International Journal of Machine Learning and Cybernetics (2022) 13:1997–2011
https://doi.org/10.1007/s13042-021-01501-7

ORIGINAL ARTICLE

3D transfer learning network for classification of Alzheimer's disease with MRI

Haifeng Wu[1] · Jinling Luo[1] · Xiaoling Lu[1] · Yu Zeng[1]

Received: 29 October 2020 / Accepted: 22 December 2021 / Published online: 7 January 2022

Abstract

Background As a kind of dementia, Alzheimer's disease (AD) cannot be cured once diagnosed. Hence, it is very important to diagnose early and delay the deterioration of the disease through drugs.
Objective To reduce the computational complexity of conventional 3D convolutional networks, this paper uses machine learning as an auxiliary diagnosis of AD, and proposes three-dimensional (3D) transfer network which is based on two-dimensional (2D) transfer network to classify AD and normal groups with magnetic resonance imaging (MRI).
Method First, the method uses a 2D transfer Mobilenet to extract features from 2D slices of MRI, and further perform dimension reduction for the extracted features. Then, all of the 2D slice features of one subject are merged to classify.
Results The experiment in this paper uses an open access Alzheimer's disease database to evaluate the method. The experiment result show that the classification accuracy of the proposed 3D network is better than that of the existing 2D transfer network, increased by about 10 percentage points and the classification time is only about 1/4 of the existing one.
Conclusion The proposed method is to realize the classification of 3D MRI data through an existing 2D transfer network, and it not only reduces the complexity of conventional 3D networks, but also improves the classification accuracy. Because of the shared weight of the transfer network, besides, the classification time is reduced.

Keywords AD · MRI · MobileNet · Transfer learning · Classification

1 Introduction

The symptoms of Alzheimer's disease (AD) are impaired cognition, weakened memory, and decreased ability to speak, understand, and learn. In severe cases, life can not take care of itself. As one of the most common dementia, AD is a disease that eventually causes death. At present, its cause has not been fully understood and no treatment has been found. Once the disease is diagnosed, it can only be delayed by drugs and cannot be reversed. Some mild AD will deteriorate rapidly, but some will remain stable (Thies and Bleiler 2013). This also provides an opportunity for its treatment. Therefore, it is important to diagnose AD as soon as possible.

Cognitive function tests are a commonly used manual diagnosis method, such as mini-mental state examination (Arevalo-Rodriguez et al. 2015), Hasegawa dementia scale (Kim et al. 2005) and neuropsychological test battery (Harrison et al. 2007), etc. Although the tests are highly targeted and comprehensive in evaluation, they are greatly affected by the patient's own, such as cultural level, language expression and understanding. Moreover, early AD symptoms are similar to many mental diseases and thus it is easy to cause confusion and a high rate of misdiagnosis. Therefore, the cognitive tests are mostly used for preliminary screening of AD. Using medical imaging technology to manually diagnose AD is also a commonly used diagnostic method (Johnson et al. 2012), such as computer tomography (CT), positron emission tomography (PET), magnetic resonance imaging (MRI) and other technologies. A doctor depends on his professional knowledge and experience to analyze

✉ Haifeng Wu
whf5469@gmail.com

Jinling Luo
3195734466@qq.com

Xiaoling Lu
291116385@qq.com

Yu Zeng
yv.zeng@gmail.com

[1] Department of Electrical and Information Engineering, Yunnan Minzu University, Kunming 650500, China

the patient's brain image and make a diagnosis based on the patient's symptoms. However, this diagnostic method requires a doctor to carefully observe the patient's brain imaging and takes too much time. Thus, the method is not high-efficiency.

In recent years, with the rapid development of machine learning technology, people have found that machine learning can be used as a rapid auxiliary diagnostic method. Machine learning can diagnose AD through biomarkers, such as extracting cerebrospinal fluid by puncture (Blennow et al. 2003). However, the method is invasive diagnosis and will inevitably cause some harm to the patient's body. CT imaging can also be used for machine learning diagnosis (Gao et al. 2017). It will obtain patient axis, coronal and sagittal images and thus provides complete three-dimensional information. However, the radiation of CT can also cause damage to human tissue. MRI imaging has the advantages of no radiation, high resolution and low interference, and its use in AD patients can effectively detect the atrophy of some brain structures such as hippocampus, amygdala, entorhinal cortex, and cingulate gyrus (Aël Chetelat et al. 2003; Frisoni et al. 2010). The structures are key features of AD. Thus, MRI has been widely used in the diagnosis of AD for machine learning. Support vector machine (SVM) (Orru et al. 2012; Dessouky et al. 2013) is more widely used in AD diagnosis of machine learning. It can be used not only for linear data, but also for nonlinear data. However, the accuracy of traditional MRI machine learning classification like SVM depends on the extracted features, and most of the features also need to be manually extracted. Sometimes feature extraction is more difficult. If the extracted features themselves are not correct, the classification accuracy will not be very high. Therefore, for image classification, such as AD images whose features are still controversial, there are often some uncertainties.

Compared with the traditional machine learning, MRI diagnosis of deep learning (Farooq et al. 2017; Basaia et al. 2019) often has the ability of self-learning features extraction. For example, a convolution neural network (CNN) change original MRI data into low-level features through a nonlinear model, and then forms high-level features through multiple fully connected layers. Thus, the classification objects will have more specific and effective expression of the features. Since MRI is a 3D image, some 3D CNN (Feng et al. 2019; Yagis et al. 2020) have to be built to classify MRI images. Due to the complex structure of 3D CNN, they has many tedious parameter tuning and thus a large amount of computation and high complexity. Besides, CNN networks generally require a large amount of training data for higher classification accuracy, which also produce a huge amount of calculation. Especially for AD MRI images in a medical field, it is difficult to ensure a high classification accuracy for deep learning due to limited open-access image data. Transfer learning (Glozman et al. 2016; Hon et al. 2017) is to transfer a pre-trained deep network to a target problem, where only the top layer of the pre-trained network needs to be changed. Therefore, the tuned parameters are greatly reduced and time is saved. More importantly, transfer learning only needs a small amount of data to get a good classification performance. Because of the advantages of transfer learning, some 3D transfer CNNs like (Hosseini-Asl et al. 2016) is used in AD diagnosis of MRI. Unfortunately, the 3D networks do not have network weight data available for download, which means the network needs to be trained from scratch. AlexNet (Glozman et al. 2016) and VGG16 (Hon et al. 2017; Mehmood et al. 2021; Kumar et al. 2021; Jain et al. 2019) as transferable networks, they are used to extract features from gray matter tissue extracted from brain images or brain sMRI slices, etc., and they achieve better results in the classification of AD, CN, and MCI. The classification accuracy rate finally realizes the early diagnosis of Alzheimer's disease, which overcomes the problem that deep learning algorithms require a large number of labeled data sets for training. It also shows the effectiveness of the AlexNet and VGG16 network initialization weights obtained by training on the ImageNet data set in AD diagnosis, AlexNet and VGG16 models can still extract useful features from sMRI images. High-dimensional deep neural network models such as VGG19 (Abed et al. 2020), InceptionV3 (Abed et al. 2020; Tufail et al. 2020), ResNet50 (Abed et al. 2020) and Xception (Tufail et al. 2020) are transferred to the classification of patients with AD, CN and MCI. The studies are able to obtain better classification accuracy in a short time, and show that transfer learning methods are even better than non-transfer learning methods. The DenseNet (Ashraf et al. 2021) network and 12 other different types of pre-trained CNN models are transferred to classify Alzheimer's disease. The DenseNet network shows better performance in accuracy. Currently, the weight data of some 2D transfer networks are open access. To adapt to 3D data, however, MRI has to be sliced into 2D images as the input of the 2D transfer networks, such as AlexNet (Glozman et al. 2016) and Vgg16 (Hon et al. 2017) applied to AD MRI diagnosis. Although the slicing method makes the dimension of 3D MRI and that of 2D network matched, slices will inevitably bring information loss and affect the accuracy of classification.

For the above problems, this paper proposes a novel 3D transfer network utilizing a 2D transfer CNN. It can realize the classification for an AD and a normal control (NC) groups of MRI. We do not consider the type of Alzheimer's disease. The utilized 2D transfer CNN chooses Mobilenet (Howard et al. 2017) which is a lightweight network owing to its separable convolution. The advantage of the proposed 3D transfer CNN is its lower computational complexity. The reason is not only that its weight data can be open downloaded, but also that it adopts the structure of a bottleneck layer, a dimension-reduction and a classification layer, where the weights are shared for the MRI slices. The third one merges the features of the slices into one subject. Weight sharing means that the weights of all slices is shared. Even if the slices increases, the number of weights will not increase, thus reducing the complexity of the model and reducing training and classification time. In this way, as many slices as possible can ensure less information loss and higher classification accuracy. In experiments, we use an open access series of imaging studies (OASIS) database (Marcus et al. 2007) to evaluate the proposed method. The experiment results show that compared with the existing 2D transfer network, although the proposed 3D transfer network increases the input image slices, the training and classifying time does not increase and the classification accuracy is also improved.

2 Related transfer network

The essence of AD diagnosis with MRI is to classify different images, and the deep learning for the image classification usually requires a huge data set for training to obtain high classification accuracy. However, the symptoms of AD are not single, and their forms at different stages are also different. This leads to differences in MRI images. Therefore, in order to improve the accuracy of AD diagnosis, it is necessary to use large data sets of different symptoms and different stages of AD as much as possible. At present, some of the larger AD databases in the world, such as OASIS (Marcus et al. 2007), Alzheimer's Disease Neuroimaging Initiative (ADNI) (Petersen et al. 2010) and CAD Dementia (Bron et al. 2015), are available. Nevertheless, the type and amount of the data are still insufficient to support the high amount of data for deep learning training.

Transfer learning is a better image classification for small data sets. Even when a large training data set is not available, it may also obtain a higher accuracy (Chollet 2015). Therefore, this paper focuses more on the AD diagnosis of transfer learning. In addition, compared with PET and CT technology, because of MRI with lower radiation and without contrast agent, it is less harmful to humans. For the reason, this paper mainly uses MRI to study AD classification. Earlier transfer learning used in MRI AD diagnosis is a 3D CNN method (Hosseini-Asl et al. 2016), where convolutional layers are at lower layers of the network and fully connected layers at higher layers. The 3D CNN can be trained and verified on the CAD Dementia dataset, and also can be transferred to the ADNI dataset (Petersen et al. 2010). The method would obtain a high classification accuracy, but 3D convolution is used and thus the number of weights is huge. Its pre-training weights are not available for download although the network files can be open-access. This results in excessive training time and become a factor restricting its further application.

For too many weights and long training time of the 3D transfer learning, several 2D transfer networks are proposed for AD diagnosis. A 2D network uses AlexNet as a pre-training network (Glozman et al. 2016), which slices the MRI image by position, selects several slices in the middle position as an input, and then utilizes a top-layer network to perform the final classification. Since the method slices the 3D MRI image into 2D images, it solves the dimensional match of the 3D image to the 2D network. Unfortunately, similar to the 3D CNN transfer network above, the method does not provide the pre-trained network weights downloaded. Another 2D transfer network uses Vgg16 as a pre-training network (Hon et al. 2017). The 2D network still need to slice the MRI image, but it selects several slices by the largest information entropy for classification, instead of slices by position. The biggest advantage of the method is that the weights of the pre-training network can be publicly downloaded, which can reduce the time on pre-training network. In its cross-validation, however, it randomly allocates the slices from subjects to a train and a test set. That is, some slices of the train and the test set may belong to the same subject. Therefore, although this method has a high classification accuracy, it is difficult in applications.

3 Problem for MRI slices

Due to the limited open-access AD MRI data, this paper will use a transfer CNN for a small data set to complete AD classification. Furthermore, since many 2D transfer network's weights could be available for download, this paper will utilize the 2D transfer networks to complete the final 3D MRI classification.

As a high-definition imaging technology, MRI is a 3D image showing a brain structure, i.e. a 3D image with $N_X \times N_Y \times N_Z$, where N_X,N_Y and N_Z represent the three dimensions of the brain, respectively. To let the 3D image used as input for a 2D network, MRI needs to be sliced into N_1 two-dimensional image of $N_2 \times N_3$, where N_i, $i = 1$, 2 or 3 can be any one of N_X, N_Y and N_Z, so that coronal, sagittal, or axial slices can be obtained. Then, the N_1 2D slices are input to the 2D transfer network. In theory, a larger N_1 will be better because the information loss will be less. However, the weights in a deep network will become large if the images are too many. Taking a CNN network as an example, it has N_C convolutional layers in total, where each convolutional layer is composed of L_1, L_2, … L_N feature matrices, the size of the feature matrices is $F_1 \times F_1$, $F_2 \times F_2$, …$F_N \times F_N$, and the size of the used convolution kernel is $M_1 \times M_1$, $M_2 \times M_2$,

$\ldots M_N \times M_N$, respectively. If the network's stride is 1, then the total weight is

$$W_{CNN} = N_1 \sum_{i=1}^{N_c} L_i(M_i M_i + 1) \tag{1}$$

where 1 is the value of a bias. It can be seen from (1) that the number of weights of the network is closely related to the number N_1 of MRI image slices, and larger N_1 will produce larger W_{CNN}.

To reduce the number of slices, some criteria can be adopted. One is to sort by position. The slices closer to the middle of the brain will be reserved, other slices discarded (Glozman et al. 2016). Another one is to sort by information entropy, and the slices with greater entropy will be reserved (Hon et al. 2017). However, no matter what criterion is adopted, a reduction of slices will necessarily lead to loss of information. On the other hand, an increase of slices will make the network complicated and the training time extended, as shown in Fig. 1. Therefore, the key issue of this study is to find an optimal solution between the loss of MRI information and the complexity of the transmission network in order to obtain the best classification accuracy and training time.

4 Proposed 3D transfer network

4.1 Overview of proposed 3D transfer network

In this paper, we use a transfer CNN network with MRI to classify AD. The overview of the network is shown in Fig. 2. First, slice the MRI of a subject as the input to a pre-trained transfer network and get the bottleneck features of the slices. For an example of the training set, the feature dimension is (5120, 5, 5, 1024). And then, input the bottleneck features to a dimension-reduction layer based on an auto-encoder (AE) network and train the layer. The dimensionality becomes (5120, 128). Next, we merge the AE features of the slices into one subject, and the dimensionality is (160, 32, 128). Input the merged features to a classification layer to obtain a classification result, and let the classification layer network trained. After the training is completed, the entire network training is completed, the whole dimensionality reduction is completed. In this method, the 2D transfer bottleneck network is used to extract the features of 2D slices, where the information loss from 3 to 2D can be reduced as long as enough slices are generated for the 3D MRI. Throughout the training, the weights of the bottleneck layer, the dimension-reduction layer are shared for all slices. Therefore, even if the number of slices is increased, the number of weights will not be increased. In addition, although the AE features extracted from bottleneck features are merged, the AE network further reduces the dimension of the bottleneck features. Therefore, the dimension of the features will become smaller, and the complexity of the classification layer will also be reduced.

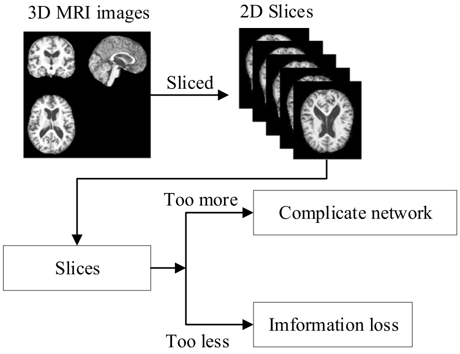


Fig. 1 Slice quantity for AD classification with MRI

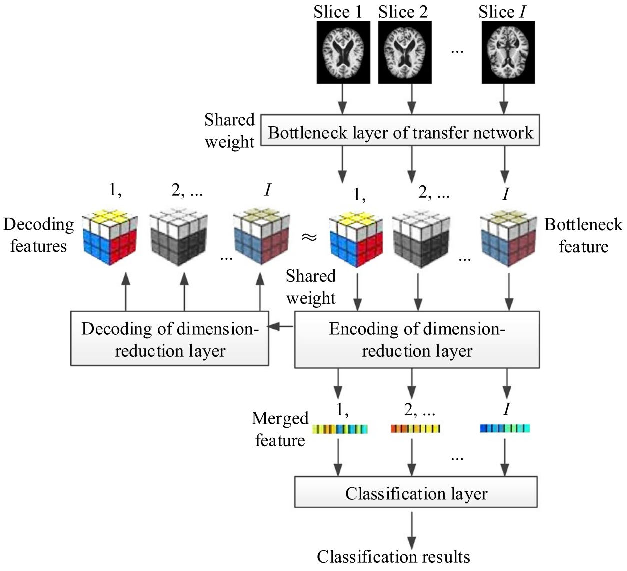


Fig. 2 Overview of this 3D transfer network

4.2 3D feature extraction

An MRI signal is generally a 3D data, and cannot be directly used as an input to a 2D image classifier. To realize the feature extraction, we use a 2D transfer network to extract the features of the MRI slices and then merge them. Figure 3 give the flow charts of the feature extraction and the network training. The training has two steps, pre-training and target training. The pre-training step usually does not need to be completed on a local client. Even if the number of the weights of the bottleneck layer are very large, it can be completed by others in advance. Thus, the local client spends no time on pre-training the bottleneck layer. In order to better

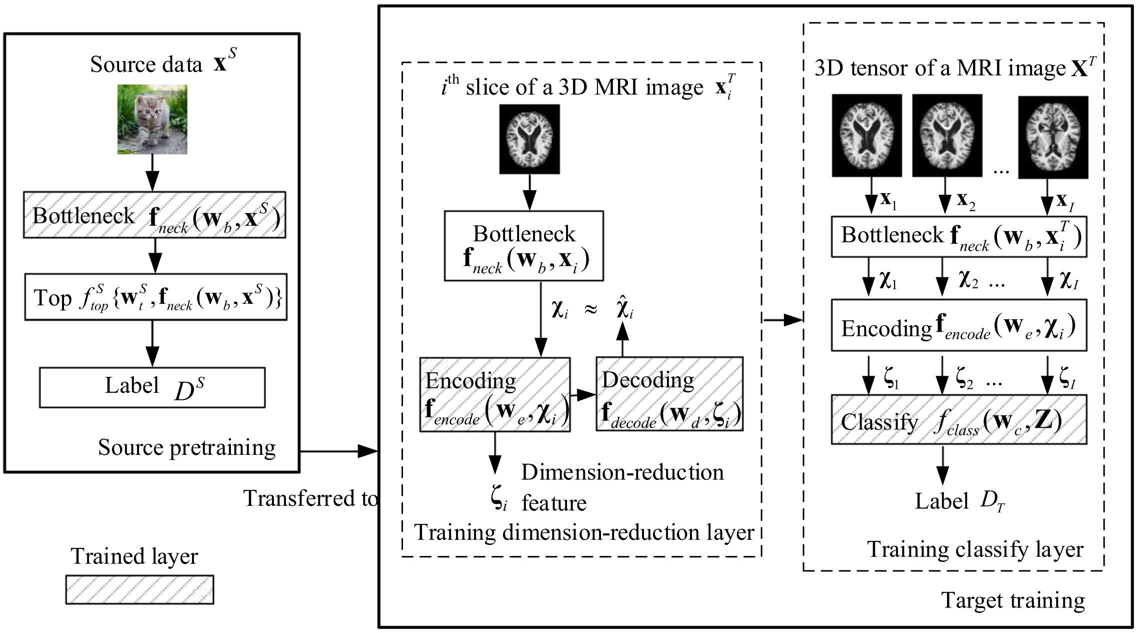

Source data $\mathbf{x}^S$
Bottleneck $\mathbf{f}_{neck}(\mathbf{w}_b,\mathbf{x}^S)$
Top $f_{top}^S\{\mathbf{w}_t^S,\mathbf{f}_{neck}(\mathbf{w}_b,\mathbf{x}^S)\}$
Label D^S
Source pretraining
Transferred to
Trained layer
i^{th} slice of a 3D MRI image $\mathbf{x}_i^T$
Bottleneck $\mathbf{f}_{neck}(\mathbf{w}_b,\mathbf{x}_i)$
$\chi_i \approx \hat{\chi}_i$
Encoding $\mathbf{f}_{encode}(\mathbf{w}_e,\chi_i)$
Decoding $\mathbf{f}_{decode}(\mathbf{w}_d,\zeta_i)$
Dimension-reduction
ζ_i feature
Training dimension-reduction layer
3D tensor of a MRI image $\mathbf{X}^T$
...
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_I$
Bottleneck $\mathbf{f}_{neck}(\mathbf{w}_b,\mathbf{x}_i^T)$
χ_1 χ_2 ... χ_I
Encoding $\mathbf{f}_{encode}(\mathbf{w}_e,\chi_i)$
ζ_1 ζ_2 ... ζ_I
Classify $f_{class}(\mathbf{w}_c,\mathbf{Z})$
Label D_T
Training classify layer
Target training

Fig. 3 Flow charts of training and feature extraction for this transfer learning

extract bottleneck features, besides, the pre-training is often performed in a very huge source data set, such as ImageNet dataset (Deng et al. 2009) which has a large number of pictures with many types, complete labels and high resolution. Therefore, the transfer network can deal with a small data set better than a CNN network without transfer, especially when large amounts of public AD MRI data is still not available.

The AD target training in Fig. 3 also has two sub-steps, the dimension-reduction layer training and classification layer training. The dimension-reduction layer training is unsupervised AE training. Its inputs are the bottleneck layer feature extracted by slices of the subject and its outputs are the encoding and decoding features, respectively. For the classification layer training, the slices of a subject, i.e. a 3D slice tensor, will passes the trained AE encoder and be merged, and then enters the classification layer to complete the final training. It is noted that the dimension-reduction layer in Fig. 3 is similar to principal component analysis (PCA) (Jolliffe et al. 2016), which can reduce the dimension and find the main features of an original image. The dimension-reduction layer's encoding and decoding network weights are shared by all slices and thus the weight is only $1/I$ of non-shared method. In the classification layer training, therefore, the trained weights will not be huge after dimension-reducing and weight-sharing. Next, we give more details of the extraction and training.

Let $\mathbf{x}^S$ and D^S be a 2D image vector and its label in a source dataset, respectively. If we pre-train a CNN network in the source dataset to make it satisfy

$$D^S = f_{top}^S\{\mathbf{w}_t^S, \mathbf{f}_{neck}(\mathbf{w}_b, \mathbf{x}^S)\} \tag{2}$$

then the pre-training will be completed, where $f_{top}^S(\cdot)$ and $\mathbf{f}_{neck}(\cdot)$ are the top layer and bottleneck layer function of the pre-trained CNN network,$\mathbf{w}_t^S$ and $\mathbf{w}_b$ are the weight vectors of the top layer and the bottleneck layer, respectively.

After pre-trained, the network will be trained on a target MRI data set to complete the feature extraction of transfer learning. First, complete an dimension-reduction training. Take the i-th slice image matrix $\mathbf{x}_i$ of a subject in the MRI training set as the input of the bottleneck layer $\mathbf{f}_{neck}()$ of the transfer network and let χ_i=$\mathbf{f}_{neck}(\mathbf{w}_b, \mathbf{x}_i)$ be the output feature of the corresponding bottleneck layer. If $\chi_i \approx \hat{\chi}_i$ is satisfied, the dimension-reduction training will be completed, where

$$\hat{\chi}_i = \mathbf{f}_{decode}(\mathbf{w}_d, \zeta_i) \tag{3}$$

$$\zeta_i = \mathbf{f}_{encode}(\mathbf{w}_e, \chi_i) \tag{4}$$

$\mathbf{f}_{encode}()$ and $\mathbf{f}_{decode}()$ are the encoding and decoding functions of the dimension-reduction layer, $\mathbf{w}_d$ and $\mathbf{w}_e$ are their weight vectors, respectively.

Next, complete a classification layer network training. An 3D MRI image of a subject in the training set is sliced into I 2D images to obtain a tensor $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_I]$, which is used as the input of the transfer network $\mathbf{f}_{neck}()$, so that $\mathbf{X}$ and its label D^T satisfy

$$D^T = f_{class}(\mathbf{w}_c, \mathbf{Z}) \tag{5}$$

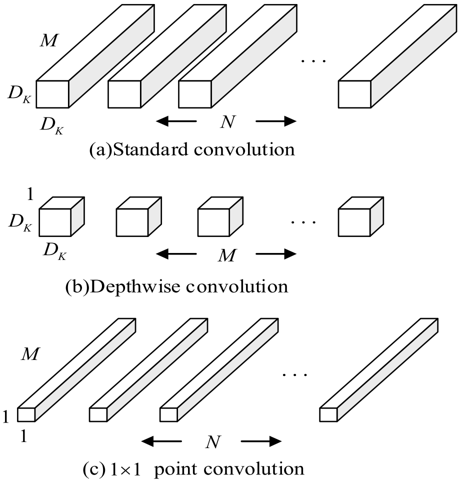
Thus, the training of the target network classification layer $f_{class}()$ is completed, where $\mathbf{w}_c$ is the weight vector of the classification layer, and $\mathbf{Z}=[\boldsymbol{\zeta}_1^{\mathrm{T}}, \boldsymbol{\zeta}_2^{\mathrm{T}}\ldots\boldsymbol{\zeta}_I^{\mathrm{T}}]^{\mathrm{T}}$ is the merged vector.

In fact, the network in Fig. 3 only needs to train the dimension-reduction network and the classification network. Since both of them are shallow networks and the number of weights $\mathbf{w}_e$,$\mathbf{w}_d$ and $\mathbf{w}_c$ is not large, it can ensure that the training can be completed in a short time. In addition, the dimension-reduction feature comes from several slice images of a subject. As long as the number of slices is sufficient, the information loss of the MRI image will be small enough. Moreover, the weights of the dimension-reduction network is shared for all slices. Therefore, the number of weights will not increase with slices. This also ensures less network training time. Of course, dimension-reduction features is extracted from the bottleneck features. Although the weight $\mathbf{w}_b$ of the bottleneck layer $\mathbf{f}_{neck}()$ can be pre-trained, the bottleneck feature extraction will be performed via $\mathbf{f}_{neck}()$. Therefore, a too complex network $\mathbf{f}_{neck}()$ will increase the computational complex of feature extraction. In this paper, we will choose a light-weight CNN network to complete it.

4.3 MobileNet network

In the target training of transfer learning, we will use MobileNet as the bottleneck layer network $\mathbf{f}_{neck}()$. The network is a lightweight CNN network since it changes a standard convolution into a depthwise convolution and a pointwise convolution, as shown in Fig. 4. Thus, it can greatly reduce the computational complex. The details of its application to MRI transfer learning are as follows.

For a slice with N_1×N_2, of MRI data, supposes that its input channel is M, and a feature with $N_o \times N_1 \times N_2$ will be generated after the slice goes through a standard convolution layer, where N_o is the number of output channels. If the standard convolution layer has a convolution kernel with $D_K \times D_K \times M \times N_o$ where D_K is the dimension of the convolution kernel, the computational complexity of the standard convolution, i.e. the number of multiplication is



(a)Standard convolution

(b)Depthwise convolution

(c) 1×1 point convolution

Fig. 4 An illustration for Separable convolution in Mobilenet

$$D_K \times D_K \times M \times N_o \times N_1 \times N_2 \tag{6}$$

In the depthwise convolution, each input channel applies a single filter, and then the computational complexity of the depthwise convolution is

$$D_K \times D_K \times M \times N_1 \times N_2 \tag{7}$$

Then applying a 1 × 1 convolution filter, we will have the complexity with

$$M \times N_o \times N_1 \times N_2 \tag{8}$$

Thus, the computational complexity of the total separable convolution is

$$D_K \times D_K \times M \times N_1 \times N_2 + M \times N_o \times N_1 \times N_2 \tag{9}$$

From (6–9), the ratio for the number of multiplications of the depthwise separable convolution to the standard convolution is

$$\frac{D_K \times D_K \times M \times N_1 \times N_2 + M \times N_o \times N_1 \times N_2}{D_K \times D_K \times M \times N_o \times N_1 \times N_2} = \frac{1}{N_o} + \frac{1}{D_K^2} \tag{10}$$

From (10), it can be seen that the computational complexity of MobileNet can be greatly reduced, compared with the network of the same scale. Therefore, it is a better choice to adopt MobileNet as the transfer network, from the complexity.

4.4 Dimension-reduction and classification layer

In the target training, the bottleneck layer network adopts the transferred MobileNet and does not need to be designed, but the dimension-reduction and the classification layer need to be designed to ensure good feature extraction. The purpose of the dimension-reduction layer is to reduce the dimension of the bottleneck layer features and further extract the features. The dimension-reduction layer design is shown in Fig. 5 and consists of an encoder and a decoder. Both the encoder and decoder have N_S fully connected layers. The purpose of the classification layer network is to merge the features extracted from each slice and complete the final classification. The design of the classification layer $f_{class}()$ is shown in Fig. 6, which mainly includes a flatten layer, N_C fully connected layers, a dropout layer and an output layer. From Figs. 5 and 6, the number of weights in the dimension-reduction and classification layers is

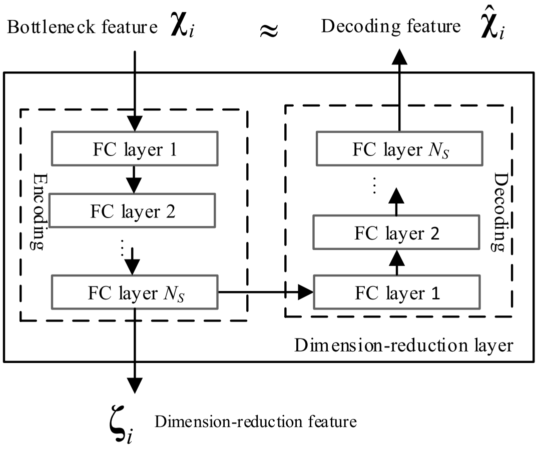

Bottleneck feature χ_i
≈
Decoding feature $\hat{\chi}_i$
FC layer 1
FC layer 2
FC layer N_S
Encoding
FC layer N_S
FC layer 2
FC layer 1
Decoding
Dimension-reduction layer
ζ_i Dimension-reduction feature

Fig. 5 Dimension-reduction layer

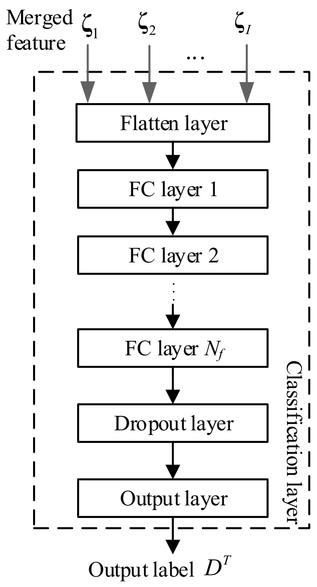


Fig. 6 Classification layer

$$W_{SC} = 2\sum_{n=1}^{N_S}\left(V_{n-1}^S + 1\right) \times V_n^S + \sum_{n=1}^{N_C}\left(V_{n-1}^C \times d_{n-1} + 1\right) \times V_n^C \tag{11}$$

where V_n^S and V_n^C are the number of neurons in the n-th layer of dimension-reduction and classification network, respectively,d_n is the dropout rate of neurons in the n-th of classification network, and 1 is the number of bias. Since the dimension-reduction weight is shared for each slice, the value of V_0^S is only the bottleneck feature dimension of a single slice. If the number of the dimension-reduction layers N_S is small, the first item on the right side of (11) will not be too large. Moreover, the extracted and merged feature dimension V_0^C in the classification layer also decreases to a lower level. If the number of layers N_C is smaller, the second item on the right side of (11) will not be too large.

It should be noted that some parameter settings in the dimension-reduction and the classification layer will affect the final classification result, such as the number of fully connected layers, the number of neurons in the fully connected layer, and the weight dropout rate, etc. How to choose appropriate values of the parameters can be tested by experiment. The discussion for it will be introduced in Section Experiment.

4.5 Steps of algorithm

Finally, the whole training steps for this AD classification algorithm are given in Table 1.

5 Experiment setup

In this experiment, the MRI data used are all from the OASIS database of the University of Washington Alzheimer's Disease Research Center, its website is http://www.oasis-brains.org/, and the downloaded data is OASIS-1 Data set. The data contains 416 male and female subjects aged 18 to 96 years old, and all of the subjects are right-handed, including 100 AD subjects and 316 NC subjects. Table 2 gives the parameters of the data. In addition, each subject data downloaded contains source data and pre-processed data. We chose the pre-processed data, which has been processed through facial features, smoothing, correction, normalization and registration (Marcus et al. 2007). Finally, 100 AD and 100 NC data are selected, the AD group contains 70 very mild, 28 mild and 2 moderate AD, and the NC group data is randomly selected from the database. We do not consider the type of Alzheimer's disease in this paper.

In this experiment, we choose the following transfer learning methods to be evaluated. The parameters for them are listed in Table 3, and some details are as follows.

VGG16_entropy_img: According to the method in (Hon et al. 2017), 32 slices with the highest information entropy of each subject are selected. The bottleneck features are extracted by VGG16, and then classified by the top-layer network. It should be noted that the network is still a 2D network because it can only classify slice images. In particular, in (Hon et al. 2017), several slices of the same subject are randomly allocated to the training set and the test set. Here, the slices of the training set and the test set of this method are divided by the subjects. That is, the slices of the same subject can only be allocated to the training set or the verification set. It cannot

Table 1 Steps of this ad classification algorithm

Input:

A subject's MRI slice vector $\mathbf{X}^{\mathrm{T}} = [\mathbf{x}_1^{\mathrm{T}}, \mathbf{x}_2^{\mathrm{T}}, ...\mathbf{x}_I^{\mathrm{T}}]$ and its label D^T in a training set or a test set

Output:

Dim-reduction weight $\mathbf{w}_e$ $\mathbf{w}_d$ and classification layer weight $\mathbf{w}_c$

known conditions:

Bottleneck layer $\mathbf{f}_{neck}()$ and its weight $\mathbf{w}_b$

Pre-trained top layer $\mathbf{f}_{top}^{S}()$ and its weight $\mathbf{w}_t^S$

Dimension-reduction layer $\mathbf{f}_{encode}()$ and $\mathbf{f}_{decode}()$

Classification layer $f_{class}()$

Initial conditions:

Initial values of weights $\mathbf{w}_e$, $\mathbf{w}_d$ and $\mathbf{w}_c$ are randomly generated

Steps:

1. Bottleneck feature extraction: let $\mathbf{X}^{\mathrm{T}}$ go through the bottleneck layer $\mathbf{f}_{neck}(\mathbf{w}_b, \mathbf{x}_i)$ and get bottleneck features;
2. Dimension-reduction training: in a training set, train the weights $\mathbf{w}_e$ and $\mathbf{w}_d$ via (3–4);
3. Classification layer training: In the training set, train the classification layer weight $\mathbf{w}_c$ via (5);
4. Test: In a test set, get classification results via (5) and calculate a classification accuracy rate;
5. Repeat steps 2–4 until a higher classification accuracy obtained

Table 2 Parameters for OASIS MRI data in this experiment

Parameter	Value
Database	OASIS-1
TR	9.7 ms
TE	4.0 ms
Flip angle	10°
TI	20 ms
TD	200 ms

occurs that some slices of a subject are in the training set and the other ones of the same subject are in the test set.

VGG16_entropy_32: According to the method in (Hon et al. 2017), 32 slices with the highest information entropy of each subject are selected. To enable the literature (Hon et al. 2017) to achieve 3D data classification, we improve it as follows. Besides the slices extracted by Vgg16, the bottleneck feature will go through a top layer and get top-layer features. Finally, all of the top-level features of each slice of the subject are merged and sent to the classification layer for classification.

MobileNet_axial_32: According to the method of slicing by position in (Glozman et al. 2016), the 32 axial slices closest to the center of each subject are selected. After the bottleneck features are extracted by MoblieNet, the bottleneck features are then extracted by the top-layer network. Finally, all of the top-layer features of each slice are merged and sent to the classification layer for classification. Similar to VGG16_entropy_32, this method can also realize 3D MRI data classification. The difference is that the transfer network here is Mobilenet.

AE_axial_32: It is the proposed method, where the slicing method is the same as MobileNet_axial_32, and the other steps are shown in Table 1.

The main difference between MobileNet_axial_32 and AE_axial_32 is feature extraction. The former uses a supervised top layer to reduce dimensionality and extract features, while the latter uses an unsupervised autoencoder (AE).

All methods above use fivefold cross-validation. Randomly divide the total data into 5 parts, where one is selected as a test set, and the others as a training set. Each part must be used as a test set each time and repeat the cross-validation 5 times. It should be noted that the fivefold cross-validation is to divide the data by subjects. That is, all the slices of the same subject are in the same data set. It is avoided that that some slices of a subject are in a training set and the other slices of the same subject are in a test set. According to the above cross-validation, the classification accuracy of each classification method is the average of the five times.

This experiment also gives the results of the running time of each transfer method, including the time to extract bottleneck features, the time to extract top-level features, the time to extract features from AE, the time to classify layers, and the total time to complete a fivefold cross-validation. All methods are performed on Anaconda Python2.7 under Ubuntu, and the transfer learning platform is Keras with TensorFlow as the back end. The operating hardware is a PC with Inter(R) Core(TM) i5-6200U (4-core) Central Processing Unit (CPU), and no graphics processor (Graphics Processing Unit, GPU) is used.

Table 3 Parameters for evaluated methods

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
VGG_entroy_img		**VGG16_entroy_32**		**MobileNet_ axial_32**		**AE_axial_32**	
Transfer Network	VGG16[1]	Transfer Network	VGG16[1]	Transfer Network	MobileNet[1]	Transfer Network	MobileNet[1]
Pre-training data[2]	ImageNet	Pre-training library[2]	ImageNet	Pre-training library[2]	ImageNet	Pre-training library[2]	ImageNet
Number of slices	32	Number of slices	32	Number of slices	32	Number of slices	32
Image width	150	Image width	150	Image width	160	Image width	160
Image height	150	Image height	150	Image height	160	Image height	160
Batchsize	32	Batchsize	32	Alpha	1	Alpha	1
Top layer	Unused	Top layer		Batchsize	32	Batchsize	32
Classification layer		Pooling layer	Global Average	Top layer		AE encoder	
Flatten layer	1	FC layers	1	Pooling layer	Global Average	FC layers	2
FC layers	1	Activation	relu	FC layers	1	Activation	relu
Activation	relu	Number of neurons	256	Activation	relu	Number of neurons	512 128
Number of neurons	256	Dropout	0.5	Number of neurons	256	Dropout	0.5
Dropout	0.5	Output activation	sigmoid	Dropout	0.5	Output activation	sigmoid
Output activation	sigmoid	Classification layer		Output activation	sigmoid	AE decoder	
		Flatten layer	1	Classification layer		FC layers	2
		FC layers	3	Flatten layer	1	Activation	relu
		Activation	relu	FC layers	3	Number of neurons	512 1024
		Number of neurons	1024 1024 256	Activation	relu	Dropout	0.5
		Dropout	0.5	Number of neurons	1024 1024 256	AE Batchsize	32
		Output activation	sigmoid	Dropout	0.5	AE epochs	100
				Output activation	sigmoid	Classification layer	
						Flatten layer	1
						FC layers	2
						Activation	relu
						Number of neurons	512 256
						Dropout	0.5
						Output activation	sigmoid

[1] Both MobileNet and VGG16 use the functions in the Keras platform with Tensorflow as the backend, https://github.com/tensorflow/tensorflow

[2] The pre-trained weights is downloaded from https://github.com/fchollet/deep-learning-models/rele

Table 4 Classification accuracy of evaluated methods

Classification method	Accuracy
VGG16_entropy_img	67.81%
VGG16_entropy_32	75.00%
MobileNet_axial_32	68.75%
AE_axial_32	80.50%

6 Experiment result

6.1 Classification accuracy

First, Table 4 gives the classification accuracy of each method. In the Table, the classification accuracy of VGG16_entropy_image is 67.81%. Since this method only classifies a single slice, and does not merge the slices of a subject, the classification accuracy rate is low. The other three methods in the table merge the slices of a subject. In the three methods, the classification accuracy of MobileNet_axial_32 is low, only 68.75%, and the accuracy of the other two methods exceeds 70%. The result shows that the method to extract top-level features from the slices by position is not very accurate. The highest classification accuracy rate is AE_axial_32 with a classification accuracy of about 80.50%. Compared with the three methods of VGG16_entropy_img, VGG16_entropy_32, and MobileNet_axial_32, it has improved by nearly 12.69%, 5.50%, and 11.75%, respectively. This also shows that the method to use AE to extract features from slices by position has higher classification accuracy than the method to use top layer to extract feature from slices by entropy.

Figure 7 is the feature map extracted from the second fully connected layer in the classification layer, where (a) and (b) are the features of the AD and NC group in the training set, respectively. From the figure, their patterns are different. For example, the values in the first 7 columns of row 3 in Fig. 7a are close to 0, while those in (b) are greater than zero. Figure (c) and (d) are the features of the AD and NC group in the test set, respectively. Their patterns are still different, but the patterns in the test set and the training set are similar. As shown in (a) and (c), the values in the first 12 columns of row 5 are close to 0, and the values in this position in (b) and (d) are also close.

Figure 8 shows the classification accuracy curves of the four classification methods for five trials in the cross validation. From the figure, although the curve of VGG16_entropy_img has less fluctuation, the accuracy of the five trials is low. For MobileNet_axial_32, its fourth trial's classification accuracy is higher, but the other results are lower and its curve fluctuation is large. Although VGG16_entropy_32's curve is higher and stable, its classification accuracy is still lower than AE_axial_32. The curve of AE_axial_32 is at the top, and the classification accuracy of each trial is high. This also indicates that its high average classification accuracy is not owing to some high values in the trials.

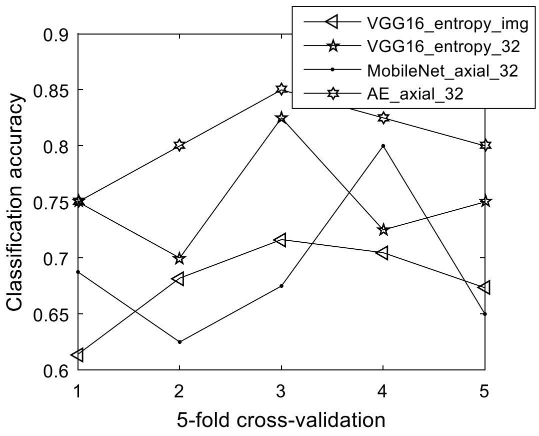
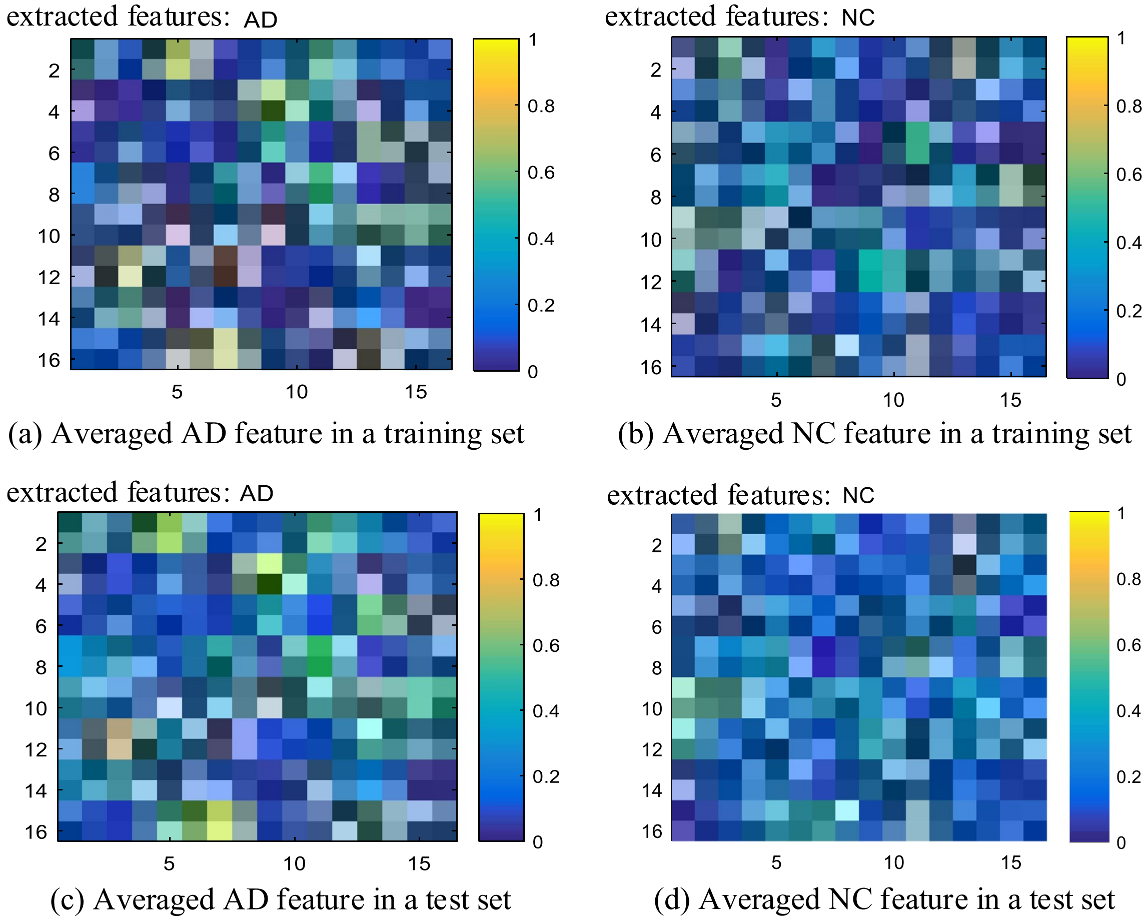


Fig. 8 Classification accuracy curves for classification methods



(a) Averaged AD feature in a training set

(b) Averaged NC feature in a training set

(c) Averaged AD feature in a test set

(d) Averaged NC feature in a test set

Fig. 7 Features extracted from the second fully connected layer in the classification layer

Table 5 Running time for classification methods (unit: second)

Classification algorithm	VGG16_entropy_img	VGG16_entropy_32	MobileNet_axial_32	AE_axial_32
Extract bottleneck feature	2092.6	2092.6	325.5	325.5
Extract top/AE feature	/	110.6	189.4	513.4
Classification layer	976.4	25.6	25.6	25.2
Total time	3069	2228.8	540.5	864.1

6.2 Running time

Table 5 shows the time to extract the bottleneck features, the time to extract the top-layer/AE feature, the time to the classification layer, and the total time. From the table, MobileNet to extract the bottleneck feature takes less time than VGG16, and is a reduction of nearly 84%. It can be seen that, the depthwise separable convolution in MobileNet can greatly reduce the computational complexity. In addition, AE feature extraction takes about 3–4 times more time than MobileNet and Vgg16 to extract top-level features. The reason is that AE's computational complexity is higher due to its unsupervised feature extraction. However, note that the total time of AE_axial_32 is less than VGG16_entropy_32, and the total time does not exceed 2 times more than MobileNet_axial_32. In particular, compared with the 2D transfer network VGG16_entropy_img, the total time of our 3D network AE_axial_32 is only about 1/4 of it. The reason is that the 3D network in this paper uses shared network weights, so the final training time and testing time are greatly reduced. On the other hand, the conventional 2D transfer network does not share the weights of image slices, the large number of the 2D slices increase the training and testing time.

6.3 Other factors in classification methods

This sub-section gives the influence of other factors on the classification of this 3D transfer learning network. First, the influence of the slicing methods on the results is shown in Fig. 9 and Table 6. The slicing methods refer to literatures (Glozman et al. 2016; Hon et al. 2017), as follows.

> MobileNet_acs_32: slicing the MRI images of a subject and select 32 MRI slices close to the center of the MRI, including 12 axial, 10 sagittal and 10 coronal slices;
> AE_entropy_32: slicing the MRI images of a subject and select 32 axial slices with the largest information entropy;
> AE_axial_32: slicing the MRI images of a subject and select 32 axial slices close to the center.

After slicing, the above methods use the steps in Table 1 to classify, and their parameters are the same as AE_axial_32. From Fig. 9 that except the third trial, the classification accuracy of AE_axial_32 in the other four trials is higher. The results show that the method of slices by position is better than the method of slices by entropy, and the slice selection should be in the same direction, such as axial slices.

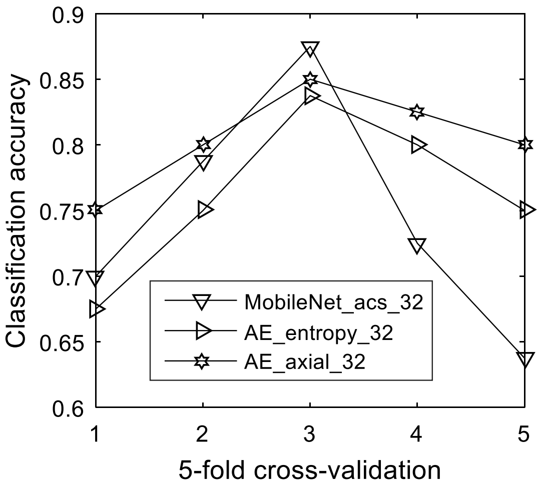


Fig. 9 Classification accuracy curves of slicing methods

Table 6 Classification accuracy of slicing methods

Classification method	Accuracy
MobileNet_acs_32	74.50%
AE_entropy_32	76.25%
AE_axial_32	80.50%

Next, we give the influence of the number of slices on this classification method. Select 80, 60, 32, 20 and 10 axial slices close to the center, respectively. The parameters and steps of the classification method are the same as the above method. The classification results are shown in Fig. 10 and Table 7. From the results, when the number of slices is 20 and 32, the classification accuracy is close. The accuracy of 32 slices is slightly higher than that of 20 slices, but significantly higher than the other three ones. Too many slices will result in higher network complexity and too few slices result in lower classification accuracy. It can be seen from Table 8 that as the number of slices increases, the time for extracting bottleneck features, the

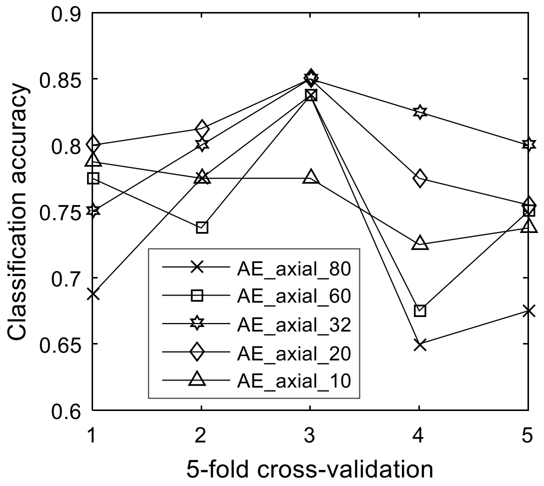


Fig. 10 Classification accuracy curves of different slice quantities

Table 7 Classification accuracy for slice quantity

Classification method	Accuracy
AE_axial_80	72.50%
AE_axial_60	75.50%
AE_axial_32	80.50%
AE_axial_20	79.85%
AE_axial_10	76.00%

time for extracting AE features, and the required classification time are all longer. we find that 32 is the value corresponding to the number of slices when the classification accuracy is highest. And for 32 slices, there are other slices that have more training time than it, so its training time is not the most. Therefore, the selection of 32 slices is the best way.

We give the influence of the number of fully connected layers in the classification layer on the classification of this 3D transfer network. Table 9 shows the average classification accuracy of AE_axial_32 when the number of fully connected layers is 1, 2, 3, and 4, respectively. Figure 11 shows the classification accuracy curves in five trials. It can be seen from the figure that the four classification accuracy curves are close. From the table, the average classification accuracy of 2 fully connected layers is slightly higher than the other three cases. Therefore, from the classification accuracy, a classification layer with 2 fully connected layers is a better choice.

Table 9 Classification accuracy for the number of fully connected layers

Classification algorithm	Accuracy
AE_axial_32_one_layer	79.50%
AE_axial_32_two_layers	80.50%
AE_axial_32_three_layers	78.50%
AE_axial_32_four_layers	79.38%

We also transfer the currently popular network models InceptionV3, DenseNet and VGG16 to compare with the ones in our article to verify the superiority of our method in classification accuracy and classification time. The number of neurons in the AE encoder and decoder in various methods is set as shown in Table 10. It can be seen from Table 11 that the classification accuracy of the AE_axial_32 method is still the highest. We can know from Fig. 12 that the classification accuracy of the five-time cross-validation of the AE_axial_32 method is higher than that of the other three methods, and it can also be seen from Table 12, the transfer learning method we used takes the least time to classify.

We select the raw data of 90 AD and 90 NC subjects in the ADNI dataset (without any preprocessing) to test the several transfer learning methods used in this paper again, and what we select is MRI data. Its website is http://adni.loni.usc.edu/. The steps are briefly described as follows:

ADNI_VGG16_axial_32_img: Select the 32 axial slices closest to the center of the MRI image slices of each subject, put them into VGG16 to extract the bottleneck features, and then classify them through the top-level network.

ADNI_VGG16_axial_32: After the same slice selection, the bottleneck feature is extracted by VGG16, and then the top-level network is used to further extract the top-level features. Finally, the top-level features of each slice of the subject are combined and sent to the classification layer for classification.

ADNI_MobileNet_axial_32: The difference from the ADNI_VGG16_axial_32 method is the use of MobileNet for bottleneck feature extraction.

Table 8 Classification time for slice quantity

Classification algorithm	AE_axial_10	AE_axial_20	AE_axial_32	AE_axial_60	AE_axial_80
Extract bottleneck feature	74.8	142.2	325.5	507.0	715.0
Extract AE feature	169.0	308.6	513.4	928.5	1265.0
Classification layer	14.0	16.6	25.2	46.0	83.0
Total time	257.8	467.4	864.1	1481.5	2063

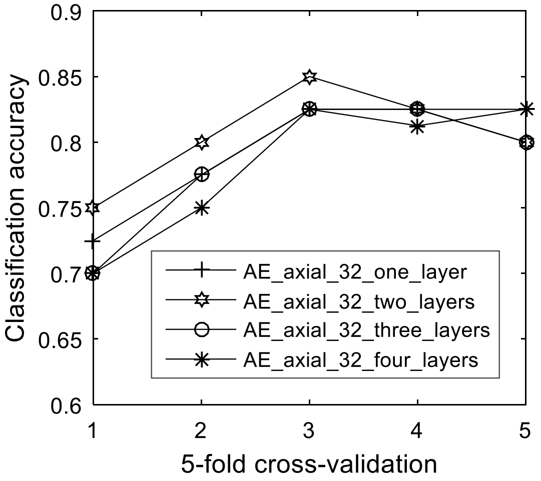


Fig. 11 Classification accuracy curves for the number of fully connected layers

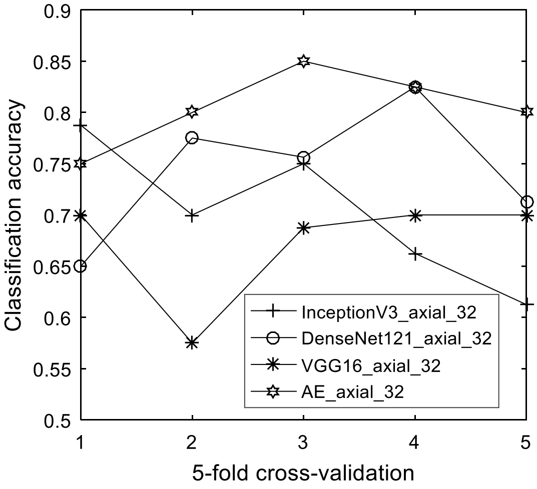


Fig. 12 The classification accuracy curve of different transferred CNN pre-training networks

Table 10 Number of AE encoder and decoder neurons when transferring different network models

Classification method	Number of AE encoder neurons	Number of AE decoder neurons
InceptionV3_axial_32	512 、128	512 、2048
DenseNet121_axial_32	512 、128	512 、1024
VGG16_axial_32	256 、128	256 、512
AE_axial_32	512 、128	512 、1024

Table 11 Classification accuracy of different transferred CNN pre-training networks

Classification method	Accuracy
InceptionV3_axial_32	70.25%
DenseNet121_axial_32	74.38%
VGG16_axial_32	67.25%
AE_axial_32	80.50%

ADNI_AE_axial_32: The same slicing method. After extracting bottleneck features using MobileNet, it enters the autoencoder for AE feature extraction. Finally, the AE features from each slice are combined and sent to the classification layer for classification.

From Table 13 and Fig. 13, we can see that using ADNI data for experiments, ADNI_AE_axial_32 has the highest classification accuracy, which is consistent with the conclusions we obtained from experiments using OASIS data.

7 Conclusion and discussion

This paper proposes a 3D transfer machine learning method that classify an AD and NC group with MRI. In the experiment, we use OASIS-1 MRI data to compared the proposed method with other traditional methods. The results show that the proposed 3D transfer network improves the classification accuracy of the conventional 2D transfer network by about 10 percentage points, and the total classification time is only its 1/4. We also verify our method with the data in the ADNI database, and the conclusion is consistent with the OASIS data. In addition, different transferrable CNN networks are also used by us to compare with the method in this paper, and our method showed better performance in training time and classification accuracy. The contribution of the proposed method is to utilize a 2D transfer CNN to establish a 3D transfer CNN, which reduces network complexity, improves classification accuracy, and reduces classification time. The method of the feature extraction and mergence with subjects instead of the feature itself is more reasonable, and transferring the MobileNet network, a deep separable convolution has lower complexity. Of course, although the 3D transfer network in this paper has achieved good performance in classification accuracy and classification time, there are several points that need further discussion.

Compared with the traditional 3D CNN networks that directly takes 3D MRI images as input (Hosseini-Asl et al. 2016), the classification accuracy of this transfer learning has not been significantly improved. However, the conventional 3D CNN method will produce too many weights and too long training time due to the inputs of 3D images. On the other hand, our method utilizes shared weights for 2D slices images and transfer pre-trained 2D networks to our target data. This makes the amount of trained weights smaller and

Table 12 Classification time for different transferred CNN pre-training networks

Classification algorithm	InceptionV3_axial_32	DenseNet121_axial_32	VGG16_axial_32	AE_axial_32
Extract bottleneck feature	327.6	750.8	1140	325.5
Extract AE feature	935.2	535.4	175.6	513.4
Classification layer	26.2	26.6	26.4	25.2
Total time	1289	1312.8	1342	864.1

Table 13 Classification accuracy of different ADNI classification methods

Classification method	Accuracy
ADNI_VGG16_axial_32_img	55.49%
ADNI_VGG16_axial_32	60.83%
ADNI_MobileNet_axial_32	63.80%
ADNI_AE_axial_32	66.25%

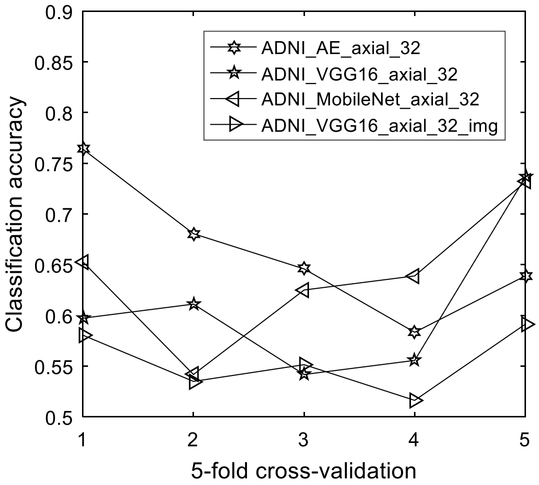


Fig. 13 Classification accuracy curve of different ADNI classification methods

the training time shorter. Therefore, as a assisted AD diagnosis, the transfer network in this paper has some advantages in computing costs and training time. In addition, this paper only uses MRI for AD classification while the high classification accuracy of the literature (Zhang et al. 2011; Suk et al. 2014; Liu et al. 2014; Lee et al. 2019) is based on the multimodal classification. Besides MRI, the multimodal method uses PET and cerebrospinal fluid. Moreover, there are AD classifications with fMRI data (Wang et al. 2007, 2006). Therefore, we can also consider more types of data to further improve the performance of this transfer network.

This experiment uses the data from the OASIS and ADNI, and has not used other data sets. Rigorously, more databases should be tried to obtain a more reliable classification accuracy. Since the traditional transfer network (Hon et al. 2017) compared in this paper uses the OASIS data set, the selection of this database can get a more direct comparison result. ADNI data is used by us for comparison. In future work, we can also use databases such as CADDementia. In addition, we only complete the classification of AD and NC. In the future, we can also try to achieve the classification of AD, MCI, and NC (Glozman et al. 2016), which will be helpful to early diagnosis and early treatment.

For the number of slices, this experiment gives only the results of 10, 20, 30, 60 and 80 slices of a subject, and does not try the other number of slices. Because the slices that are far from the center contain less structural information and more slices will produce more redundant information, too many slices reduce the classification accuracy and increase the running time. Therefore, more slices are not considered.

In this paper, the number of neurons of encoder in the dimension-reduction network should be equal to that in the decoder, and the number of neurons in its hidden layer should be less than that in its input layers. Therefore, the number of neurons in the encoder's two fully connected layers is set to 512, 128, and the decoder is set to 512, 1024, respectively. In dimension-reduction training, epochs are set to 5000 times and 100 times, respectively, but the classification accuracy of the two is nearly the same. This shows that 100 times training can also extract good features. For the parameters in the classification layer, this experiment only gives the results of the number of fully connected layers, and does not give the discussion of other parameters. This is mainly because compared with other parameters, the number of fully connected layers has a greater influence on the classification results. For the activation function, we use a Relu and Sigmoid function. The reason why the Relu function is selected in the hidden layer is that, it can solve the problem of gradient disappearance and the calculation is efficient. Since this method is for a two-classification, the Sigmoid function is selected in the output layer. Of course, there are some parameters important to the performance of the classification layer, such as the number of neurons in the fully connected layer. Generally, the number of neurons can be determined empirically. If the number is too small, the network cannot adapt to large-size images. If the number is too large, it will increase training time and may cause overfitting. Therefore, the number of neurons in the two fully connected layers of the classification layer network is set to 512 and 256, respectively.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant 62161052 and by the Program for Innovative Research Team (in Science and Technology), University of Yunnan Province.

References

Abed MT, Fatema U, Nabil SA et al (2020) Alzheimer's disease prediction using convolutional neural network models leveraging preexisting architecture and transfer learning. Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR). IEEE, 1–6

Aël Chetelat G, Baron JC (2003) Early diagnosis of Alzheimer's disease: contribution of structural neuroimaging. Neuroimage 18(2):525–541

Arevalo-Rodriguez I, Smailagic N, Figuls MR et al (2015) Mini-mental state examination (MMSE) for the detection of Alzheimer's disease and other dementias in people with mild cognitive impairment (MCI). Cochrane Database Syst Rev 3:2

Ashraf A, Naz S, Shirazi SH et al (2021) Deep transfer learning for alzheimer neurological disorder detection. Multimed Tools Appl 2:1–26

Basaia S, Agosta F, Wagner L et al (2019) Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. NeuroImage Clin 21:101645

Blennow K, Hampel H (2003) CSF markers for incipient Alzheimer's disease. Lancet Neurol 2(10):605–613

Bron EE, Smits M, Van Der Flier WM et al (2015) Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CADDementia challenge. Neuroimage 111:562–579

Chollet F (2015) Keras. https://keras.io/.

Deng J, Dong W, Socher R et al (2009) Imagenet: a large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp. 248–255

Dessouky MM, Elrashidy MA, Abdelkader HM (2013) Selecting and extracting effective features for automated diagnosis of Alzheimer's disease. Int J Comput Appl 81:4

Farooq A, Anwar S M, Awais M et al (2017) A deep CNN based multi-class classification of Alzheimer's disease using MRI. In 2017 IEEE International Conference on Imaging systems and techniques (IST). IEEE, pp. 1–6

Feng C, Elazab A, Yang P et al (2019) Deep learning framework for Alzheimer's disease diagnosis via 3D-CNN and FSBi-LSTM. IEEE Access 7:63605–63618

Frisoni GB, Fox NC, Jack CR et al (2010) The clinical use of structural MRI in Alzheimer disease. Nat Rev Neurol 6(2):67–77

Gao XW, Hui R, Tian Z (2017) Classification of CT brain images based on deep learning networks. Comput Methods Programs Biomed 138:49–56

Glozman T, Liba O (2016) Hidden cues: Deep learning for Alzheimer's disease classification CS331B project final report

Harrison J, Minassian SL, Jenkins L et al (2007) A neuropsychological test battery for use in Alzheimer disease clinical trials. Arch Neurol 64(9):1323–1329

Hon M, Khan N M (2017) Towards Alzheimer's disease classification through transfer learning. In 2017 IEEE International conference on bioinformatics and biomedicine (BIBM) (pp. 1166–1169). IEEE.

Hosseini-Asl E, Keynton R, El-Baz A (2016) Alzheimer's disease diagnostics by adaptation of 3D convolutional network. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 126–130). IEEE.

Howard AG, Zhu M, Chen B et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv: 1704.04861.

Jain R, Jain N, Aggarwal A et al (2019) Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. Cogn Syst Res 57:147–159

Johnson KA, Fox NC, Sperling RA et al (2012) Brain imaging in Alzheimer disease. Cold Spring Harbor Perspect Med 2(4):a006213

Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Philos Trans R Soci Math Phys Eng Sci 374(2065):20150202

Kim KW, Lee DY, Jhoo JH et al (2005) Diagnostic accuracy of mini-mental status examination and revised hasegawa dementia scale for Alzheimer's disease. Dement Geriatr Cogn Disord 19(5–6):324–330

Kumar SS, Nandhini M (2021) Entropy slicing extraction and transfer learning classification for early diagnosis of Alzheimer diseases with sMRI. ACM Trans Multimed Comput Commun Appl 17(2):1–22

Lee G, Nho K, Kang B et al (2019) Predicting Alzheimer's disease progression using multi-modal deep learning approach. Sci Rep 9(1):1–12

Liu S, Liu S, Cai W et al (2014) Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease. IEEE Trans Biomed Eng 62(4):1132–1140

Marcus DS, Wang TH, Parker J et al (2007) Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. J Cogn Neurosci 19(9):1498–1507

Mehmood A, Yang S, Feng Z et al (2021) A transfer learning approach for early diagnosis of alzheimer's disease on MRI images. Neuroscience 460:43–52

Orru G, Pettersson-Yeo W, Marquand AF et al (2012) Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. Neurosci Biobehav Rev 36(4):1140–1152

Petersen RC, Aisen PS, Beckett LA et al (2010) Alzheimer's disease neuroimaging initiative (ADNI): clinical characterization. Neurology 74(3):201–209

Suk HI, Lee SW, Shen D et al (2014) Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. Neuroimage 101:569–582

Thies W, Bleiler L (2013) 2013 Alzheimer's disease facts and figures. Alzheimers Dement 9(2):208–245

Tufail AB, Ma YK, Zhang QN (2020) Binary classification of Alzheimer's disease using sMRI imaging modality and deep learning. J Digit Imaging 33(5):1073–1090

Wang L, Zang Y, He Y et al (2006) Changes in hippocampal connectivity in the early stages of Alzheimer's disease: evidence from resting state fMRI. Neuroimage 31(2):496–504

Wang K, Liang M, Wang L et al (2007) Altered functional connectivity in early Alzheimer's disease: A resting-state fMRI study. Hum Brain Mapp 28(10):967–978

Yagis E, Citi L, Diciotti S et al (2020) 3D convolutional neural networks for diagnosis of Alzheimer's disease via structural MRI

Zhang D, Wang Y, Zhou L et al (2011) Multimodal classification of Alzheimer's disease and mild cognitive impairment. Neuroimage 55(3):856–867

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.