a relatively smaller training dataset, the proposed model obtains 97.27%, 82.33%, 90.41%, 84.22%, 84.26%, and 77.1% accuracies for the Alzheimer's disease (AD) versus cognitively normal (CN), early mild cognitive impairment (EMCI) versus CN, late mild cognitive impairment (LMCI) versus CN, EMCI versus AD, LMCI versus AD, and EMCI versus LMCI classification tasks, outperforming current state-of-the-art methods, and indicating the effectiveness of our proposed model.

Keywords Alzheimer's disease · Convolutional neural networks · Ensemble learning · Transfer learning

Introduction

Alzheimer's disease (AD) is an irreversible progressive neurodegenerative disorder that deteriorates memory cells and subsequently hampers other important brain functions. There are 50 million people worldwide suffering from dementia, and Alzheimer's disease contributes to 60–70% of those cases [1]. Alzheimer's disease has no cure and no treatment to stop its progression, yet it is essential to give diagnosis as early and accurately as possible. The mild cognitive impairment (MCI) is a transitional state between the normal aging and Alzheimer's disease, and MCI is most likely to be converted to AD later [1]. Based on the time of conversion to MCI, the MCI category is further divided into early MCI (EMCI) and late MCI (LMCI) [2]. Due to subtle differences in brain morphology and brain functions between subjects with MCI and subjects who are cognitively normal (CN), it is quite challenging to distinguish between MCI and CN. As such, MCI is often misdiagnosed as the symptoms of normal aging, which results in missing the timely treatment. Therefore, the accurate diagnosis of MCI is essential for the early diagnosis and treatment of AD [3]. The different stages of AD are shown in Fig. 1 through some sample brain MRI images.

Early diagnosis of AD and MCI can be successful in improving intellectual capacity, treating melancholy, improving guardian state of mind, and delaying institutionalization. It will permit individuals to prepare in advance

✉ Jitendra Tembhurne
jtembhurne@iiitn.ac.in

Sriram Dharwada
bt19cse058@iiitn.ac.in

Tausif Diwan
tdiwan@iiitn.ac.in

[1] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

[2] Department of Computer Science and Engineering, Indian Institute of Information Technology, Nagpur, Maharashtra 441108, India

Published online: 06 February 2024

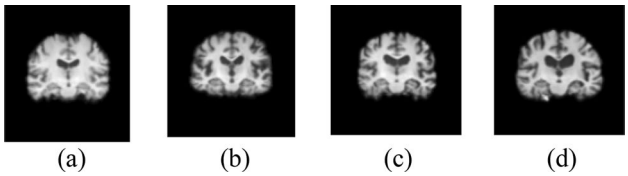


Fig. 1 The coronal view of brain MRI of subjects belonging to the four classes associated with Alzheimer's disease. **a** Cognitively normal (CN); **b** early mild cognitive impairment (EMCI); **c** late mild cognitive impairment (LMCI); **d** Alzheimer's disease (AD)

regarding their future care while they actually have the ability to settle on significant choices. In addition, they and their families can receive timely practical information, counsel, and backing [3].

MRI is a medical imaging technique which creates 3D representation of organs in the body using magnetic fields and radio waves. Since MRI scans can provide useful biomarkers such as patterns of atrophy, they have been widely used for AD detection. Latest trends in the current literature show that deep learning methods such as convolutional neural networks (CNNs) are very efficient in classifying subjects with Alzheimer's disease [25–40].

Motivation

2D CNNs have become quite mainstream for AD detection because they are computationally inexpensive to train and can support transfer learning and an increased dataset size as multiple 2D slices can be extracted from a single 3D scan. However, they are not efficient in encoding the latent information of the 3D images due to the absence of kernel sharing across the third dimension. It has also been found by Wen et al. [25] that data leakage is extremely common studies that employ 2D CNNs for AD classification. In studies [41, 42], train-test split would be done at the slice level and not the subject level, i.e., the 2D slices from the same subject's MRI scan would appear in both train and test sets, leading to biased results.

3D subject-level CNNs are an excellent substitute to 2D CNN, as they can fully encode the spatial dependencies between adjacent slices in an efficient manner. They take, as input, the whole 3D MRI scan and can sufficiently express the connections in the huge interconnected network. Since the whole MRI is used at once, classification is performed at subject level, avoiding any data leakage. In recent years, transfer learning techniques have been introduced to 3D CNNs to effectively utilize the resources and improve efficiency [11, 12]. Through transfer learning, we can acquire the knowledge that neural networks learned on one task and utilize the same for another task. By initializing the 3D CNNs with the weights of 2D CNNs that were pre-trained on large and general datasets such as ImageNet [7], the feature maps learned by the pre-trained model can be utilized for the classification of ADs on our dataset.

Another popular strategy that has been known to boost the performance of not only 3D CNNs, but also any machine learning or deep learning model, is to train an ensemble of models and combine their predictions [21–23]. A standalone classifier would approach a given problem in only one manner and can suffer from a high generalization error. However, multiple such classifiers can work together to make better predictions and improve generalization. A relatively simple way to combine the predictions of all the models is an equally weighted average where every single classifier has a same weightage in the final prediction. But sometimes we might want some exceptional performers to contribute more, and some inferior ones to contribute less, and this is done using an optimally weighted average. Each model is assigned a weight that is proportional to its performance on the validation set, allowing better performing models to contribute more to the final prediction.

Approach and Outcome

In this study, we examine an optimally weighted average ensemble of 3D CNN architectures for six binary classification tasks related to AD classification—AD versus CN, EMCI versus CN, LMCI versus CN, EMCI versus AD, LMCI versus AD, and EMCI versus LMCI—using data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) project. The 3D CNNs are based on the residual neural networks [13] and their variations [14, 15] with their convolutional filters bootstrapped from their 2D variants pre-trained on ImageNet. The optimal weights assigned to each classifier of the ensemble such that the weighted average of their predictions would result in the highest possible accuracy are found through the Sequential Quadratic Programming (SQP) algorithm [24].

Main Contributions

The main contributions of this paper are as follows:

1. For ADs detection, we propose a weighted average ensemble of the 3D variants of the ResNet-50, ResNeXt-50, SE-ResNet-50, SE-ResNeXt-50, and SE-Net-154 models.
2. We applied 3D transfer learning to the aforementioned five models to maximize their efficiency and performance.
3. By finding the optimal contribution of each model to the final prediction, we were able to bring forth an ensemble accuracy greater than any single model and an equally weighted ensemble.

4. The proposed framework learns from a small dataset and still demonstrates superior generalized performance.

The rest of this paper is organized as follows. In "Literature Review" section, we review related literature. Next in "Materials and Methods" section, we describe the data selection, data augmentation techniques, our proposed model, and the performance metrics in detail. Then in "Results and Discussion" section, we present our experimental results and discussion over the obtained results. Finally, we conclude the paper in "Conclusions" section and present future research directions.

Literature Review

In recent years, numerous studies based on deep learning, especially convolutional neural networks, have been proposed to assist in the diagnosis of AD.

A modern and systematic review of state-of-the-art on classification of Alzheimer's disease using CNNs by Wen et al. [25] pointed out that a significant number of studies reported biased results due to data leakage. The authors then performed rigorous assessment of various CNN frameworks and observed that the 3D approaches significantly outperformed the 2D approaches for AD classification. The highest accuracy (88% on the AD vs. CN classification task) was obtained by the 3D ROI based approach wherein the left and right hippocampus were chosen as ROIs. Moreover, they displayed the generalizability of their models by training them on the ADNI dataset and evaluating them on the AIBL [4] and OASIS [5] datasets.

Classical image classification architectures such as VGG, ResNet, and Inception have extensively been adapted to the third dimension for AD classification. Korolev et al. [26] derived two 3D architectures—VoxCNN and VoxResNet from smaller versions of VGG and ResNet, respectively, for AD classification. The authors did not use complex preprocessing, handcrafted feature generation and complex model stacking. While they were able to set a baseline accuracy of 80% on the AD versus CN task, they achieved only 52% and 56% accuracies on the EMCI versus LMCI and EMCI versus CN tasks, respectively. The authors of [30] utilized an inception module based 3D convolutional autoencoder for the AD classification task. They later transferred the representations of the AD versus CN classifier to the pMCI (Progressive MCI) versus sMCI (Stable MCI) classification task and achieved 86.60% and 73.95% on the AD and pMCI classification tasks, respectively. A 11 layered 3D CNN based on the VGG 512 architecture called ADNet was proposed by Folego et al. [31]. They used domain adaptation to adapt ADNet, trained on the ADNI dataset, to ADNet-DA which was used for classification on the CADDementia dataset [6]. While they didn't make use of any prior information about AD, they achieved only 52.3% accuracy on the CADDementia dataset.

One of the current trends in AD classification is to use transfer learning which helps to save resources and improve efficiency, and it has been used extensively for various 2D CNN methods. On the OASIS dataset, Hon et al. [40] employed the VGG-16 and InceptionV4 architectures that were pre-trained on the ImageNet dataset, and repurposed them for the AD classification problem. With a training size 10 times smaller than other state-of-the-art, an accuracy of 96.25% was obtained on the AD versus CN task. More recently, Bae et al. [28] used a 2D CNN transfer learning approach for the AD versus CN task. They extracted 30 2D slices from the coronal view of the brain per subject. The 30 slices were independently fed into an InceptionV4 model pre-trained on ImageNet, and the results averaged to classify a single subject. They obtained MRI scans from the ADNI dataset, and a custom dataset from the Seoul National University Bundang Hospital (SNUBH), trained their proposed model individually on each dataset, and performed within-dataset validation as well as between-dataset validation. Both between-dataset validation accuracies were above 80% indicating their high generalizability.

In recent years, transfer learning has been introduced to 3D CNNs as well. For instance, Hara et al. [11] showed that similar to 2D CNNs pre-trained on ImageNet for image recognition, simple 3D CNNs pre-trained on the Kinetics dataset [8] could achieve remarkable advancements in action recognition and related tasks. Prior to them, the authors of [12] bootstrapped the filters of an ImageNet pre-trained Inception-V1 model on to a "Two-Stream Inflated 3D ConvNet" (I3D), trained it on the Kinetics dataset, and finally fine-tuned it on the HMDB-51 [9] and UCF-101 [10] action recognition datasets, setting benchmark results on them. The technique of bootstrapping the filters of a 2D CNN to a 3D CNN has been employed for AD classification by Ebrahimi et al. [29] who initialized a 3D ResNet-18 with weights of a 2D ResNet-18 pre-trained on ImageNet for the AD versus CN task, and reported a 28.13% boost in accuracy using transfer learning as compared to training from scratch.

Showing contrast to training a single model, the usage of an ensemble of CNNs for AD detection has been substantially explored in the past. Islam et al. [33] constructed a max-voting ensemble of three variants of the DesneNet [19] architecture for AD classification on the OASIS dataset, outperforming single models such as ResNet [13], Inception-v4 [20], and ADNet [39] on the same dataset. The authors of [32] first sliced the 3D MRI into 123 sagittal, coronal, and transverse 2D images, and trained a base 2D CNN AD versus CN classifier for each of those images, resulting in 123 base models. Using only the best 5 performing models from each view, three single-axis ensembles were built. Finally, a

three-axis ensemble was created on top of these three single-axis ensembles to give subject-level prediction. A weighted average ensemble of 2D CNNs for AD detection was developed by Choi et al. [34]. On 2D slices from each of the three views of the brain, three deep learning models—VGG-16 [16], GoogLeNet [17], and AlexNet [18]—were trained. A deep ensemble generalization loss which accounted for the interaction and cooperation between the deep models was created, and the sequential quadratic programming algorithm was used to find the optimum weights of each model. This method helped them achieve a staggering AD versus CN versus MCI multi-class accuracy of 93.84%. In [45], a multi-channel 2D CNN is presented to extract the 3D sMRI for the classification AD. The CNN model is trained on the different planes of the view for the feature extractions and achieved the accuracy of 98.33%.

Inspired by the success of 3D CNNs, transfer learning, and ensemble learning, in our work, we propose an optimal weighted average ensemble of 3D CNNs that use pre-trained weights from 2D CNNs trained on the ImageNet dataset.

Materials and Methods

This section provides the details of our pipeline comprising data selection, data augmentation, the proposed CNN architectures, 3D transfer learning, the optimal weighted average ensemble model, and performance metrics for evaluating the proposed model.

Data Selection

For our work, we utilized structural MRI data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. Specifically, T1-weighted MRI data marked as "Spatially Normalized, Masked, and N3 corrected T1 images" are utilized. These images were already preprocessed with alignment and skull-stripping and were of size $110 \times 110 \times 110 \times 1$. A total of 228 subjects (50 Alzheimer's disease, 62 cognitively normal, 39 late mild cognitive impairment, 77 early mild cognitive impairment) belonged to this dataset. General inclusion/exclusion criteria are as follows:

1. Cognitively normal (CN) subjects: MMSE (Mini-Mental State Examination) scores between 24 and 30 (inclusive), a Clinical Dementia Rating (CDR) of 0, non-depressed, non-MCI (full form), and non-demented.
2. Mild cognitive impairment (MCI) subjects: MMSE scores between 24 and 30 (inclusive), a memory complaint, objective memory loss measured by education adjusted scores on Wechsler Memory Scale Logical Memory II, a CDR of 0.5, absence of significant levels of impairment in other cognitive domains, essentially preserved activities of daily living, and an absence of dementia. MCI is further divided into early mild cognitive impairment (EMCI) and late mild cognitive impairment (LMCI) based on the time at which a subject progressed to MCI.
3. Alzheimer's disease (AD): MMSE scores between 20 and 26 (inclusive), CDR of 0.5 or 1.0, and meets the National Institute of Neurological and Communicative Disorders and Stroke (NINCDS) and the Alzheimer's Disease and Related Disorders Association (ADRDA) criteria for probable AD.

We discarded 2 CN, 6 LMCI, and 1 EMCI subjects that did not fit into this criterion. The demographics of the subjects are presented in Table 1. Moreover, one MRI scan was used per subject to avoid data leakage. In order to save memory, all images were down sampled to $55 \times 55 \times 55 \times 1$.

Data Augmentation

Due to the limited amount of data, data augmentation was applied on the fly to the training images using the TorchIO library [43] to make our models learn efficient features that remained unaffected by changes in geometry and intensity. Wanting to simulate anatomical variations and artifacts produced by the MRI scanner, the following three data augmentation strategies were applied in combination at random to the input MRIs:

Rotation—Rotation of the input image with the rotation angle set to a random value in the range $[-15°$ to $+15°]$. The axis of rotation would be chosen randomly from Left-Right, Anterior–Posterior, and Superior-Inferior axes.
Shift—The image would be shifted along any random axis with the shift value set to a random value in the range $[-3, +3]$.
Zoom—The whole image would be randomly zoomed in or out by a random value 'z' in the range [0.85, 1.5]. The input image would be zoomed out if z would be less than 1, and zoomed in if z would be greater than 1. The same zoom factor would be applied on all three axes.

Table 1 Subject Demographics

	AD (50)	CN (60)	EMCI (76)	LMCI (33)
Female/male	20/30	33/27	12/21	32/44
Age	75.62 ± 8.55	73.72 ± 6.27	73.58 ± 6.01	74.3 ± 7.9
MMSE	22 ± 4	29 ± 1	26 ± 2	28 ± 2
CDR	1	0	0.5	0.5

Proposed Model

For this study, we used five popular pre-trained image classification models namely ResNet, ResNeXt, SE-Net, SE-ResNet, and SE-ResNeXt and inflated them to the third dimension in order to be able to take a whole 3D MRI scan as input. All models that we have chosen for this study are based on ResNet and its extended variations due to their simple architectures and improvised performances. We trained each model individually first and computed a weighted average of each model's prediction to generate the subject-level prediction.

We first discuss these five CNN architectures and the main intuition for adopting these particular architectures. Then we describe the process of bootstrapping 3D parameters of these models from their 2D counterparts pre-trained on ImageNet to leverage the knowledge gained by the 2D models by training on ImageNet for 3D medical image analysis. Finally, the weighted average ensemble and the process of finding the optimal weights for each model is explained.

CNN Architectures

(a) **ResNet:** With an increase in network depth, accuracy gets saturated and then degrades eventually. Intuitively, if a shallow network is able to achieve an optimal accuracy, a deeper network should also work well by simply learning to compute identity functions in the newly added layers. However, in these newly added layers, it is difficult for the model to exactly learn the identity mapping, as it is one of the countless solutions that the network can arrive at, causing accuracy to decrease with an increase in the network's depth. The authors of ResNet [8] solved this by introducing residual or skip connections to provide an alternative pathway for data and gradients to flow. The skip connections made (see Fig. 2) it easy for residual blocks to learn the identity function, allowing the authors to train deeper neural networks while having fast convergence.

(b) **ResNeXt:** ResNext [9] is a highly flexible and simple image classification model developed by Facebook AI Research in 2017. The authors noted it would be difficult to adapt the popular image classification architecture Inception to new datasets and tasks given that the Inception architecture requires intricate customization of hyper-parameters at each stage. To overcome this, they took inspiration from VGG and ResNet to improve upon the limitations of the Inception architecture and introduced ResNeXt. The ResNeXt architecture makes use of the repeating structures as in VGG, split-merge strategy like in Inception, and the residual connections from ResNet. A ResNeXt block splits the input into a number of uniform branches, transforms each branch using multiple-sized convolutions, merges the outputs of each branch, and adds the input to the result using a skip-connection. The number of branches called cardinality is the next dimension on top of the depth and width of ResNet (see Fig. 3). The ResNeXt architecture requires lesser hyper-parameters than Inception because each block follows the same topology.

(c) **Squeeze and Excitation Networks:** In the traditional convolutional operation, while constructing the output feature maps, the network weighs each of its trainable convolutional filters equally. Hu et al. [10] modified this by adding channel attention in order to prioritize certain channels over others. This was done using a Squeeze-Excitation block. The SE Block first produces a channel descriptor by aggregating feature maps across their spatial dimension (see Fig. 4). In other words, global average pooling is used to squeeze each input channel into a single value resulting in one neuron for each channel. Next, a fully connected layer is used to reduce the dimensions by a certain factor r, and is followed by a ReLU layer to introduce some non-linearity.

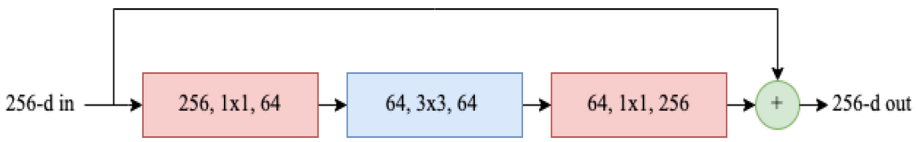


Fig. 2 A Residual Block. A layer is shown as (#in channels, filter size, #out channels)

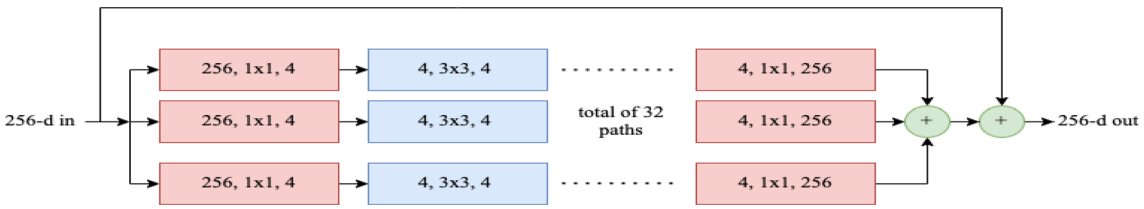


Fig. 3 A ResNeXt Block with cardinality of 32

Subsequently, another fully connected layer is used to project the output back to the original dimensions. Finally, these outputs are passed through a sigmoid function to obtain a weighted tensor, which tells importance of each channel. This tensor is then broadcasted and multiplied element-wise with each feature map of the CNN block. The SE Block can be integrated with the ResNet and ResNeXt architectures to construct SE-ResNet and SE-ResNeXt, respectively, as shown in Figs. 5 and 6.

In this work, we considered the ResNet-50, ResNeXt-50 (32×4d), SE-ResNet-50, SE-ResNeXt-50 (32×4d), and SE-Net-154, variants of the aforementioned five architectures, and modified them to their 3D versions for working with 3D MRI scans, by inflating the two-dimensional convolutional filters and pooling kernels to the third dimension. Given that we were interested in binary classification tasks only, the output of last layer was then passed through a sigmoid activation function. The architectures of the five models are shown in Table 2.

3D Transfer Learning

For all five models, we used parameters from their respective 2D variants that were pre-trained on the ImageNet dataset as described by [7]. Given a 2-dimensional convolutional weight filter pre-trained on ImageNet, it can be converted into its respective 3-dimensional weight filter, by simply replicating it N times along the time dimension, and rescaling the duplicated filters by dividing them by N, where N is the number of frames in the time dimension (see Fig. 7). As we are using pre-trained models which are mostly trained on colored input and using grayscale images might not work well without significant retraining or adaptation. For optimal utilization of the transfer learning, we are converting 2D variants of the input to its equivalent 3D counterpart for all the experimentation, results, and analysis.

Now, the images in the ImageNet dataset are RGB images with dimensions Height×Width×3. In order to get the same convolutional response from the 3D inflated CNNs, the input to these 3D CNNs must be a sequence of RGB images, i.e., the input must be of size N×Height×Width×3, where N is the number of frames along the time dimension. However, the 3D MRI scans in the ADNI dataset are all grayscale and have channel dimension equal to 1. Aiming to make

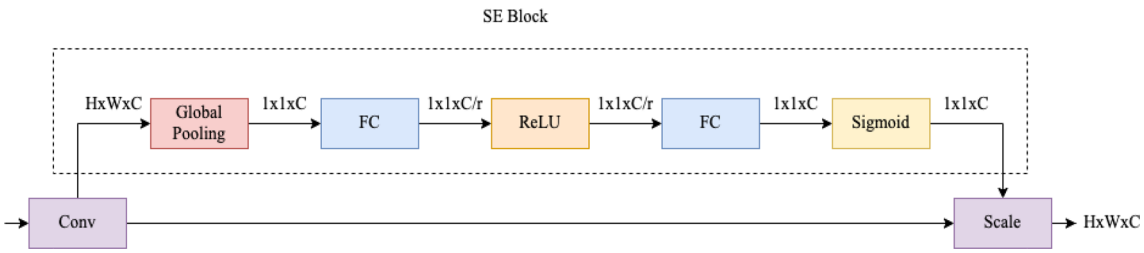
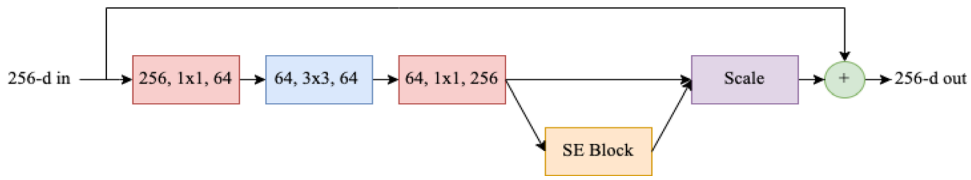


Fig. 4 A Squeeze-Excitation (SE) Block

Fig. 5 An SE-ResNet Block



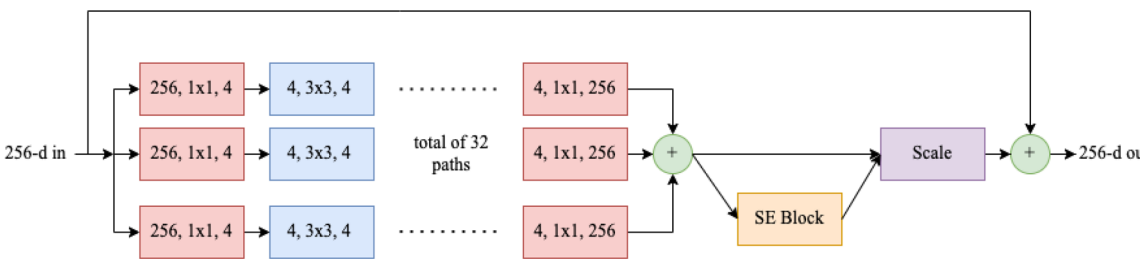


Fig. 6 An SE-ResNeXt Block

Table 2 The Proposed 3D CNN Architectures

Stage	3D ResNet-50	3D ResNeXt-50 (32×4d)	3D SE-ResNet-50	3D SE-ResNeXt-50 (32×4d)	3D SE-Net-154
conv1	conv, 7×7×7, 64, stride 2				$conv, 3\times3\times3, 64, stride2$ $conv, 3\times3\times3, 64$ $conv, 3\times3\times3, 128$
conv2	maxpool, 3×3×3, stride 2				
	$\begin{bmatrix} conv, 1\times1\times1, 64 \\ conv, 3\times3\times3, 64 \\ conv, 1\times1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 128 \\ conv, 3\times3\times3, 128, C=32 \\ conv, 1\times1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 64 \\ conv, 3\times3\times3, 64 \\ conv, 1\times1\times1, 256 \\ fc, [16, 256] \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 128 \\ conv, 3\times3\times3, 128, C=32 \\ conv, 1\times1\times1, 256 \\ fc, [16, 256] \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 128 \\ conv, 3\times3\times3, 256, C=64 \\ conv, 1\times1\times1, 256 \\ fc, [16, 256] \end{bmatrix}\times3$
conv3	$\begin{bmatrix} conv, 1\times1\times1, 128 \\ conv, 3\times3\times3, 128 \\ conv, 1\times1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} conv, 1\times1\times1, 256 \\ conv, 3\times3\times3, 256, C=32 \\ conv, 1\times1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} conv, 1\times1\times1, 128 \\ conv, 3\times3\times3, 128 \\ conv, 1\times1\times1, 512 \\ fc, [32, 512] \end{bmatrix}\times4$	$\begin{bmatrix} conv, 1\times1\times1, 256 \\ conv, 3\times3\times3, 256, C=32 \\ conv, 1\times1\times1, 512 \\ fc, [32, 512] \end{bmatrix}\times4$	$\begin{bmatrix} conv, 1\times1\times1, 256 \\ conv, 3\times3\times3, 512, C=64 \\ conv, 1\times1\times1, 512 \\ fc, [32, 512] \end{bmatrix}\times8$
conv4	$\begin{bmatrix} conv, 1\times1\times1, 256 \\ conv, 3\times3\times3, 256 \\ conv, 1\times1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} conv, 1\times1\times1, 512 \\ conv, 3\times3\times3, 512, C=32 \\ conv, 1\times1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} conv, 1\times1\times1, 256 \\ conv, 3\times3\times3, 256 \\ conv, 1\times1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix}\times6$	$\begin{bmatrix} conv, 1\times1\times1, 512 \\ conv, 3\times3\times3, 512, C=32 \\ conv, 1\times1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix}\times6$	$\begin{bmatrix} conv, 1\times1\times1, 512 \\ conv, 3\times3\times3, 1024, C=64 \\ conv, 1\times1\times1, 1024 \\ fc, [64, 1024] \end{bmatrix}\times36$
conv5	$\begin{bmatrix} conv, 1\times1\times1, 512 \\ conv, 3\times3\times3, 512 \\ conv, 1\times1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 1024 \\ conv, 3\times3\times3, 1024, C=32 \\ conv, 1\times1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 512 \\ conv, 3\times3\times3, 512 \\ conv, 1\times1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 1024 \\ conv, 3\times3\times3, 1024, C=32 \\ conv, 1\times1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix}\times3$	$\begin{bmatrix} conv, 1\times1\times1, 1024 \\ conv, 3\times3\times3, 2048, C=64 \\ conv, 1\times1\times1, 2048 \\ fc, [128, 2048] \end{bmatrix}\times3$
	Global average pool 1-d fc, Sigmoid				
#params	46.2×10^6	25.8×10^6	48.7×10^6	28.4×10^6	170.2×10^6

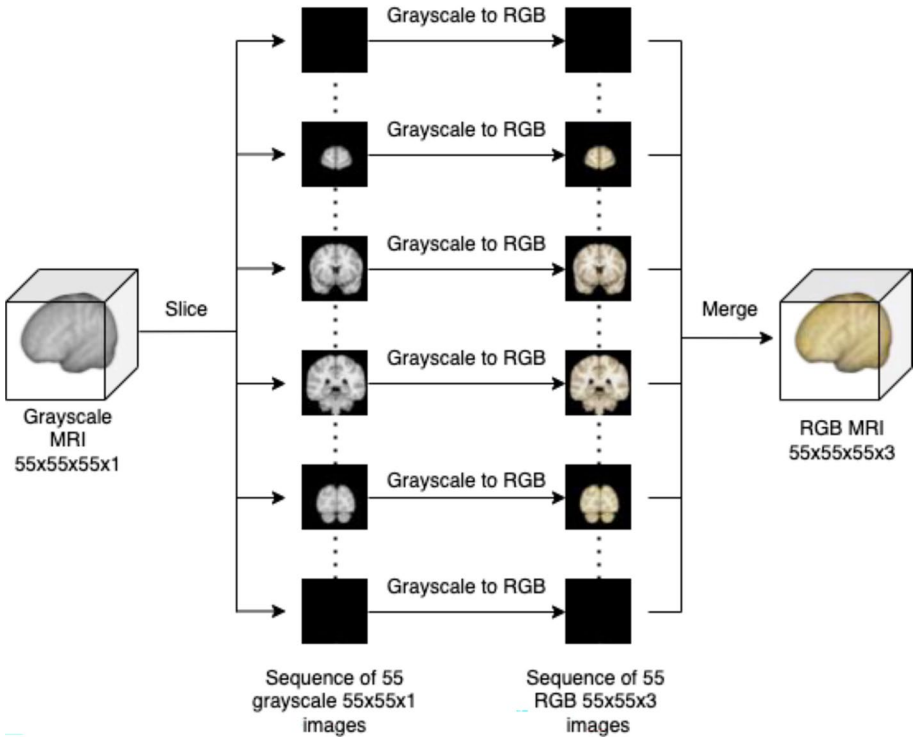


Fig. 7 The procedure of converting a Grayscale MRI to an RGB MRI for 3D transfer learning that employs the weights of a 2D CNN pre-trained on ImageNet

use of the knowledge gained from pre-training on the RGB ImageNet dataset, we cut up the resized 3D $55 \times 55 \times 55 \times 1$ MRI scans along the time dimension into 55 slices of shape $55 \times 55 \times 1$. Each $55 \times 55 \times 1$ grayscale slice was converted to an RGB image of shape $55 \times 55 \times 3$ by tripling the grayscale slices into three channels. Then, all the 55 RGB slices were put back together to form a sequence of 55 RGB images. We call this resultant of shape $55 \times 55 \times 55 \times 3$ an RGB MRI, and this process allowed for the 3D CNNs to apply features learned on the ImageNet dataset to the AD classification task.

Weighted Average Ensemble and Optimal Weights

For a given binary classification task, all the 5 models—3D ResNet-50, 3D ResNeXt-50, 3D SE-ResNet-50, 3D

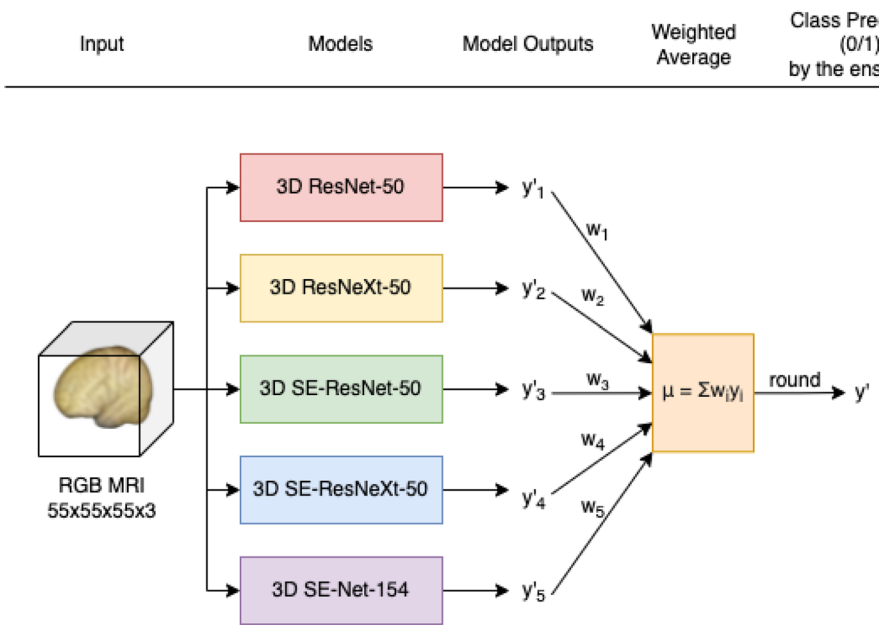


Fig. 8 The proposed Weighted Average Ensemble framework that is used to obtain the class prediction on a single subject

SE-ResNeXt-50, and 3D SE-Net-154—were assigned a weight that would tell how relevant that model's contribution was to the final prediction (see Fig. 8). The weights were non-negative values whose sum was equal to 1. The weights were learned and optimized by minimizing a loss function that represented the inaccuracy or entropy of the ensemble. To get predictions on a single subject, the weighted average of the class predictions of all 5 models was computed, and then rounded to the nearest integer—0 (negative class) or 1 (positive class) to get the class prediction by the ensemble. Fivefold cross-validation was employed to train each model of the ensemble.

Without loss in generality, the 3D MRIs were split into K folds using K-fold cross validation and the ensemble consisted of M models for all 6 binary classification tasks. For each of the k folds, all the M models would train using K-1 folds and make predictions on the holdout fold. This resulted in M prediction vectors for each fold. The models were assigned weights $w_i (i = 1, \ldots, M)$ ($w_i \geq 0$ and $\Sigma w_i = 1$) that would reflect their importance in the ensemble. The weighted average of the M prediction vectors was computed to get a single prediction vector on each fold. The task was to optimize the weights assigned to each model such that mean accuracy across all K folds would be maximum.

We define $\hat{y}_i(x)$ to be the output of model i given an image input x. $\hat{y}_i(x)$ describes the probability that x belongs to the positive class according to model i. Given a fold k, assuming that there are N images (x_k [1], …, x_k [N]) contained in its validation set, each model $m_i (i = 1, \ldots, M)$ computes a vector of predicted probabilities as [[$\hat{y}_1$ (x_k [1]), …, $\hat{y}_1$ (x_k [N])], [$\hat{y}_2$ (x_k [1]), …, $\hat{y}_2$ (x_k [N])], …, [$\hat{y}_M$ (x_k [1]), …, $\hat{y}_M$ (x_k [N])]]. The weighted average of the M vectors for fold k, μ_k, is calculated as:

$$\mu_k = \left[\sum_{i=1}^{M} w_i * \hat{y}_i(x_k[1]), \sum_{i=1}^{M} w_i * \hat{y}_i(x_k[2]), ..., \sum_{i=1}^{M} w_i * \hat{y}_i\left(x_k[N]\right) \right] \tag{1}$$

The kth fold's loss, l_k, is then calculated as:

$$l_k = \frac{1}{N}\left(\sum_{i=1}^{N} \left| y[i] - \text{round}\left(\mu_k[i]\right) \right| \right) \tag{2}$$

l_k calculates the number of incorrect predictions made by the weighted average ensemble on fold k. Repeating this process for all K folds yields in a set of losses $\{l_k: k = 1, \ldots, K\}$. The final loss function is the mean of losses computed for all K folds given by:

$$L(W) = \frac{1}{K}\left(\sum_{k=1}^{K} l_k \right) \tag{3}$$

The loss function L is a metric of inaccuracy (1—accuracy) of the weighted average ensemble across all k folds and has to be minimized. It is a function of the set of weights $W = \{w_i : i = 1, \ldots, M\}$. We define the following optimization problem:

$$\min_x L(W) \quad \text{subject to} \quad \sum_{i=1}^{M} w_i = 1 \quad \text{and} \quad w_i \geq 1\forall \quad 1 \leq i \leq M \tag{4}$$

Solve Eq. 4 using the Sequential Quadratic Programming (SQP) algorithm [24] provided as "Sequential Least SQuares Programming" (SLSQP) by the Scientific Python (SciPy) library [44]. Through the SQP algorithm, the optimum weights for each deep model are found such that the inaccuracy of the weighted average ensemble across all k folds would be minimized (or accuracy maximized), thus allowing the weighted average of the predictions of the M models to perform better than or as good as any single model or an equally weighted ensemble.

Table 3 Performance of the 5 3D CNNs on AD versus CN classification task with and without transfer learning

Model	Training method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
3D ResNet-50	From scratch	90.91 ± 2.87	92.05 ± 4.88	91.67 ± 5.27	91.65 ± 2.56	90.0 ± 6.32
	Transfer learning	94.55 ± 4.45	98.33 ± 3.33	91.67 ± 7.45	94.7 ± 4.34	98.0 ± 4.0
3D ResNeXt-50	From scratch	89.09 ± 5.45	92.03 ± 6.9	88.33 ± 8.5	89.73 ± 5.2	90.0 ± 8.94
	Transfer learning	92.73 ± 3.64	95.0 ± 4.08	91.67 ± 5.27	93.18 ± 3.42	94.0 ± 4.9
3D SE-ResNet-50	From scratch	91.82 ± 4.45	93.66 ± 5.57	91.67 ± 7.45	92.37 ± 4.25	92.0 ± 7.48
	Transfer learning	94.55 ± 5.3	96.52 ± 4.27	93.33 ± 6.24	94.86 ± 5.02	96.0 ± 4.9
3D SE-ResNeXt-50	From scratch	87.36 ± 7.96	88.28 ± 11.23	86.33 ± 4.08	88.06 ± 4.83	86.0 ± 4.0
	Transfer learning	91.82 ± 4.45	94.85 ± 4.22	90.0 ± 6.24	92.24 ± 4.26	94.0 ± 4.9
3D SE-Net-154	From scratch	88.18 ± 6.17	94.55 ± 4.45	83.33 ± 10.54	88.17 ± 6.5	94.0 ± 4.9
	Transfer learning	94.55 ± 3.4	94.29 ± 7.0	96.67 ± 4.08	95.18 ± 2.83	92.0 ± 9.8

Performance Matrices

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

The performance of the classifier is reported using the following metrics represented by Eqs. 5–9:

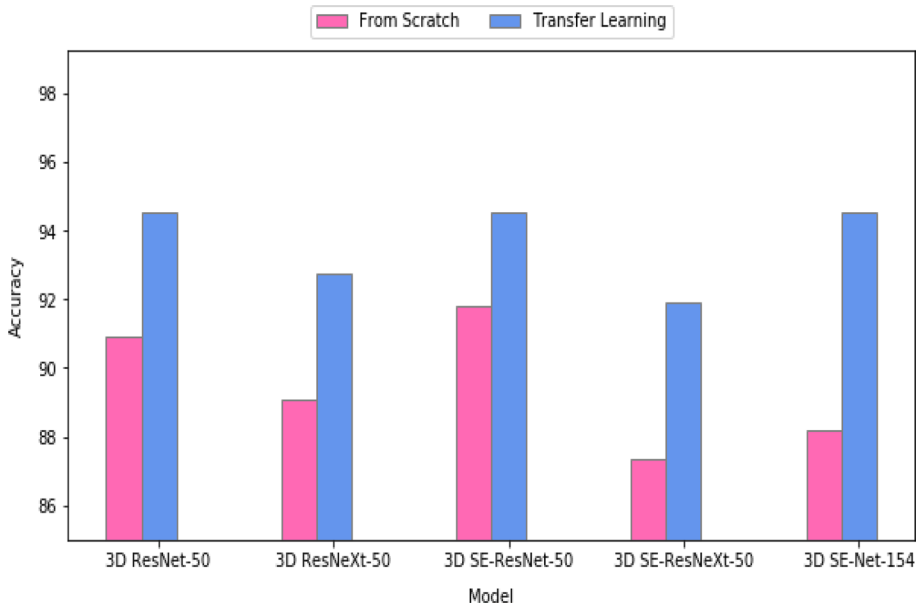


Fig. 9 Improvement in accuracy on the AD versus CN Classification task through transfer learning

Table 4 Performance of ensemble of 5 3D CNNs on AD versus CN Classification Task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.29678286	3D ResNet-50	94.55 ± 4.45	98.33 ± 3.33	91.67 ± 7.45	94.7 ± 4.34	**98.0 ± 4.0**
0.09608975	3D ResNeXt-50	92.73 ± 3.64	95.0 ± 4.08	91.67 ± 5.27	93.18 ± 3.42	94.0 ± 4.9
0.267783	3D SE-ResNet-50	94.55 ± 5.3	96.52 ± 4.27	93.33 ± 6.24	94.86 ± 5.02	96.0 ± 4.9
0.02428686	3D SE-ResNeXt-50	91.82 ± 4.45	94.85 ± 4.22	90.0 ± 6.24	92.24 ± 4.26	94.0 ± 4.9
0.31505753	3D SE-Net-154	94.55 ± 3.4	94.29 ± 7.0	**96.67 ± 4.08**	95.18 ± 2.83	92.0 ± 9.8
	Equally weighted average	96.36 ± 3.4	**98.33 ± 3.33**	**96.67 ± 4.08**	96.59 ± 3.14	**98.0 ± 4.0**
	Optimal weighted average	**97.27 ± 3.64**	**98.33 ± 3.33**	**96.67 ± 4.08**	**97.46 ± 3.35**	**98.0 ± 4.0**

Bold values indicate the optimal results obtained by the proposed models

Table 5 Performance of ensemble of 5 3D CNNs on EMCI versus CN Classification Task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.14870427	3D ResNet-50	74.97 ± 4.44	72.44 ± 6.99	71.67 ± 6.67	71.68 ± 4.21	77.5 ± 8.2
0.07614262	3D ResNeXt-50	74.21 ± 6.78	71.99 ± 11.64	**73.33 ± 3.33**	71.96 ± 4.77	74.83 ± 13.75
0.24526926	3D SE-ResNet-50	74.95 ± 3.87	76.8 ± 12.41	66.67 ± 7.45	70.23 ± 2.42	81.33 ± 11.47
0.4477129	3D SE-ResNeXt-50	76.43 ± 3.96	71.02 ± 6.12	71.67 ± 9.72	75.29 ± 3.53	72.33 ± 10.73
0.08217095	3D SE-Net-154	74.23 ± 3.5	72.94 ± 8.91	71.67 ± 15.46	70.46 ± 5.66	76.08 ± 13.83
	Equally weighted average	77.91 ± 7.48	77.11 ± 11.3	71.67 ± 15.46	74.83 ± 7.41	81.42 ± 10.76
	Optimal weighted average	**82.33 ± 5.5**	**81.18 ± 9.23**	71.67 ± 15.46	**80.24 ± 4.95**	**84.08 ± 10.06**

Bold values indicate the optimal results obtained by the proposed models

Table 6 Performance of ensemble of 5 3D CNNs on LMCI versus CN Classification Task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.05121145	3D ResNet-50	85.03 ± 7.94	87.49 ± 7.39	90.0 ± 6.24	88.58 ± 5.94	76.19 ± 15.06
0.16870546	3D ResNeXt-50	84.97 ± 1.99	88.18 ± 6.99	90.0 ± 8.16	88.45 ± 1.92	76.67 ± 14.32
0.18966876	3D SE-ResNet-50	87.13 ± 7.42	**91.44 ± 4.94**	88.33 ± 8.5	89.69 ± 6.04	**85.24 ± 9.09**
0.04099955	3D SE-ResNeXt-50	84.91 ± 5.36	86.9 ± 8.18	91.67 ± 5.27	88.82 ± 3.46	72.86 ± 18.29
0.54941478	3D SE-Net-154	88.19 ± 4.05	89.31 ± 4.68	**93.33 ± 6.24**	91.04 ± 3.06	79.05 ± 11.0
	Equally weighted average	88.25 ± 7.1	90.56 ± 6.86	**93.33 ± 6.24**	90.98 ± 5.27	82.38 ± 13.93
	Optimal weighted average	**90.41 ± 3.81**	91.13 ± 6.4	**93.33 ± 6.24**	**92.77 ± 2.69**	82.38 ± 13.93

Bold values indicate the optimal results obtained by the proposed models

Table 7 Performance of ensemble of 5 3D CNNs on EMCI versus AD Classification Task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.00761035	3D ResNet-50	77.05 ± 7.49	86.15 ± 2.68	73.92 ± 13.83	78.86 ± 8.33	**82.0 ± 4.0**
0.20785016	3D ResNeXt-50	73.88 ± 5.55	79.44 ± 13.56	85.33 ± 18.09	79.37 ± 4.46	56.0 ± 34.41
0.20546232	3D SE-ResNet-50	78.65 ± 5.56	82.15 ± 10.25	86.75 ± 14.02	82.71 ± 5.07	66.0 ± 22.45
0.33793241	3D SE-ResNeXt-50	79.42 ± 6.17	85.68 ± 7.77	80.33 ± 10.97	82.22 ± 5.69	78.0 ± 11.66
0.24114476	3D SE-Net-154	76.25 ± 3.94	83.72 ± 7.38	**77.83 ± 13.78**	79.41 ± 4.68	74.0 ± 18.55
	Equally weighted average	81.05 ± 9.62	86.56 ± 8.09	**77.83 ± 13.78**	83.05 ± 10.08	78.0 ± 13.27
	Optimal weighted average	**84.22 ± 8.84**	**87.59 ± 7.96**	**77.83 ± 13.78**	**86.7 ± 7.82**	80.0 ± 14.14

Bold values indicate the optimal results obtained by the proposed models

Table 8 Performance of ensemble of 5 3D CNNs on LMCI versus AD classification task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.42254655	3D ResNet-50	80.74 ± 4.32	83.43 ± 9.46	66.19 ± 12.27	72.64 ± 6.98	90.0 ± 6.32
0.10075757	3D ResNeXt-50	79.49 ± 3.12	81.67 ± 15.28	69.05 ± 14.6	72.21 ± 5.1	86.0 ± 12.0
0.05018952	3D SE-ResNet-50	77.13 ± 4.29	82.5 ± 15.0	60.95 ± 19.44	66.53 ± 8.85	88.0 ± 11.66
0.02901572	3D SE-ResNeXt-50	80.81 ± 4.12	83.62 ± 9.28	**66.67 ± 11.76**	73.0 ± 6.4	90.0 ± 6.32
0.39749065	3D SE-Net-154	81.91 ± 8.75	**88.1 ± 10.86**	65.71 ± 25.22	70.71 ± 22.13	92.0 ± 7.48
	Equally weighted average	79.41 ± 9.45	82.67 ± 18.31	65.71 ± 25.22	69.38 ± 15.72	92.0 ± 7.48
	Optimal weighted average	**84.26 ± 9.33**	86.67 ± 12.47	65.71 ± 25.22	**76.3 ± 17.23**	**94.0 ± 4.9**

Bold values indicate the optimal results obtained by the proposed models

Table 9 Performance of ensemble of 5 3D CNNs on EMCI versus LMCI Classification Task

Optimal weight	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
0.27390805	3D ResNet-50	75.28 ± 4.4	74.51 ± 4.07	**98.67 ± 2.67**	84.8 ± 2.48	21.43 ± 15.06
0.14375216	3D ResNeXt-50	75.24 ± 2.09	74.37 ± 2.55	**98.67 ± 2.67**	84.74 ± 1.26	20.95 ± 11.0
0.02845428	3D SE-ResNet-50	75.28 ± 3.34	74.36 ± 2.58	**98.67 ± 2.67**	84.77 ± 2.06	20.95 ± 12.27
0.14598411	3D SE-ResNeXt-50	75.24 ± 2.09	75.08 ± 3.37	97.33 ± 5.33	84.57 ± 0.98	23.33 ± 19.19
0.4079014	3D SE-Net-154	76.19 ± 4.25	**76.47 ± 4.6**	96.08 ± 5.29	84.92 ± 2.37	**30.48 ± 18.22**
	Equally weighted average	72.51 ± 2.55	71.75 ± 2.21	96.08 ± 5.29	83.53 ± 1.5	9.05 ± 7.44
	Optimal weighted average	**77.1 ± 4.83**	75.43 ± 4.15	96.08 ± 5.29	**85.93 ± 2.74**	24.76 ± 12.56

Bold values indicate the optimal results obtained by the proposed models

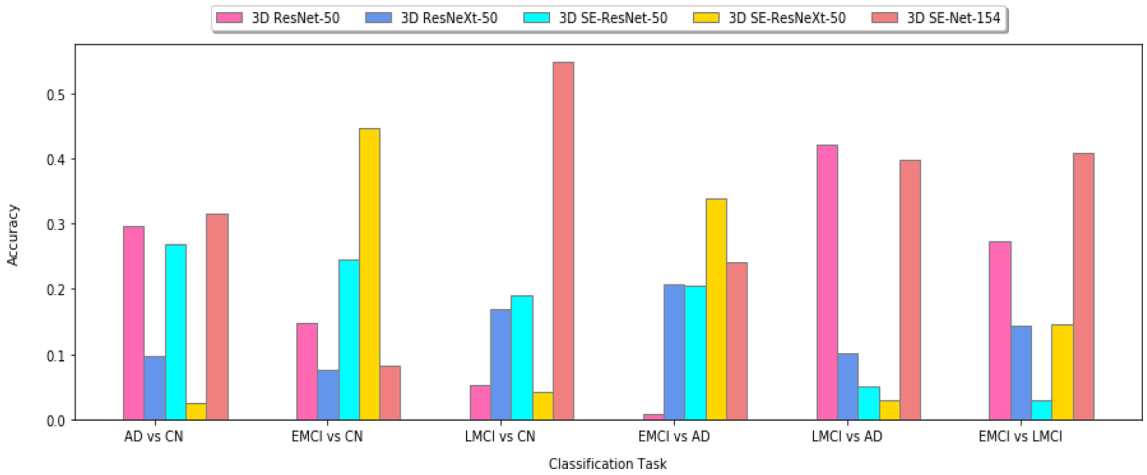


Fig. 10 Optimal weight assigned to each model for the six classification tasks

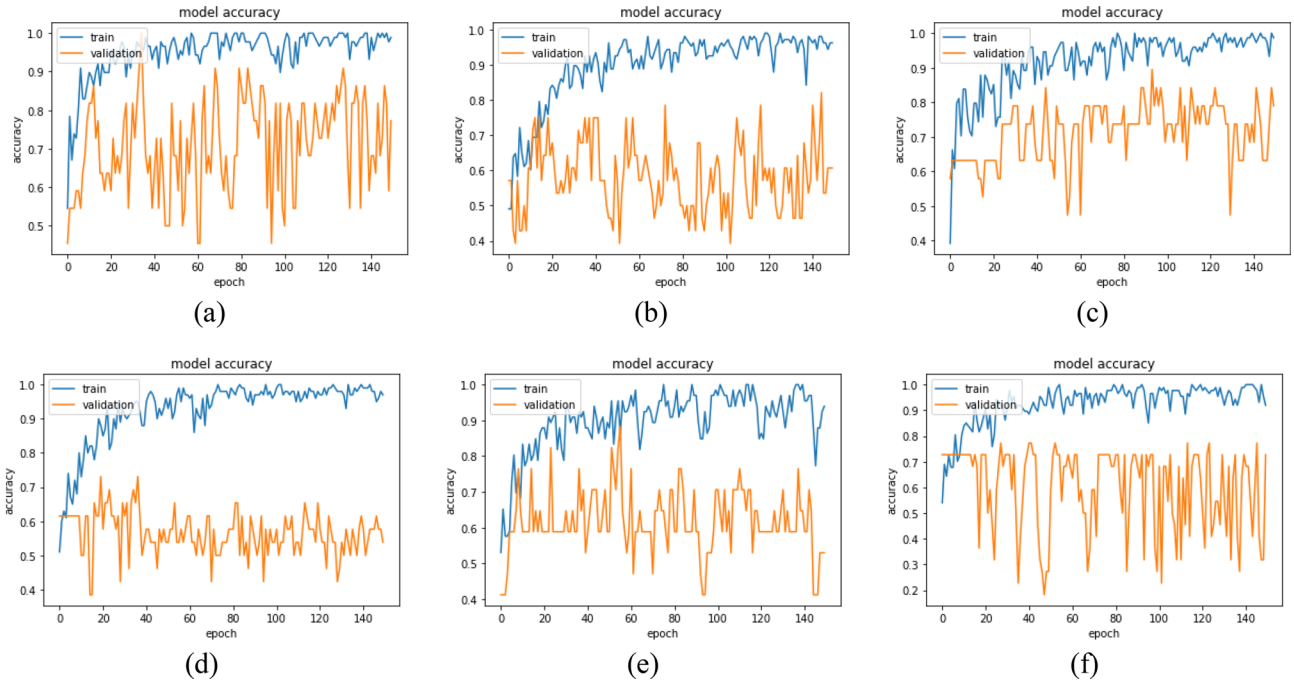


Fig. 11 Training and Validation accuracy for the best performing model on the initial fold for the following classification tasks—**a** AD versus CN (3D ResNet-50), **b** EMCI versus CN (3D SE-ResNet-50), **c** LMCI versus CN (3D SE-ResNeXt-50), **d** EMCI versus AD (3D SE-ResNeXt-50), **e** LMCI versus AD (3D SE-Net-154), and **f** EMCI versus LMCI (3D ResNeXt-50)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$F1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{9}$$

where TP, TN, FP, and FN denote the number of True Positive, True Negative, False Positive, and False Negative classification results, respectively.

Results and Discussion

As aforementioned, 50 subjects diagnosed with AD, 76 subjects with EMCI, 33 subjects with LMCI, and 60 CN subjects are considered. With these four classes, there are six binary classification tasks. Given that this is a small dataset, fivefold cross validation was performed to better evaluate model performance, and data augmentation was applied on the fly for better generalization as described in section III B.

The experiments were performed using Keras v2.4.0 framework with a TensorFlow v2.4.2 backend using a cloud-based NVIDIA TESLA P100 GPU (16 GB). The same training parameters are utilized for all models of the ensemble. The Adam optimization model was applied with the following parameters—learning rate of 0.001, 0.9 and 0.999 as

Table 10 Comparison with state-of-the-art for AD versus CN classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[26]	50 AD + 62 CN	VoxResNet	80.00	–	–	–	
[32]	137 AD + 162 CN	2D CNN + ensemble learning	84.00	–	–	–	–
[27]	592 AD + 960 CN	Spectral graph CNN	85.8	82.31	83.5	82.9	87.5
[30]	198 AD + 230 CN	Convolutional Autoencoder based on Google InceptionV2	86.60	–	88.55	–	84.54
[25]	336 AD + 330 CN	ROI based 3D CNN with autoencoder pre-training	88.00	–	–	–	–
[37]	358 AD + 429 CN	Landmark based multi-instance 3D CNN	91.09	91.49	88.05	89.74	93.50
[35]	49 CN + 51 AD	2D CNN + attention mechanism	92	97	85	91	–
[38]	900 AD + 900 CN	LSTM	92.20	–	–	–	–
[36]	198 AD + 229 CN	3D CNN + 3D CLSTM	94.19	–	93.75	–	94.57
[29]	132 AD + 132 CN	Pre-trained 3D ResNet-18	96.88	–	**100**	–	93.75
Our model	50 AD + 60 CN	Optimal weighted average ensemble of 5 3D CNNs	**97.27**	**98.33**	96.67	**97.46**	**98.0**

Bold values indicate the optimal results obtained by the proposed models

Table 11 Comparison with state-of-the-art for EMCI versus CN classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[27]	899 EMCI + 960 CN	Spectral graph CNN	51.8	50.15	55.3	52.6	48.6
[26]	77 LMCI + 61 CN	VoxResNet	56	–	–	–	–
Our model	76 EMCI + 60 CN	Optimal weighted average ensemble of 5 3D CNNs	**82.33**	**81.18**	**71.67**	**80.24**	**84.08**

Bold values indicate the optimal results obtained by the proposed models

Table 12 Comparison with state-of-the-art for LMCI versus CN classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[27]	638 LMCI + 960 CN	Spectral graph CNN	69.3	62.85	65.6	64.2	72.0
[26]	43 LMCI + 61 CN	VoxResNet	61	–	–	–	–
Our model	33 LMCI + 60 CN	Optimal weighted average ensemble of 5 3D CNNs	**90.41**	**91.13**	**93.33**	**92.77**	**82.38**

Bold values indicate the optimal results obtained by the proposed models

Table 13 Comparison with state-of-the-art for EMCI versus AD classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[27]	899 EMCI + 592 AD	Spectral graph CNN	79.2	78.88	70.4	74.4	85.8
[26]	77 EMCI + 50 AD	VoxResNet	63	–	–	–	–
Our model	76 EMCI + 50 AD	Optimal weighted average ensemble of 5 3D CNNs	**84.22**	**87.59**	**77.83**	**86.7**	**80.0**

Bold values indicate the optimal results obtained by the proposed models

Table 14 Comparison with state-of-the-art for LMCI versus AD classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[27]	638 LMCI + 592 AD	Spectral graph CNN	65.2	65.67	62.6	64.1	68.0
[26]	43 LMCI + 50 AD	VoxResNet	59	–	–	–	–
Our model	33 LMCI + 50 AD	Optimal weighted average ensemble of 5 3D CNNs	**84.26**	**86.67**	**65.71**	**76.3**	**94.0**

Bold values indicate the optimal results obtained by the proposed models

Table 15 Comparison with state-of-the-art for EMCI versus LMCI classification

Ref.	Subjects	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)
[27]	899 EMCI + 638 LMCI	Spectral graph CNN	60.9	54.53	52.5	53.5	**67.8**
[26]	77 LMCI + 43 EMCI	VoxResNet	52	–	–	–	–
Our model	76 EMCI + 33 LMCI	Optimal weighted average ensemble of 5 3D CNNs	**77.1**	**75.43**	**96.08**	**85.93**	24.76

Bold values indicate the optimal results obtained by the proposed models

the exponential decay rates for the first and second moment estimates, and a batch size of 8. The learning rate was scheduled using Exponential Decay with the decay steps set to 100,000, decay rate set to 0.96, and staircase set to 'True'. We trained for 150 epochs.

Effects of 3D Transfer Learning

For investigating the effects of 3D transfer learning on AD classification, we fixed the classification task to AD versus CN and trained all 5 models of the ensemble from scratch as well as with ImageNet pre-training as described in III C 2). The results are presented in Table 3 and Fig. 9. We have not presented the results for other classification tasks as the trends are similar for other classification tasks also.

From these experiments it is observed that the model performance could most definitely be improved by using transfer learning. All 5 3D CNN models had successfully exploited the information learned by their 2D equivalents on the ImageNet dataset and used that information to improve their classification ability on the AD versus CN task. On average, transfer learning had increased the model accuracy by 3.43%. Moving forward, we applied transfer learning for all experiments.

Effects of Weighted Average Ensemble

After training the 5 pre-trained 3D CNN models on each of the six binary classification tasks one by one, we found the optimum weights to be assigned to each model for each task by minimizing Eq. (4). Then an equally weighted average and the optimal weighted average of the models' predictions were computed to test the effectiveness of the optimal fusion. The performance of the individual models, equally weighted average, and the optimal weighted average along with the optimal weight assigned to each model are shown in Table 4, 5, 6, 7, 8 and 9 and Fig. 10.

Across all binary classification problems, the optimal weighted average outperforms the equally weighted average as well as the individual models showcasing the effectiveness of minimizing loss function defined in (3) through the SQP algorithm. It shows that by evaluating all the models individually first, and placing our trust in those models we know perform better than others, we can achieve substantially better classification performance. This is evident from Tables 4, 5, 6, 7, 8, and 9 where it is observed that generally higher weights are given to better performing models. Averaging across all the tasks, the optimal weighted average exceeds the single best model by 3.15% and the equally

weighted average by 3.35% in terms of accuracy. Figure 11, shows the accuracy curves for the best performing 3D CNN model on the first fold for all six binary classification tasks. After data augmentation and applying multifold cross validation, we could also observe the sign of overfitting. For most of the six binary classification tasks, we observe a steady peak in the model performance on the validation set. The probable reason behind the same is less amount of available training data and inherent variance associated among the dataset instances. We could also notice sharp performance drop for some of the instances due to again the same reason that the augmentation could not sketch the generalization aspects of the dataset for the respective samples.

In several cases, the precision, recall, and specificity metrics of a few individual models outperform those of the optimal weighted ensemble as weights of the ensemble members have been optimized to maximize accuracy and not any other metric. Another interesting remark is that for the EMCI versus LMCI classification task, recall is very high and specificity is very low for all the models. This means the models are able to correctly classify most of the subjects suffering from LMCI (positive class), but for all the subjects suffering from EMCI (negative class), the number of correct predictions is low, conveying that all the models are classifying most of the subjects as LMCI.

We conclude that while the proposed method is successful in discriminating between normal cohorts and MCI subjects even though the brain morphology between subjects of these classes is quite similar, it is not quite able to reach the same level of success in distinguishing between the early and late stages of MCI, suggesting that the differences in brain structures between subjects suffering from EMCI and LMCI are even subtler than the differences between the brain structures of cognitively normal subjects and MCI subjects.

Comparison with State-of-the-Art

The results of our model are compared with state-of-the-art deep learning models that trained and reported performance using MRI data from the ADNI dataset [25–27, 29, 30, 32, 35–38]. From Table 10, 11, 12, 13, 14, and 15, it can be seen that our model outperforms aforementioned methods in terms of accuracy despite employing a very small subset of ADNI. The superior performance of our models can especially be seen in the binary classification tasks involving patients suffering from early and late MCI. For example in the EMCI versus CN classification task, our model improves upon [27] by a tremendous 30.53% despite the dataset being 92.68% smaller than [27]. The only significant drawback to our method is its low specificity in the EMCI versus LMCI task. But it should be taken into account that in practice, there is not much significant value in distinguishing EMCI and LMCI patients. It is, however, extremely important and useful to differentiate normal patients from those suffering from MCI and AD, which our method has proven to excel as compared to state-of-the-arts.

Conclusions

In this paper, we proposed a weighted average ensemble of 3D CNNs for the classification of structural MRI images. The ensemble consisted of the 3D variants of the ResNet-50, ResNeXt-50, SE-ResNet-50, SE-ResNeXt-50, and SE-Net-154, each initialized with the convolutional filters of their 2D equivalents pre-trained on ImageNet, and combined their predictions using an optimal weighted average. We found the optimal weights of the members of the ensemble by devising a loss function that computes the inaccuracy of the ensemble, and minimizing it using sequential quadratic programming. The proposed method achieved outperformed state-of-the-art on six AD binary classification tasks, displaying the effectiveness of transfer learning and an optimal weighted ensemble.

Despite the promising performance, the proposed model is not without limitations. Taking the whole brain as an input results in high feature dimensionality and makes training computationally expensive. In future, we would like to experiment with region-of-interest and patch-based approaches toward accomplishing higher accuracy. We also plan to test the robustness of our model with subjects classified as stable and progressive MCI, as it is important to determine whether someone with MCI may develop AD.

Funding No funding was received for conducting this study.

Data Availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ebrahimighahnavieh MA, Luo S, Chiong R. Deep learning to detect Alzheimer's disease from neuroimaging: a systematic literature review. Comput Methods Programs Biomed. 2020;187:105242.
2. Aisen PS, Petersen RC, Donohue MC, Gamst A, Raman R, Thomas RG, Walter S, Trojanowski JQ, Shaw LM, Beckett LA, Jack CR Jr. Clinical core of the Alzheimer's disease

neuroimaging initiative: progress and plans. Alzheimers Dement. 2010;6(3):239–46.
3. Prince M, Bryce R, Ferri C. World Alzheimer report 2011: the benefits of early diagnosis and intervention; 2011. https://www.alzint.org/u/WorldAlzheimerReport2011.pdf.
4. Ellis KA, Bush AI, Darby D, De Fazio D, Foster J, Hudson P, Lautenschlager NT, Lenzo N, Martins RN, Maruff P, Masters C. The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer's disease. Int Psychogeriatr. 2009;21(4):672–87.
5. Marcus DS, Wang TH, Parker J, Csernansky JG, Morris JC, Buckner RL. Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. J Cogn Neurosci. 2007;19(9):1498–507.
6. Bron EE, Smits M, Van Der Flier WM, Vrenken H, Barkhof F, Scheltens P, Papma JM, Steketee RM, Orellana CM, Meijboom R, Pinto M. Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CAD-Dementia challenge. Neuroimage. 2015;111:562–79.
7. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition; 2009. p. 248–55.
8. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, Suleyman M. The kinetics human action video dataset. Preprint arXiv:1705.06950; 2017.
9. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: a large video database for human motion recognition. In: International conference on computer vision; 2011. p. 2556–63.
10. Soomro K, Zamir AR, Shah M. UCF101: a dataset of 101 human actions classes from videos in the wild. Preprint arXiv:1212.0402; 2012.
11. Hara K, Kataoka H, Satoh Y. Learning spatio-temporal features with 3d residual networks for action recognition. In: IEEE international conference on computer vision workshops; 2017. p. 3154–60.
12. Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 6299–308.
13. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
14. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: IEEE conference on computer vision and pattern recognition; 2017. p. 1492–500.
15. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: IEEE conference on computer vision and pattern recognition; 2018. p. 7132–41.
16. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Preprint arXiv:1409.1556; 2014.
17. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition; 2015. p. 1–9.
18. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inform Process syst. 2012;25:1.
19. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition; 2017. p. 4700–08.
20. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence; 2017.
21. Brownlee J. Ensemble learning methods for deep learning neural networks. Machine Learning Mastery; 2018. https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/. Accessed: 03-Jan-2022.
22. Yan WQ. Computational methods for deep learning. London: Springer; 2021.
23. Seni G, Elder JF. Ensemble methods in data mining: improving accuracy through combining predictions. Synth Lect Data Min Knowl Disc. 2010;2(1):1–26.
24. Boggs PT, Tolle JW. Sequential quadratic programming. Acta Numer. 1995;4:1–51.
25. Wen J, Thibeau-Sutre E, Diaz-Melo M, Samper-González J, Routier A, Bottani S, Dormont D, Durrleman S, Burgos N, Colliot O. Alzheimer's Disease Neuroimaging Initiative. Convolutional neural networks for classification of Alzheimer's disease: overview and reproducible evaluation. Med image Anal. 2020;63:101694.
26. Korolev S, Safiullin A, Belyaev M, Dodonova Y. Residual and plain convolutional neural networks for 3D brain MRI classification. In: 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017). IEEE; Apr-2017.
27. Korolev S, Safiullin A, Belyaev M, Dodonova Y. Residual and plain convolutional neural networks for 3D brain MRI classification. In: IEEE 14th international symposium on biomedical imaging (ISBI 2017); 2017. p. 835–8.
28. Bae JB, Lee S, Jung W, Park S, Kim W, Oh H, Han JW, Kim GE, Kim JS, Kim JH, Kim KW. Identification of Alzheimer's disease using a convolutional neural network model based on T1-weighted magnetic resonance imaging. Sci Rep. 2020;10(1):1.
29. Ebrahimi A, Luo S, Chiong R. Introducing transfer learning to 3D ResNet-18 for Alzheimer's disease detection on MRI images. In: 35th international conference on image and vision computing New Zealand (IVCNZ); 2020. p. 1–6.
30. Oh K, Chung YC, Kim KW, Kim WS, Oh IS. Classification and visualization of Alzheimer's disease using volumetric convolutional neural network and transfer learning. Sci Rep. 2019;9(1):1–6.
31. Folego G, Weiler M, Casseb RF, Pires R, Rocha A. Alzheimer's disease detection through whole-brain 3D-CNN MRI. Front Bioeng Biotechnol. 2020;8:534592.
32. Pan D, Zeng A, Jia L, Huang Y, Frizzell T, Song X. Early detection of Alzheimer's disease using magnetic resonance imaging: a novel approach combining convolutional neural networks and ensemble learning. Front Neurosci. 2020;14:259.
33. Islam J, Zhang Y. Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain Inform. 2018;5(2):1–4.
34. Choi JY, Lee B. Combining of multiple deep networks via ensemble generalization loss, based on MRI images, for Alzheimer's disease classification. IEEE Signal Process Lett. 2020;27:206–10.
35. Xing X, Liang G, Blanton H, Rafique MU, Wang C, Lin AL, Jacobs N. Dynamic image for 3D MRI image Alzheimer's disease classification. In: European conference on computer vision; 2020. p. 355–64.
36. Xia Z, Yue G, Xu Y, Feng C, Yang M, Wang T, Lei B. A novel end-to-end hybrid network for Alzheimer's disease detection using 3D CNN and 3D CLSTM. In: IEEE 17th international symposium on biomedical imaging (ISBI); 2020. p. 1–4.
37. Liu M, Zhang J, Adeli E, Shen D. Landmark-based deep multi-instance learning for brain disease diagnosis. Med Image Anal. 2018;43:157–68.
38. Hong X, Lin R, Yang C, Zeng N, Cai C, Gou J, Yang J. Predicting Alzheimer's disease using LSTM. IEEE Access. 2019;7:80893–901.
39. Islam J, Zhang Y. A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data. In: International conference on brain informatics; 2017. p. 213–22.

40. Hon M, Khan NM. Towards Alzheimer's disease classification through transfer learning. In: IEEE international conference on bioinformatics and biomedicine (BIBM); 2017. p. 1166–9.
41. Wang S, Shen Y, Chen W, Xiao T, Hu J. Automatic recognition of mild cognitive impairment from MRI images using expedited convolutional neural networks. In: International conference on artificial neural networks; 2017. p. 373–80.
42. Farooq A, Anwar S, Awais M, Rehman S. A deep CNN based multi-class classification of Alzheimer's disease using MRI. In: IEEE international conference on imaging systems and techniques (IST); 2017. p. 1–6.
43. Pérez-García F, Sparks R, Ourselin S. TorchIO: a Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. Comput Methods Programs Biomed. 2021;208:106236.
44. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, Van Der Walt SJ. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat Methods. 2020;17(3):261–72.
45. Dharwada S, Tembhurne J, Diwan T. Multi-channel deep model for classification of Alzheimer's disease using transfer learning. In: International conference on distributed computing and internet technology; 2022. p. 245–59.