Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Deep sequence modelling for Alzheimer's disease detection using MRI

Amir Ebrahimi [*], Suhuai Luo, Raymond Chiong, for the Alzheimer's Disease Neuroimaging Initiative[1]

School of Electrical Engineering and Computing, The University of Newcastle, NSW 2308, Australia

ARTICLE INFO

Keywords:
Convolutional neural network
Temporal convolutional network
Recurrent neural network
Alzheimer's disease
MRI

ABSTRACT

Background: Alzheimer's disease (AD) is one of the deadliest diseases in developed countries. Treatments following early AD detection can significantly delay institutionalisation and extend patients' independence. There has been a growing focus on early AD detection using artificial intelligence. Convolutional neural networks (CNNs) have proven revolutionary for image-based applications and have been applied to brain scans. In recent years, studies have utilised two-dimensional (2D) CNNs on magnetic resonance imaging (MRI) scans for AD detection. To apply a 2D CNN on three-dimensional (3D) MRI volumes, each MRI scan is split into 2D image slices. A CNN is trained over the image slices by calculating a loss function between each subject's label and each image slice's predicted output. Although 2D CNNs can discover spatial dependencies in an image slice, they cannot understand the temporal dependencies among 2D image slices in a 3D MRI volume. This study aims to resolve this issue by modelling the sequence of MRI features produced by a CNN with deep sequence-based networks for AD detection.

Method: The CNN utilised in this paper was ResNet-18 pre-trained on an ImageNet dataset. The employed sequence-based models were the temporal convolutional network (TCN) and different types of recurrent neural networks. Several deep sequence-based models and configurations were implemented and compared for AD detection.

Results: Our proposed TCN model achieved the best classification performance with 91.78% accuracy, 91.56% sensitivity and 92% specificity.

Conclusion: Our results show that applying sequence-based models can improve the classification accuracy of 2D and 3D CNNs for AD detection by up to 10%.

1. Introduction

The main idea behind deep learning—part of a broader family called machine learning—is based on neural networks inspired by data processing nodes in biological systems; 'deep' refers to multiple layers in the networks. Not a relatively new concept, the application of deep learning has not been well investigated until recent advancements in graphics processing units (GPUs) and the development of various new algorithms for efficiently training deep-learning models [1]. In recent years, numerous studies have used deep learning for classification, regression and segmentation. In particular, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have performed outstandingly in image-based and sequence-based decision-making tasks, respectively. CNNs employ spatial information of images and extract features by assembling convolutional layers to create a hierarchy of features to make a decision [2]. Instead of using vector-based inputs, which is the case in typical neural networks, CNNs capture the structural information among neighbouring pixels. Conversely, RNNs have a memory to capture temporal dependencies in sequence-based tasks. The output for the most recent sequential input is calculated using the corresponding input data and by considering previous input data stored in hidden units.

CNNs and RNNs can be applied to image- and vector-based time-series tasks, respectively. Although three-dimensional (3D) CNNs are the most straightforward method for a sequence of images like video-based applications and 3D medical images, their structure is highly complex. It necessitates many parameters for training; 2D CNNs can be fed by 2D or

[*] Corresponding author.
E-mail address: Amir.Ebrahimi@uon.edu.au (A. Ebrahimi).

[1] Data utilised in this paper were provided by the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). Researchers in the ADNI participated in the implementation and design of ADNI but not in writing or analysis of this study. A list of ADNI researchers is available at: http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

https://doi.org/10.1016/j.compbiomed.2021.104537
Received 19 February 2021; Received in revised form 27 May 2021; Accepted 27 May 2021
Available online 1 June 2021

3D images. A 3D image volume can be handled as a sequence of 2D image slices. For 3D images, 2D CNNs can capture spatial dependencies from images, but are incapable of understanding temporal relations in a sequence of images. Conversely, RNNs can be applied in time-series applications with their embedded memory. They can only be fed by numerical vectors as features, but vectorisation eliminates structural information in images in image-based applications.

This paper investigates the application of deep learning in Alzheimer's disease (AD) detection using magnetic resonance imaging (MRI). AD, the most widespread kind of dementia (about 60–80% of all dementia cases), is a fatal disorder that causes brain cells to die [3]. According to estimates, dementia affects about 50 million people worldwide and 459,000 Australians in 2020 [3,4]. With 15,016 deaths in 2019, it is currently the second-highest cause of death in Australia [5]. In practice, clinical checks and questionnaires are used to detect AD, but this is challenging given the limited current knowledge about the disease. In recent years, the exploration of novel deep models, especially for medical image processing, has become popular [1]. The number of published articles in this area of research exploded in 2017 [6]. Several deep models have been utilised for AD detection, using brain scans such as MRI. Successful classification requires distinguishing specific patterns in MRI scans, leading to classifying patients with AD from healthy normal controls (NCs).

AD detection from neuroimaging is difficult. Various machine-learning methods have been explored for AD detection. However, mainstream machine-learning approaches are incapable of addressing such a complex problem, since highly discriminative features are required to distinguish similar brain patterns [2,7,8]. The main purpose of feature extraction is to establish a set of information that should convey the disease-related patterns for AD detection. To use 3D MRI scans to detect AD using deep models, input data management should be considered. According to the literature [6], input data management methods can be arranged into four categories: slice-based, patch-based, voxel-based and ROI-based. Slice-based structures reduce the number of learnable parameters by supposing that features of interest are included in 2D image slices. Patch-based methods can take brain AD-related patterns by extracting features from small 3D cubes in the brain, called 'patches'. Voxel-based methods are the most direct, using voxel intensity values from the entire 3D brain scan. ROI (region of interest) methods emphasise specific AD-related segments of the brain, rather than the whole brain. The definition of ROIs usually requires previous knowledge of the abnormal regions related to AD, such as the hippocampus. The main challenge in ROI- and patch-based approaches is to select the most informative AD-related image regions or patches.

Along with data management methods, different types of deep-learning models have been employed for AD detection. On top of them, there are CNNs [6]. 2D and 3D CNNs are usually applied to slice- and voxel-based methods, respectively. CNNs can efficiently capture disease-related patterns in brain scans. However, deep models require a large dataset to be trained on, but large datasets are not available for AD. In addition to the previously mentioned benefits of 2D CNNs over 3D CNNs, the former can employ the idea of *transfer learning*, which refers to transferring knowledge from one task to another. Thus, a 2D CNN model can be trained on an arbitrary dataset of millions of samples and retrained on a specific dataset of AD patients. This is possible since the filters associated with convolutional layers of a CNN can extract general features that are beneficial to many tasks.

The main disadvantage of using 2D CNNs on 3D MRI scans (slice-based approaches) is that 2D CNNs are incapable of understanding voxels' dependencies in MRI volumes. When converting 3D MRI scans to 2D image slices, the loss of data will happen. That is due to the fact that brain regions span over 2D slices of an MRI scan. By splitting the scan, features related to brain regions' sizes and shapes will be lost. The main disadvantage of 3D CNNs (voxel-based approaches) is that their structure is highly complex and requires many training parameters, which may cause overfitting. Also, they cannot benefit from transfer learning using datasets with millions of 2D images such as ImageNet. To address this issue, we propose sequence-based approaches and compare them with slice-based and voxel-based approaches. We investigate the possibility of using a combination of image- and sequence-based models to detect AD. Thus, after dividing 3D MRI scans into 2D image slices, a 2D CNN is used to extract the features. Then, a sequence-based deep model is employed to find the relation between sequences of features. The sequence-based deep model can be a type of RNN, such as long short-term memory (LSTM) [9], bidirectional LSTM (BiLSTM) [10] and gated recurrent unit (GRU) [11]. We also design a temporal convolutional network (TCN) [12] to perform feature extraction from images and understand temporal dependencies simultaneously. To the best of our knowledge, TCNs have not been employed for AD detection previously in the literature.

Section 2 reviews recent studies on AD detection using CNNs and RNNs. We then explain the proposed structure of CNNs, RNNs and TCNs together with their fundamental mathematics. Following this, the presented models' results are compared and discussed, and conclusions are drawn.

2. Related work

CNNs are the most common deep model utilised to detect AD [6]. Inspired by the brain's visual cortex, CNNs can take 2D or 3D images as input data and extract features by assembling several convolutional layers. In contrast to typical classifiers in machine learning, CNNs combine feature extraction and classification in a single entity. Initially presented in Ref. [13], CNNs attracted great attention after their outstanding performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [14]. CNNs, such as ResNet [15], have been successfully employed to classify 1000 different classes on a dataset of about one million images.

CNNs are built using convolutional layers ending with a Softmax layer and with other layers in between, including, but not limited to, activation layers, batch normalisation layers, pooling layers, drop-out layers and fully connected layers. The key part is the convolutional layer, which applies filters to the input images and extracts features. Batch normalisation layers typically appear after convolutional layers to normalise the previous layer's output on each mini-batch by deducting the mini-batch mean and dividing by the mini-batch standard deviation. A nonlinear activation function—typically ReLU—generally follows to enable learning of complex representations. Pooling layers down-sample features by calculating the local average or maximum, which reduces the number of learnable parameters while preserving influential features. Drop-out layers prevent a model from overfitting by dropping some neurons randomly at each update of the training process, thereby reducing computational load and forcing neurons to act independently. Neurons in each fully connected layer are connected to all feature elements in their previous layer, just like traditional neural networks. In the end, a Softmax function presents the greatest value in the output vector while suppressing the rest.

Initially, CNNs were applied to 2D images for decision-making tasks. However, employing 3D CNNs is also popular, since it is the most straightforward way to detect AD from 3D MRI scans; 3D CNN models were proposed based on VGGNet and ResNet in [16–20]. However, 3D CNN models have many learnable parameters compared with 2D CNNs and cannot benefit from transfer learning. Hence, the use of 2D CNNs is more common in this field of research. Supposing that the main features in 3D MRIs are preserved in 2D image slices, MRI volumetric data can be divided into 2D images. Generally, 2D CNNs are trained on MRI scans in the form of 2D image slices; and all image slices of one patient are classified for disease detection. In 3D medical image analysis using 2D CNNs, research studies employ standard planes of brain scans, such as the coronal plane, the sagittal plane, the axial plane or a combination of these. It has been shown that the discriminative AD-related features are covered by the coronal view of a brain scan [7,21].

Among studies using 2D CNNs, several deep models were designed with two [7,22], three [23], five [24] or six convolutional layers [25]. However, pre-trained CNN models such as DenseNet-121 [21], VGGNet-16 [26], GoogLeNet, ResNet-18 and ResNet-152 [27], ResNet-18 [28], GoogLeNet and ResNet-152 [29], Inception-V3 [30], CaffeNet and GoogLeNet [31], LeNet and GoogLeNet [32], or VGGNet-16 and Inception-V4 [33,34] are also popular. These methods were all applied on a single view of brain scans; however, using all three views can offer complement features [35,36]. The main weakness of multi-view methods is possible ambiguities in the final decision. In another study, a consensus multi-view clustering model was proposed to detect AD. The authors created 12 views from an initial MRI dataset using various feature extraction methods, such as the Gabor filter. After pre-processing, the processed multi-view data were fed into a matrix factorisation model for classification [37].

Compared with CNNs, RNNs are used less often for AD detection. RNNs are specially designed for temporal tasks, such as video or text processing. Although image-based tasks contain spatial rather than sequential information, a 3D brain scan can be managed as a sequence of 2D image slices. A typical model in corresponding studies is to employ a neural network to extract features and an RNN to address the features. RNNs can be beneficial for AD detection in two types of studies: cross-sectional and longitudinal. In cross-sectional studies, RNNs are applied to evaluate a subject at a specific time. General cross-sectional methods extract features from image slices of a brain scan and find their relationship using RNNs [8,38,39]. In longitudinal studies, RNNs follow subjects over time to evaluate AD progression. General longitudinal methods extract features from brain scans captured over time and understand the disease progression among them using RNNs [40–42]. RNNs are not as deep as deep multilayer neural networks or CNNs in terms of the number of layers. They have difficulties memorising long-term sequences and require large datasets for training [2]. Fortunately, more complex structures, such as LSTM or GRU, help prevent memorising problems [38,40]. Currently, RNNs are utilised for sequence-to-one and sequence-to-sequence decision-making tasks. In sequence-to-sequence tasks, a decision is made in each time step; in contrast, in sequence-to-one tasks, a single decision is made on a predefined number of time steps. For example, a short video can have one label or one label per frame for a classification task.

Presently, in the field of deep learning, sequence-based tasks are commonly handled by RNN structures. Recently, it has been shown that CNNs can be applied to general sequence modelling tasks. Similar to recurrent networks, CNNs can operate on fixed- or variable-length input sequences and be employed to model sequence-to-sequence and sequence-to-one tasks. The original TCN proposed by Bai et al. [12] was evaluated across a wide range of standard tasks (e.g. character-level and word-level language modelling) that are usually employed to benchmark recurrent networks. It was shown that its functionality could match or even outperform RNNs. Better parallelism and control of the network's memory in the training process are other benefits of TCNs. Also, in contrast with previous AD detection approaches using RNNs, a TCN model can be employed for feature extraction and classification, simultaneously. The other applications of TCNs include, but are not limited to, generation of financial time series, human activity and gesture recognition, speech separation, enhancement, and recognition, image captioning, soccer ball detection and tracking and traffic flow forecasting [12]. Thus far, TCNs have not been applied to medical images for any purpose.

3. The proposed method

Training a CNN model consists of forward/backward steps to calculate the *loss* function between the ground truth labels and predicted output. Then, a penalisation term is applied with chain rules to update learnable parameters. While 3D CNNs can capture the full spatial information from 3D MRIs, they are challenging to train and incapable of benefiting from transfer learning. In transfer learning, learnable parameters (*weight* and *bias*) can be initialised by training deep models on other tasks with millions of images, related or unrelated. These parameters can serve in an AD detection system since they can extract general features from images. In this study, we used ImageNet, a dataset of 1000 object categories [14], to initialise our 2D CNN model. Training a 2D CNN is quite simple, but they are incapable of capturing the spatial information of 3D MRIs because of the absence of the third dimension in convolving filters [43].

To use transfer learning and understand 3D patterns in MRI scans, we extracted features from MRI image slices with a pre-trained 2D CNN and fed the sequence of the extracted features to an RNN. An overview of data processing involving these models can be found in Fig. 1. The RNN is responsible for understanding the relationship between the sequence of extracted features corresponding to MRI image slices. However, the feature extraction step in the CNN is independent of the classification step in the RNN. To avoid this, a TCN model was proposed. TCNs can perform feature extraction and classification in sequence-based tasks simultaneously. Without using CNNs, they can extract features from 2D image slices and find the relationship between a series of 2D image slices. Similar to RNNs, TCNs can be applied on fixed- or variable-length input sequences. Further, TCNs can be utilised to model sequence-to-one or sequence-to-sequence tasks.

In this section, we discuss data collection and processing. Then we explain our 2D CNN model to demonstrate insights in feature extraction. Finally, we discuss the proposed RNN and TCN models.

3.1. Data collection and processing

The ADNI[2] study, which is the most commonly used dataset in this field [6], supplied the dataset utilised in this paper. It has been used in about 90% of studies by itself or in combination with other datasets. The ADNI study's main goal is to test the effectiveness of MRI and other biomarkers in measuring the progress of AD. Baseline or screening MRI scans of 225 subjects for each class (AD and NC) were employed in our study. Subjects' identification numbers and data statistics are available in Appendix 1, while the demographic details are also presented in Table 1; the dataset itself is available online upon request. The reason for selecting the same number of samples for each class was to prevent prediction bias because of an imbalanced dataset. It is possible to access many MRI scans from healthy people and include them in our dataset to increase the dataset size. However, the number of subjects with AD is limited in medical datasets. Therefore, to avoid class imbalance and subsequent prediction bias to one of the classes—NC in this case—the same number of MRI scans was selected for each class.

Overfitting is a challenging topic in deep learning that may occur because of issues such as a low number of subjects and a large number of learnable parameters. There are numerous ways to minimise overfitting, as reported in the literature [44,45]. The first solution is to use data augmentation to increase the dataset size by slightly modifying the available data. Medical datasets have few subjects, and obtaining new data is difficult; thus, data augmentation is commonly used to classify medical data. Another workaround is to use transfer learning, a powerful tool to enable training a large network without overfitting. Transfer learning allowed us to perform feature extraction using knowledge from another dataset with more samples. Regarding the large number of learnable parameters, various CNN models—from Lenet-5 to ResNet-101—with a different number of learnable parameters were utilised to prevent overfitting. More details on the employed CNN models, their structures and training options to avoid overfitting are provided in Sections 3.2 and 4.

The two most common image pre-processing steps in the literature were adopted: intensity normalisation and registration [6]. Intensity normalisation involves mapping the intensities of all pixels or voxels onto a reference scale. In our experiments, for each pixel/voxel, we subtracted the mean and divided by the standard deviation of the whole

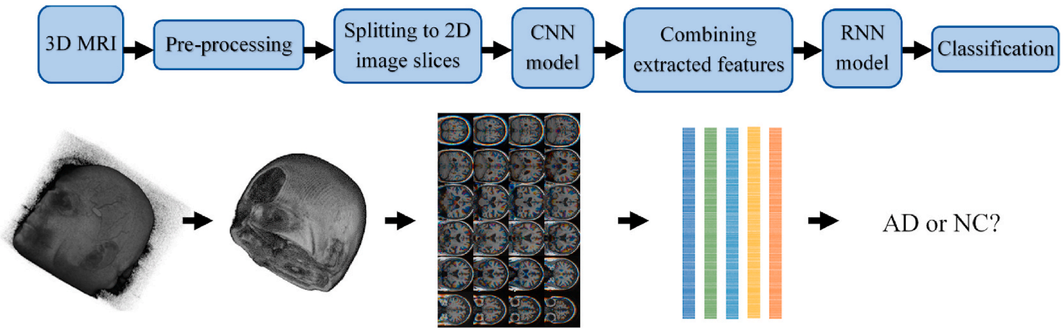

Fig. 1. A block diagram of our AD detection system involving the CNN and RNN models.

Table 1
Demographic details of our ADNI dataset.

	AD	NC
Male	117	109
Female	108	116
Age[a]	75.52 ± 7.94	73.88 ± 6.63
Total	225	225

[a] (Mean ± Standard Variation).

input data. Therefore, voxel intensities of all MRI scans were mapped to be zero-centred. In the process of training a model, there will be multiplying of (weights) and adding to (biases) these initial inputs, leading to activations that backpropagate with the gradients to train the model. In this process, each feature must have a similar range so that the gradients do not go out of control.

Registration is the process of spatially aligning image scans to a reference anatomical space. It is essential due to the complexity of brain structures and the differences between different subjects' brains. Image registration aids in standardising the MRI scans regarding a common fixed-size template. This alignment makes it possible to compare the voxel intensities of brain scans from different subjects, ensuring that a certain voxel in one scan has the same anatomical position as in the brain of another scan. To standardise MRIs to a standard pattern, they were spatially adjusted to the Montreal Neurological Institute (MNI) space [46] using the SPM12 toolbox [47].

After adjusting to the MNI space, each MRI scan's dimension was 79 × 95 × 79 (in voxels). As discussed previously, the coronal plane is reported to contain the most notable AD-related parts of the brain. Further, coronal sequences have a longer length, which is beneficial in sequence-based deep models. From 95 coronal slices of an MRI with the size of 79 × 79 (in pixels), 23 slices were marked manually and discarded from the beginning and the end of the slices, as they mostly contained the skull or background. Since every MRI scan was registered to the MNI template, all the subjects start with the same brain regions that include brain tissues. As our 2D CNN model was pre-trained on the ImageNet dataset, which takes RGB (Red-Green-Blue) colour images as the input, RGB images were required. In this case, 72 remaining grayscale images were formed into 24 RGB coronal images by stacking three adjacent slices as RGB colour channels. Then, each MRI was resized to match the input layer of our CNN model using bilinear interpolation.

3.2. The CNN model

The performance of deep CNNs might be degraded because of the vanishing gradient issue. In the backpropagation training process, the gradient might infinitely decrease when it is backpropagated to previous layers. In this paper, we utilised the well-known ResNet-18 model [15], which presents the idea of 'shortcut connections' that skip some layers to

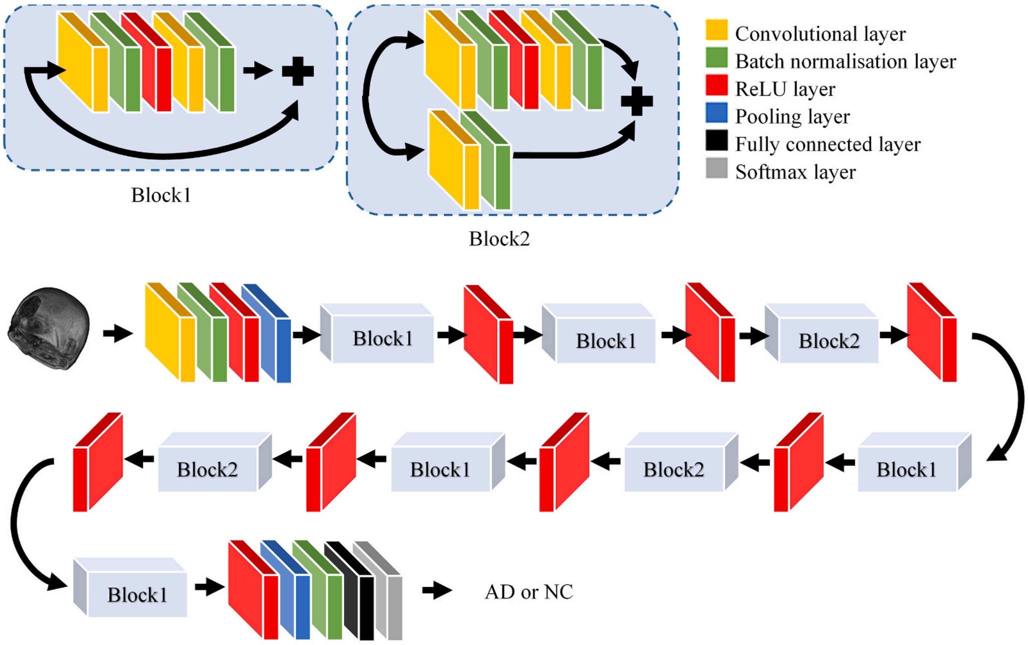

Fig. 2. The ResNet-18 implemented in this study.

avoid the vanishing gradient problem. Fig. 2 shows the implemented structure of ResNet-18. The skipped routes are shown in Block1 and Block2 of Fig. 2. This model has a depth of 18–71 layers, including 20 convolutional layers and one fully connected layer and about 12 million learnable parameters, with image input sizes of 224 × 224 (in pixels). The network depth is defined as the largest number of sequential fully connected layers and convolutional layers on the route from the input layer to the Softmax layer. Coronal images of size 79 × 79 were resized to 224 × 224 with the bilinear interpolation method, then fed to the ResNet-18 model. The corresponding features of each coronal image were extracted before the fully connected layer. A vector with 512 elements was produced with the extracted features. Considering all coronal images of an MRI scan, finally, a sequence of 512-element vectors with a length of 24 was obtained. ResNet-18 has already performed well on the

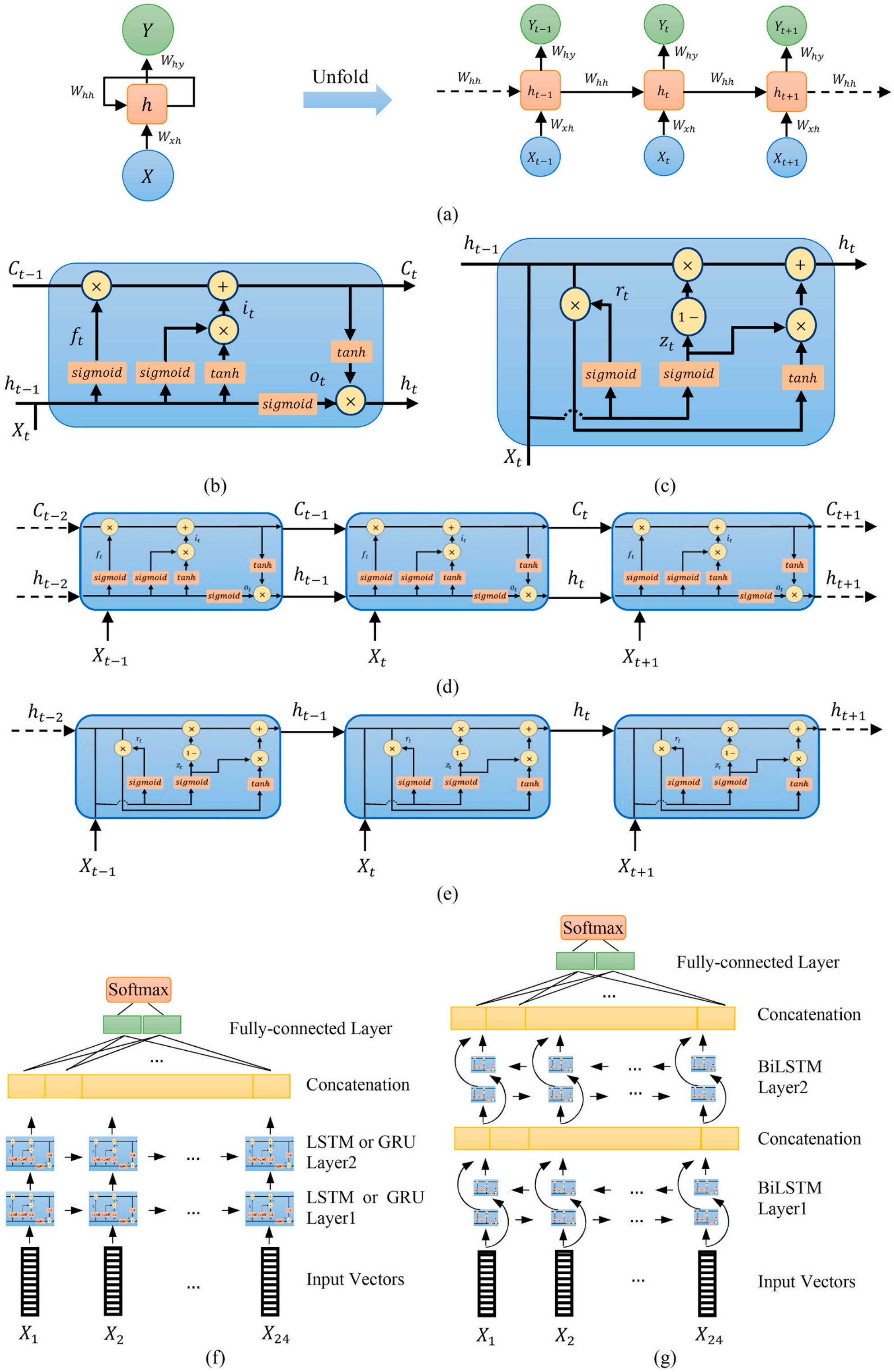

Fig. 3. (a) A general RNN structure, (b) an LSTM cell unit, (c) a GRU cell unit, (d) a chain of LSTM cells, (e) a chain of GRU cells, (f) the LSTM or GRU model implemented in this study, (g) the BiLSTM model implemented in this study.

ImageNet dataset, with 69.49% accuracy on the validation set.[3] Since it can extract general features from images, it was employed for our MRI dataset.

Some techniques are embedded into CNNs to avoid overfitting, such as max-pooling and drop-out layers. Max-pooling reduces the number of parameters, and subsequently, the dimension of extracted features to control overfitting and guides the invariance to scale, shift and rotation [48]. Drop-out layers randomly drop neurons at each update of the training phase and force neurons to act independently [26]. Another idea is to discard fully connected layers applied to networks such as SqueezeNet. Removing them results in a smaller number of learnable parameters compared to VGGNet and reduces overfitting. The overfitting issue is worst when applying 3D CNNs. For MRI analysis, the most straightforward method is to take the entire MRI volume as the input and build a deep 3D CNN. However, this requires training a large number of parameters, which simply causes overfitting [49]. In addition to these CNN-related structures, L1 and L2 regularisation have proven to prevent overfitting in the literature.

3.3. The RNN model

RNNs are a type of neural network with internal memory to model temporal dependencies in sequence-based tasks, such as video or text applications. In RNNs, past information is indirectly collected in hidden units, called state vectors. The output for the current input depends on the current input data and all previous input data using these state vectors. In the training process, hidden state vectors are updated accordingly. In Fig. 3(a), considering the sequence of extracted features from ResNet-18 to be X_1,X_2,X_3,…,X_{24}, the RNN takes X_1 from the input sequence and outputs h_1, which together with X_2, is the input for the next step. Similarly, h_2 with X_3 are the input for the next step. This process continues up to X_{24}, which leads to remembering the context while training. The output of this model at slice t can be formulated as $Y_t = W_{hy}h_t$, where $h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t)$. In these equations, W_{hh}, W_{xh}, and W_{hy} refer to weights in the previous hidden state, weights at the current input state and weights at the output state, respectively. The activation function *tanh* introduces nonlinearity to the model.

3.3.1. The LSTM model

LSTM networks are revised versions of RNNs. By resolving the vanishing gradient problem of RNNs [2], LSTM networks can more easily recall past data [40]. LSTM is suitable for sequence-based tasks, either fixed or variable length. They have a more complex structure than traditional RNNs. LSTM models contain three gate units (input, output and forget gates) and a memory cell unit. The cell unit remembers values over a sequence, and the three gates control the flow of information into and out of the cell. Gates are composed of nonlinear functions and a pointwise multiplication operation.

The input gate determines which values from the current input and the previous state must be utilised to modify the memory cell unit. A *sigmoid* function allows components to go through the model. A *tanh* function emphasises the passed values with weights according to their level of importance. In contrast to the input gate, the forget gate determines which information is discarded from memory by a *sigmoid* function. The output gate controls the input and memory of the LSTM unit to calculate the output by a *sigmoid* function. To formulate the gates and considering the input weights W, the recurrent weights R and the bias term b, we have:

$$\textit{Input gate} \quad i_t = sigmoid(W_iX_t + R_ih_{t-1} + b_i)\Theta\ tanh\left(W_gX_t + R_gh_{t-1} + b_g\right) \tag{1}$$

$$\textit{Forget gate} \quad f_t = sigmoid\left(W_gX_t + R_gh_{t-1} + b_g\right) \tag{2}$$

$$\textit{Output gate} \quad o_t = sigmoid(W_oX_t + R_oh_{t-1} + b_o) \tag{3}$$

at time step t, where Θ denotes the element-wise multiplication of vectors. The hidden state is defined by $h_t = o_t\ \Theta\ \tanh(c_t)$ and the cell state at step t is given by $c_t\ = f_t\ \Theta\ c_{t-1} + i_t$. An LSTM cell unit is shown in Fig. 3(b). A chain of LSTM cells is shown in Fig. 3(d) to illustrate their connection.

In our case study, an LSTM model with six layers was implemented, as shown in Fig. 3(f). The first layer was the input layer, which captured 24 vectors of features, each with a size of 512. Then, two LSTM layers were placed, each with 24 LSTM nodes; 24 was selected as the total so that the model could remember all of the sequence to make the decision. The first LSTM layer took the entire sequence of features as input to output a complete sequence. Similarly, the second LSTM layer captured the entire sequence from the previous layer as input to output the last step of the sequence. After concatenation, a fully connected layer was placed with two nodes because we have two classes in our classification task. Finally, a Softmax and classification layer determined the output of the entire LSTM model.

3.3.2. The BiLSTM model

LSTM models are unidirectional, meaning that the current output depends only on the current and previous inputs. A unidirectional model is required in real-time text-based or video-based applications since there is access only to the current and previous inputs. However, we had access to both past and future sequences of features extracted from our CNN model. This allowed us to use bidirectional sequence-based models. Use of BiLSTM will run inputs in two directions: from past to future (forwards) and from future to past (backwards). This way, the information will be preserved from both past and future. BiLSTM is simply a combination of two independent LSTMs, in which one receives the input in the forwards order and the other in the backwards order. The outputs of the two networks are usually concatenated at each time step to create the model output. In this study and for our BiLSTM network, a similar configuration to the LSTM network was used, as shown in Fig. 3(g). The main difference between the two models is that the BiLSTM network has twice the number of components—equal to 48—after the BiLSTM layer because of the concatenation.

3.3.3. The GRU model

The GRU is a new type of RNN that is similar to an LSTM model. By removing the cell state and using the hidden state to transfer information, it has a simpler structure than the LSTM. A GRU only has two gates: an update and a reset gate. Similar to the input and forget gates of an LSTM, the update gate determines which new information to keep and which information to discard. The reset gate decides how much past information to forget. To formulate the gates, and considering the input weights W, the recurrent weights R and the bias term b, we have:

$$\textit{Update gate} \quad z_t = sigmoid(W_zX_t + R_zh_{t-1} + b_z) \tag{4}$$

$$\text{Re}\textit{set gate} \quad r_t = sigmoid(W_rX_t + R_rh_{t-1} + b_r) \tag{5}$$

at time step t, where the hidden state is defined by $h_t\ = (1 - z_t)\ \Theta\ h_{t-1} + z_t\Theta tanh(W_hX_t\ + r_t\Theta(R_hh_{t-1})\ + b_h)$. A GRU cell unit is shown in Fig. 3(c). A chain of GRU cells is shown in Fig. 3(e) to illustrate their connection. In this study and for our GRU network, a similar configuration to the LSTM network was used, as shown in Fig. 3(f).

3.4. The TCN model

The core building blocks of a TCN are dilated causal convolutional layers that run over time steps of a sequence. In causal convolutions, a filter at time step t can only observe inputs that are no later than t; hence, there is no information leakage from future to past, similar to LSTM and GRU. To build a perspective from previous time steps, multiple convolutional layers are stacked on top of each other, as shown in Fig. 4(a). The dilation factor d in convolution layers controls the receptive field

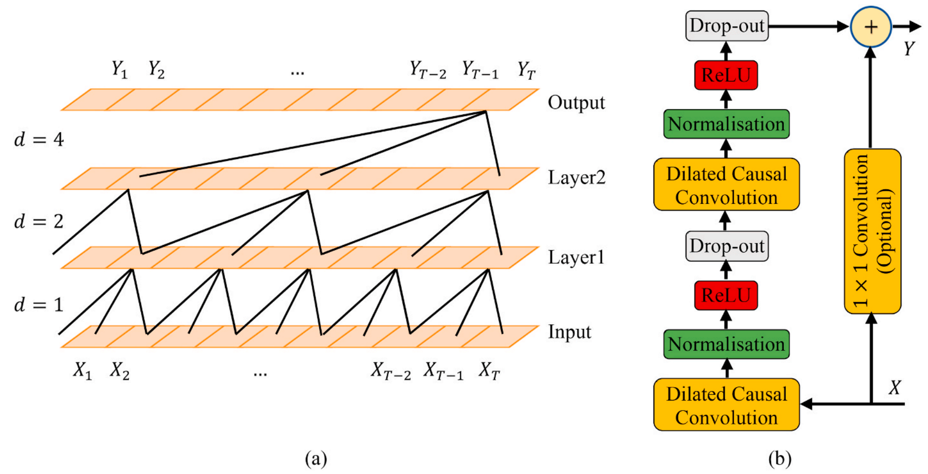

Fig. 4. (a) Stacked convolutional layers in TCNs, (b) a residual block.

size and can enable an exponentially large receptive field. The receptive field is a portion of sensory time steps that can activate neuronal responses. In simple causal convolutions, the receptive field grows linearly with every additional layer. A larger receptive field can help memorise long-term sequences, which leads to fewer layers and parameters in the TCN model.

Generally, the dilation factor of the K-th causal convolutional layer is assumed to be 2^{K-1} and the stride is 1. For the first layer, the receptive field size R_1 is equal to 1 as a causal convolution layer can always observe its current time step. For the next layer, we have $R_2 = 1 + (KernelSize - 1) \times 2$, where *KernelSize* is the size of filters in the causal convolution layer. For the K-th causal convolutional layer, we have $R_K = R_{K-1} + (KernelSize - 1) \times 2^{K-1}$, if the dilation factor increases exponentially by 2. If the stride is 1 and the kernel size is fixed for the whole model, we have $R_K = 1 + (KernelSize - 1) \times \sum_{k=1}^{K} 2^{k-1}$ or simply $R_K = 1 + (KernelSize - 1) \times (2^K - 1)$ for $K \geq 1$. By changing the filter size and number of layers, the receptive field size and number of learnable parameters are easily adjusted for any task.

A general TCN model consists of multiple residual blocks. A residual block stacks two dilated causal convolution layers together with the same dilation factor, followed by normalisation, ReLU activation and drop-out layers as shown in Fig. 4(b). The normalisation layer calculates the mean and variance of input data over each input channel and normalises it accordingly. In contrast to the batch normalisation layer, the mean and standard variance would be different for each observation in the mini-batch. The drop-out layer drops all time steps of a certain channel with the probability specified by the drop-out factor. After stacking multiple residual blocks, a fully connected and Softmax layer are connected for classification.

The input to each residual block is added to the output of the block. If the depth (number of channels) of the inputs and depth (number of filters) of the second dilated causal convolution layer differs, a 1 × 1 convolution is applied to the inputs before adding the convolution outputs to match the depths. Since each residual block has two identical dilated causal convolutions, the receptive field size for K-th residual block is calculated by $R_K = 1 + 2 \times (KernelSize - 1) \times (2^K - 1)$. By stacking several residual blocks together, TCNs can obtain a desirable receptive field size; however, it likely would not precisely match the maximum sequence length. By increasing the number of blocks, the receptive field can be larger than the maximum length and padding will be required. Otherwise, some older histories will be sacrificed.

We developed two approaches to our AD detection model. In *Approach 1*, the features extracted from our CNN model were used to train the TCN, similarly to the process used in RNN-based models. In this case, there was a sequence of 512-element vectors with a length of 24 for each subject. This way, the benefits of pre-trained CNN models could be utilised for feature extraction. However, the spatial relationship of pixels in 2D image slices may be lost after feature extraction. With $KernelSize = 3$, the receptive field size was 13 or 29 for two or three residual blocks, respectively. The feature extraction step remained independent of the classification step. In *Approach 2*, the original 72 greyscale coronal images of 79 × 79 were used directly to feed a TCN model. This way, feature extraction and classification were performed simultaneously, although filters were initialised randomly. With $KernelSize = 3$, the receptive field size was 61 or 125 for four or five residual blocks, respectively. In both approaches, each causal convolutional layer had 128 filters, the drop-out factor was 5% and the fully connected layer had two neurons.

4. Experimental results

In our experiments, 300, 50 and 100 MRI scans were utilised as training, validation and test sets, respectively. Every set contains the same number of subjects from each class to avoid imbalance. Therefore, the test set had 50 AD subjects and 50 NC subjects. We ensured that the test set remained completely unobserved and no information leaked from the test set into the training set. The same sets were used to train, validate and test all approaches—slice-based, voxel-based and sequence-based—and models (different types of 2D/3D CNNs, RNNs and TCNs) to obtain fair comparisons. The MATLAB deep-learning toolbox was employed to train and build the networks on a computer with an NVIDIA V100 GPU and 96 GB RAM.

Training a deep model requires the setting of various parameters in the backpropagation learning algorithm. Conventional values identified in a literature review [6] were applied for the backpropagation learning algorithm as the starting point. Further, an optimisation method employing Taguchi analysis was used to understand the effect of five parameters—the batch size, learning rate, drop-out factor, L2 regularisation factor and severity of data augmentation—in training deep models [19]. Inspired by these papers' findings, our experiment used trial and error to identify the optimal parameters for each approach. Our conclusions on each parameter's effect on classification performance indicate that a low level of data augmentation and a small mini-batch size negatively affect accuracy. Also, large values for the learning rate factor result in oscillation in classification accuracy at the end of the backpropagation algorithm, rather than convergence. For simplicity, similar training parameters, such as $learning\ rate = 0.01$, $mini\_batch\ size = 16$, and $momentum = 0.9$, were selected using trial and error for RNN-based models. Xavier [50] and orthogonal methods [51] were used to initialise input weights and recurrent weights, respectively.

For LSTM and BiLSTM models, the forget gate bias values were initialised with one and the remaining biases with zero. For the GRU model, all biases were initialised with zero. Weights of the fully connected layer were initialised with Xavier and biases with zeros. The stochastic gradient descent (SGD) optimiser was utilised for training; the maximum number of epochs was 50.

For TCN models, training parameters, such as *learning rate* = 0.001 and *momentum* = 0.9, were selected with the SGD optimiser. Weights for causal convolutional and fully connected layers were initialised randomly with Gaussian distribution. In *Approach 1*, we had *mini_batch size* = 200, which means the entire training set was used to update weights. In this approach, since the extracted features from ResNet-18 were discriminative, convergence occurred quickly, with the maximum number of epochs equal to 20. In *Approach 2*, we had *mini_batch size* = 4 because of the computational resources needed for input images. The maximum number of epochs was 350 since time was required for the model to learn feature extraction from scratch.

Input training subjects were shuffled at the beginning of each epoch. With shuffling, models observe subjects in a different order, but the order of extracted features or images in each sequence was preserved. Data augmentation was used because the number of patients was not enough to train deep models. Data augmentation method is a process that increases data diversity to train models without gathering new data. In all models, random ± 5% scaling and ± 5 pixel translation was performed on the training set only. Hence, in each iteration of the training process, coronal images were modified using data augmentation. For the RNN-based models and TCN model *Approach 1*, features were recalculated in ResNet-18, accordingly. In TCN *Approach 2*, augmented images were directly used for training. Consequently, in every epoch, the models observed slightly modified images or features in a different order.

As previously discussed, a ResNet-18 was used to extract features for recurrent models (LSTM, BiLSTM and GRU) and *Approach 1* of the TCN model. Fig. 5(a) shows 24 MRI coronal slices of one subject, and Fig. 5(b) shows feature maps of the first convolutional layer of ResNet-18 for a mid-coronal MRI slice. Fig. 6 shows the 64 filters of the first convolutional layer of this model. ResNet-18 delivered a vector with 512 elements for each MRI slice extracted immediately before its fully connected layer. To obtain insights from the extracted features, feature maps of the mid-coronal slice of all AD subjects in our dataset are shown in Fig. 7. The bright pixels refer to activated neurons and each activated neuron reflects a single feature for the input image. As shown in the magnified part of Fig. 7, most AD subjects activate particular neurons in ResNet-18. The responsibility of sequence-based models in this study is to understand the relationship between the activated neurons of one subject (24 vectors, each with 512 elements) to determine whether the patient suffers from AD.

We wanted to determine whether sequence-based deep models can improve the classification rate of 2D CNNs. Therefore, to calculate the accuracy, the 2D CNN models discussed in the literature were adjusted; a fully connected layer of two outputs (for AD vs. NC) with a weight/bias learning rate factor equal to 0.003 replaced the last fully connected layer. The learning rate of all other layers was equal to 0.0003. Setting a higher learning rate for the new fully connected layer enabled faster training than that of previous layers, which were trained on MNIST (for LeNet-5) or ImageNet. Other training parameters included L2 regularisation of 0.0005 and a mini-batch size of 64. The optimiser was SGD, with a momentum of 0.9. The same dataset partitions with sequence-based models and the same augmentation transforms (random ± 5% scaling and ± 5 pixel translation) were performed during training. The maximum number of epochs was 100; to avoid overfitting, early stopping was considered to stop the training process if validation accuracy did not improve after 20 consecutive epochs. The same training parameters were used for all 2D/3D CNNs; however, a mini-batch size of 8 was used for 3D CNNs because of the available computational resources. To create 3D CNNs, 2D filters of 2D CNN models were expanded to have 3D filters. The number of learnable parameters increased because of the extension of filters' dimensions. Any other layers in the structure of CNN were adjusted according to the new filters. Each 2D CNN model made a slice-based decision for each MRI coronal slice and delivered a single decision for each subject, with a majority voting strategy. For 3D CNNs, only one decision was made for each subject.

The choice of a dataset may significantly affect the results of different models found in the literature. Given the diverse datasets used and the different number of subjects, or even dissimilar subjects, it is unreasonable to compare various reported results. Even for studies on identical datasets, with matching subject counts and subject number identification codes, reported results are still incomparable because researchers might have used a different portion of subjects in the training and test sets. In a recently conducted systematic literature review on AD detection using deep learning [6], a total of 114 papers in this field of research were evaluated. As part of the literature review, the reported accuracy and the utilised dataset of each method is presented in the paper's Appendix[4].

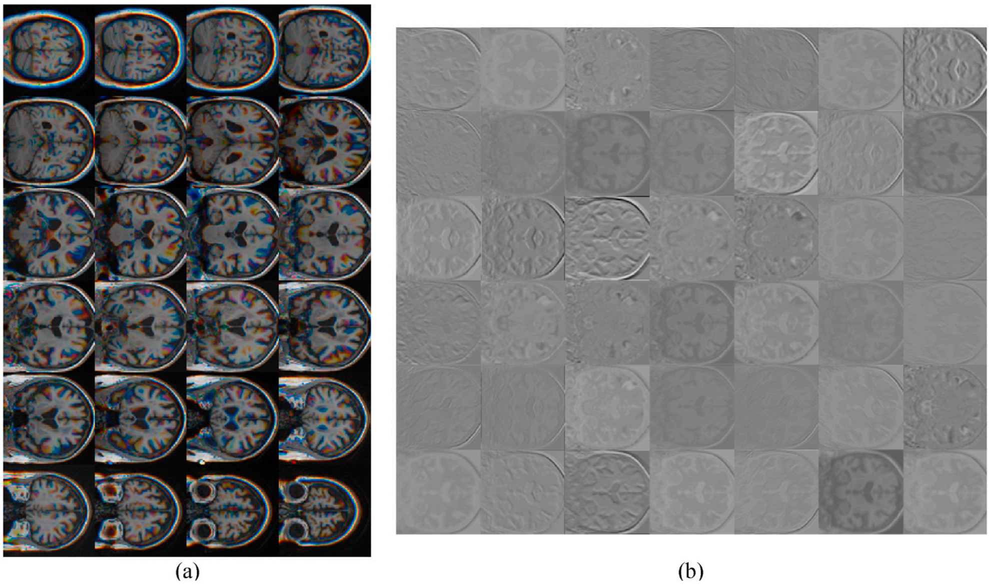

Fig. 5. (a) 24 MRI coronal slices of one subject, (b) Feature maps of the first convolutional layer of ResNet-18 for a mid-coronal MRI slice.

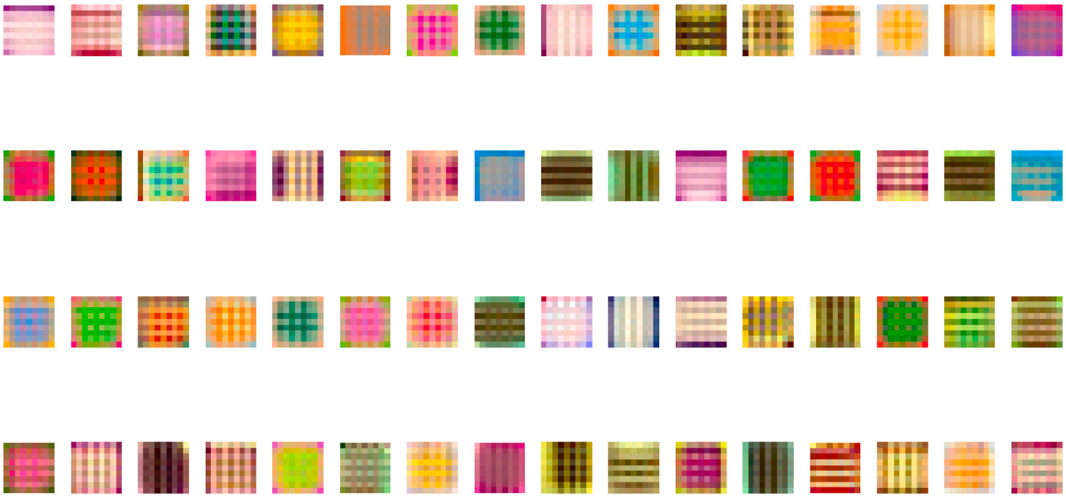

Fig. 6. The 64 filters of the first convolutional layer of ResNet-18.

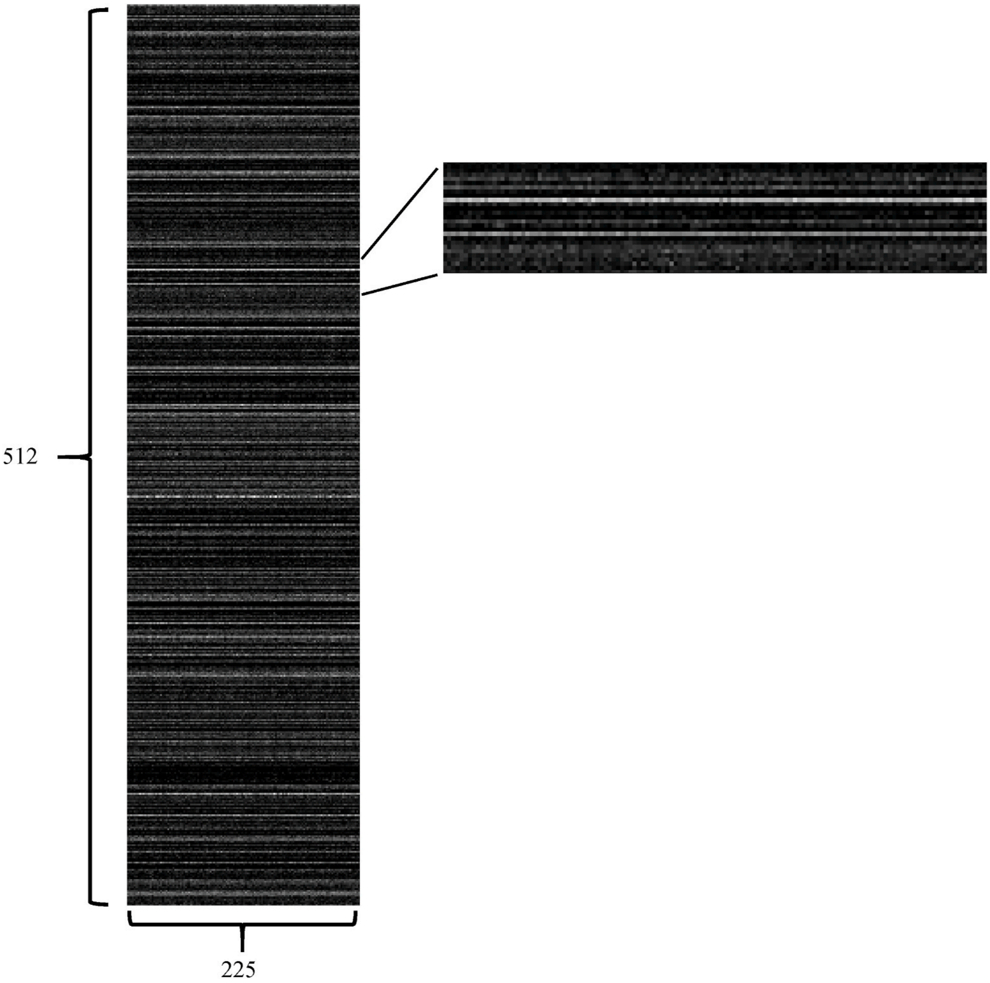

Fig. 7. Feature vectors of the mid-coronal slice of all AD subjects in our dataset.

In this study, several models from the literature were implemented to enable a fair comparison. The number of learnable parameters for each implemented model is listed in Table 2, together with other specifications, such as the depth and number of layers. The large number of learnable parameters associated with the depth and number of fully connected layers could affect overfitting significantly. After repeating several experiments, the highest accuracies of various CNNs reported in the literature, with or without transfer learning, on our selected datasets (AD vs. NC) and under the training conditions described earlier, are shown in Table 3. Xavier's initialisation method [50] was used to initialise our CNN models, while training from scratch. In this table, accuracy refers to the percentage of correctly classified test subjects.

Table 2
CNN models implemented in our study.

Network		Depth	#Layers	#Convolutional layers	#FCs	#Parameters (Millions)
2D	LeNet-5	5	16	3	2	0.062
	AlexNet	8	25	5	3	61.0
	VGG-16	16	41	13	3	138
	SqueezeNet	18	68	27	0	1.24
	ResNet-18	18	71	20	1	11.7
	VGG-19	19	47	16	3	144
	GoogLeNet	22	144	58	1	7.0
	Inceptionv3	48	315	94	1	23.9
	ResNet-50	50	177	53	1	25.6
	ResNet-101	101	347	104	1	44.6
3D	LeNet-5	5	16	3	2	0.26
	ResNet-18	18	71	20	1	34
	ResNet-50	50	177	53	1	48

Table 3
Accuracies of the CNNs discussed in the literature on our selected dataset (AD vs. NC).

Model		Results (%)		
		Accuracy	Sensitivity	Specificity
Training from Scratch	LeNet-5 [32,52–54]	77	96	58
	AlexNet [55]	78	86	70
	VGGNet-16	79	78	80
	SqueezeNet	78	76	80
	ResNet-18	80	78	82
	VGGNet-19	80	78	82
	GoogLeNet [27,32, 54,55]	76	88	64
	Inceptionv3 [36]	77	82	72
	ResNet-50	81	80	82
	ResNet-101	78	76	80
	3D LeNet-5	81	78	85
	3D ResNet-18	69	72	66
	3D ResNet-50	63	68	58
Transfer Learning	LeNet-5 [36]	78	96	60
	AlexNet [36]	80	86	74
	VGGNet-16 [26,33, 34]	81	80	82
	SqueezeNet	81	82	80
	ResNet-18 [28]	82	84	80
	VGGNet-19	81	78	85
	GoogLeNet [29,31]	81	86	76
	Inceptionv3 [30,33, 56]	78	80	76
	ResNet-50	81	80	82
	ResNet-101	79	76	82

Table 4
Accuracies of the proposed sequence-based deep models on our selected dataset (AD vs. NC).

Model		Results (%)		
		Accuracy	Sensitivity	Specificity
Approach 1	ResNet-18	82	84	80
	ResNet-18 + LSTM	84	80	88
	ResNet-18 + BiLSTM	79	82	76
	ResNet-18 + GRU	82	70	94
	ResNet-18 + TCN (with 2 residual blocks)	82	72	92
	ResNet-18 + TCN (with 3 residual blocks)	88	94	82
Approach 2	TCN (with 4 residual blocks)	91	94	88
	TCN (with 5 residual blocks)	78	82	74

Sensitivity refers to the percentage of evaluated test subjects suffering from AD who were correctly classified as such, while specificity is the percentage of evaluated healthy test subjects correctly classified as healthy. The ResNet-18 model achieved 82% accuracy, 84% sensitivity and 80% specificity on our dataset. Inspired by the novel concept of shortcut connections in ResNet models to avoid the vanishing gradient problem, ResNet-18 was used to extract features for our proposed sequence-based models. The reported accuracies in Table 3 for 3D CNNs confirm the overfitting issue caused by a large number of learnable parameters.

The performances of our proposed sequence-based models on the same dataset are listed in Table 4. In ResNet-18 + LSTM, ResNet-18 + BiLSTM, ResNet-18 + GRU and ResNet-18 + TCN, the idea was to use a sequence-based model on top of ResNet-18 to improve the accuracy of AD detection. Therefore, instead of having ResNet-18 to make a slice-based decision for each MRI coronal slice, extracted features were used to train the sequence-based models. Thus, each sequence-based model received a sequence of features corresponding to a sequence of MRI coronal slices of one subject. The sequence-based models consider the extracted features and identify relationships between the sequence of features.

Conversely, TCNs can also be practical without the need for an interface to extract features. They can extract features from MRI coronal slices and simultaneously determine their relationships. As shown in Table 4, sequence-based models can improve the accuracy of AD detection. TCN with 4 residual blocks in *Approach 2*, ResNet-18 + TCN with 3 residual blocks in *Approach 1* and ResNet-18 + LSTM achieved greater accuracy than ResNet-18 itself. However, the results for ResNet-18 + BiLSTM show that adding a sequence-based network for independently extracted features sometimes led to ambiguities in the sequence-based classification.

For a sequence of 2D MRI slices in RNN-based models, each LSTM cell preserves only the past images' information because the only inputs it has observed are from the past. Every learnable parameter in an LSTM cell is updated according to the current and previous images in a sequence of MRI slices. In contrast, the learning algorithm of BiLSTMs is fed with MRI slices once from the beginning to the end and once from the end to the beginning. Hence, every learnable parameter in a BiLSTM cell is updated according to the current, previous and future images in a sequence of MRI slices. The mechanism of the other RNN model, GRU, is similar to that of LSTM, with a simpler structure and faster training. The learnable parameters in LSTM, BiLSTM and GRU cells control the flow of information from MRI to the models—information that should be memorised or forgotten. There are debates over the performance of each type of RNN model for different applications in the literature. Our results show that, for AD detection using features extracted by ResNet-18, the LSTM model yields a better classification performance than BiLSTM and GRU.

In TCN *Approach 1*, features were extracted by ResNet-18 and the TCN model was used to understand the relationships between extracted features. Using two residual blocks limited the size of the input sequence and forced us to remove some feature vectors from our sequence of features from an MRI scan. Using three residual blocks forced us to pad vectors with zeros as their elements. In *Approach 2*, feature extraction

Table 5
A summary of the implemented input data management methods for AD detection.

Methods	Strengths	Limitations
Sliced-based	• Prevents facing millions of parameters during training and provides more simplified networks • Can benefit from transfer learning	• Loses spatial dependencies in adjacent image slices
Voxel-based	• Can capture the 3D information of a brain scan	• Involves high computation load and high feature dimensionality • Cannot benefit from transfer learning
Sequence-based	• Prevents facing millions of parameters during training and provides more simplified networks • Can benefit from transfer learning using 2D datasets[a] • Keeps spatial dependencies in adjacent slices[b]	• Feature extraction and classification are not performed simultaneously[a] • Cannot benefit from transfer learning using 2D datasets[b]

[a] Only 2D CNN + RNN and 2D CNN + TCN.

[b] Only TCN *Approach 2*.

and classification were directly conducted by the TCN model. Using four residual blocks limited the size of the input sequence and required removal of some MRI image slices from an MRI scan.

Similarly, using five residual blocks required padding image slices with zeros as pixel values. Comparing *Approaches 1* and *2*, the direct feature extraction and classification performed by the TCN in *Approach 2* achieved higher accuracy of AD detection. Additionally, this special feature of TCNs provided an advantage over RNNs for image-based tasks. In our experiments, a TCN with four residual blocks achieved the best classification performance of the tested models, with 91% accuracy, 94% sensitivity and 88% specificity. For generalisation purposes, 5-fold cross-validation was also applied, which resulted in 91.78% accuracy, 91.56% sensitivity and 92% specificity. These results for AD detection are approximately 10% better than those obtained using ResNet-18.

Employing 2D MRI slices as the input as an alternative to entire 3D MRI scans avoids confrontation of millions of learnable parameters and leaves more simplified networks, at the cost of losing spatial dependency between neighbouring slices. In contrast, voxel-based methods can obtain all 3D information in a single brain scan. However, voxel-based methods involve high computational load and high feature dimensionality. To solve the high feature dimensionality, voxel preselection techniques may be desired. This paper presents the idea of deep sequence-based models. They benefit from transfer learning, maintain spatial dependencies in adjacent slices and avoid confrontation of millions of parameters during training. In 2D CNN + RNN and 2D CNN + TCN, feature extraction and classification are not performed simultaneously, and TCN *Approach 2* cannot benefit from transfer learning using 2D datasets. A summary of the strengths and limitations of all three input data management methods is provided in Table 5.

5. Conclusion

2D CNNs can extract features from MRI slices and directly feed them into a fully connected layer and a Softmax layer for AD detection. On the other hand, 3D CNNs extract features from the whole MRI volumes for classification. The first scenario neglects temporal dependencies for the sequence of 2D MRI slices from a 3D MRI volume of a subject. The second scenario has many learnable parameters to train. In this paper, deep sequence-based models were proposed for AD detection. In these models, a sequence of features extracted by a pre-trained ResNet-18 from MRI scans was used to train a TCN and different RNNs, such as the LSTM, BiLSTM and GRU. Sequence-based models can capture features extracted from 2D MRI slices of one subject and understand the relationship between a sequence of features related to that subject.

LSTM, BiLSTM and GRU models are revised versions of RNNs with more complex structures, including gates. Gates can regulate the flow of information passing through the sequence chain and remove or add past information from the network. Among the three types of RNNs compared in this study, ResNet-18 + LSTM achieved the best classification accuracy for AD detection, with 84% accuracy, 80% sensitivity and 88% specificity.

In contrast to RNN-based models and TCN *Approach 1*, which needed features extracted from ResNet-18, TCN models can merge the feature extraction and classification stages into a single step. TCNs can understand and model spatial dependencies of an MRI slice and temporal dependencies of adjacent slices simultaneously using convolutional filters. To the best of our knowledge, this is the first time TCNs have been applied to 3D medical data (spatial data) instead of temporal data. In this study, TCNs performed better than slice-based methods (using 2D CNNs), voxel-based methods (using 3D CNNs) and RNN-based methods, with 91.78% accuracy, 91.56% sensitivity and 92% specificity.

Of the three approaches considered in this study, sequence-based models showed the best AD detection performance, compared to slice-based and voxel-based approaches. Slice-based approaches do not need to deal with a huge number of parameters during training, are more simplified networks, and can benefit from transfer learning; however, the spatial dependencies in adjacent image slices are lost. Voxel-based approaches can capture the 3D information of a brain scan, but involve high computation loads and high feature dimensions. Also, they cannot benefit from transfer learning. Our proposed sequence-based approaches avoid the need to deal with a huge number of parameters during training, similar to slice-based approaches. The proposed 2D CNN + RNN and 2D CNN + TCN combinations can benefit from transfer learning using 2D datasets, although feature extraction and classification are not performed simultaneously. On the other hand, TCN *Approach 2* keeps spatial dependencies in adjacent slices but cannot benefit from transfer learning using 2D datasets.

The advantages of employing TCNs include greater control of the receptive field size and improved parallel processing. The parameter settings for TCNs depend mainly on the receptive field applied to input data. The receptive field can be made larger by increasing the number of blocks than the maximum sequence length, and padding will be required. Otherwise, older histories will be sacrificed. Although TCNs are similar to 3D CNNs, they require less computational resources for training and have a selectable receptive field size.

The combined models in this paper can be applied to any other image-based tasks (e.g. video processing). For example, a CNN model can extract features from video frames and shape a feature vector. Then, the RNN or TCN can perform sequence-based classification. Otherwise, similar to the TCN *Approach 2* in this paper, a video can be fully processed by one TCN model. In our experiments, we initialised TCN weights with random Gaussian distribution. Future studies can improve the TCN structure using weights delivered by pre-trained CNNs.

Declaration of competing interest

All authors declare no conflict of interest.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir, Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Lumosity; Lundbeck; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research is providing funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (www.fnih.org). The grantee organisation is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compbiomed.2021.104537.

References

[1] G. Litjens, et al., A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88, https://doi.org/10.1016/j.media.2017.07.005.

[2] M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: overview, challenges and the future. Classification in BioApps, Springer, 2018, pp. 323–350.

[3] Alzheimer's Association, Alzheimer's disease facts and figures, Alzheimer's Dementia 16 (3) (2020) 1–94.

[4] Dementia Australia, The Dementia Guide for People Living with Dementia, Their Families and Carers, 2020.

[5] Australian Bureau of Statistics, Causes of Death, 2019. Australia.

[6] A. Ebrahimighahnavieh, S. Luo, R. Chiong, Deep learning to detect Alzheimer's disease from neuroimaging: a systematic literature review, Comput. Methods Progr. Biomed. 187 (Apr 2020) 105242, https://doi.org/10.1016/j.cmpb.2019.105242.

[7] K. Gunawardena, R. Rajapakse, N. Kodikara, Applying convolutional neural networks for pre-detection of Alzheimer's disease from structural MRI data, in: Proceedings of the 24th International Conference on Mechatronics and Machine Vision in Practice, 2017, pp. 1–7, https://doi.org/10.1109/M2VIP.2017.8211486.

[8] C. Feng, A. Elazab, P. Yang, T. Wang, B. Lei, X. Xiao, 3D convolutional neural network and stacked bidirectional recurrent neural network for Alzheimer's disease diagnosis, in: Proceedings of the International Workshop on Predictive Intelligence in Medicine, 2018, pp. 138–146.

[9] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[10] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Network. 18 (5–6) (2005) 602–610.

[11] K. Cho, et al., Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation, 2014 arXiv preprint arXiv:1406.1078.

[12] S. Bai, J.Z. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018 arXiv preprint arXiv:1803.01271.

[13] Y. LeCun, et al., Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (Win 1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.

[14] O. Russakovsky, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (Dec 2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[16] C. Yang, A. Rangarajan, S. Ranka, Visual explanations from deep 3D convolutional neural networks for Alzheimer's disease classification, in: Proceedings of the AMIA Annual Symposium, 2018, pp. 1571–1580.

[17] H. Karasawa, C.-L. Liu, H. Ohwada, Deep 3D convolutional neural network architectures for Alzheimer's disease diagnosis, in: Proceedings of the Asian Conference on Intelligent Information and Database Systems, 2018, pp. 287–296, https://doi.org/10.1007/978-3-319-75417-8_27.

[18] H. Tang, E. Yao, G. Tan, X. Guo, A fast and accurate 3D fine-tuning convolutional neural network for Alzheimer's disease diagnosis, in: Proceedings of the International CCF Conference on Artificial Intelligence, 2018, pp. 115–126, https://doi.org/10.1007/978-981-13-2122-1_9.

[19] A. Ebrahimi, S. Luo, R. Chiong, Introducing transfer learning to 3D ResNet-18 for Alzheimer's disease detection on MRI images, in: Proceedings of the 35th International Conference on Image and Vision Computing New Zealand, 2020, pp. 1–6, https://doi.org/10.1109/IVCNZ51579.2020.9290616.

[20] A. Ebrahimi, S. Luo, A.s.D.N. Initiative, Convolutional neural networks for Alzheimer's disease detection on MRI images, J. Med. Imaging 8 (2) (2021), 024503.

[21] J. Islam, Y. Zhang, Deep convolutional neural networks for automated diagnosis of Alzheimer's disease and mild cognitive impairment using 3D brain MRI, in: Proceedings of the International Conference on Brain Informatics, 2018, pp. 359–369, https://doi.org/10.1007/978-3-030-05587-5_34.

[22] J.M. Ortiz-Suárez, R. Ramos-Pollán, E. Romero, Exploring Alzheimer's anatomical patterns through convolutional networks, in: Proceedings of the 12th International Symposium on Medical Information Processing and Analysis, vol. 10160, 2017, 10160Z, https://doi.org/10.1117/12.2256840.

[23] S. Luo, X. Li, J. Li, Automatic Alzheimer's disease recognition from MRI data using deep learning method, J. Appl. Math. Phys. 5 (9) (2017), https://doi.org/10.4236/jamp.2017.59159.

[24] G. Awate, S. Bangare, G. Pradeepini, S. Patil, Detection of Alzheimers Disease from MRI Using Convolutional Neural Network with Tensorflow, 2018 arXiv preprint arXiv:1806.10170.

[25] S.-H. Wang, P. Phillips, Y. Sui, B. Liu, M. Yang, H. Cheng, Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max Pooling, J. Med. Syst. 42 (5) (2018) 85, https://doi.org/10.1007/s10916-018-0932-7.

[26] C.D. Billones, O.J.L.D. Demetria, D.E.D. Hostallero, P.C. Naval, DemNet: a convolutional neural network for the detection of Alzheimer's disease and mild cognitive impairment, in: Proceedings of the IEEE Region 10 Conference, 2016, pp. 3724–3727, https://doi.org/10.1109/TENCON.2016.7848755.

[27] A. Farooq, S. Anwar, M. Awais, S. Rehman, A deep CNN based multi-class classification of Alzheimer's disease using MRI, in: Proceedings of the IEEE International Conference on Imaging Systems and Techniques, 2017, pp. 1–6, https://doi.org/10.1109/IST.2017.8261460.

[28] A. Valliani, A. Soni, Deep residual nets for improved Alzheimer's diagnosis, in: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, 2017, https://doi.org/10.1145/3107411.3108224.

[29] A. Farooq, S. Anwar, M. Awais, M. Alnowami, Artificial intelligence based smart diagnosis of Alzheimer's disease and mild cognitive impairment, in: Proceedings of the Smart International Cities Conference, 2017, pp. 1–4, https://doi.org/10.1109/ISC2.2017.8090871.

[30] V. Wegmayr, D. Haziza, Alzheimer classification with MR images: exploration of CNN performance factors, in: Proceedings of 1st Conference on Medical Imaging with Deep Learning, MIDL, 2018, pp. 1–7.

[31] C. Wu, et al., Discrimination and conversion prediction of mild cognitive impairment using convolutional neural networks, Quant. Imag. Med. Surg. 8 (18) (2018) 992–1003, https://doi.org/10.21037/qims.2018.10.17.

[32] S. Sarraf, G. Tofighi, Classification of Alzheimer's disease structural MRI data by deep learning convolutional neural networks, arXiv preprint arXiv:1607.06583, 2016.

[33] M. Hon, N. Khan, Towards Alzheimer's disease classification through transfer learning, in: Proceedings of IEEE International Conference on Bioinformatics and Biomedicine, 2017, pp. 1166–1169, https://doi.org/10.1109/BIBM.2017.8217822.

[34] R. Jain, N. Jain, A. Aggarwal, D.J. Hemanth, Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images, Cognit. Syst. Res. 57 (2019) 147–159, https://doi.org/10.1016/j.cogsys.2018.12.015.

[35] J. Islam, Y. Zhang, Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks, Brain Informatics 5 (2) (2018) 1–14, https://doi.org/10.1186/s40708-018-0080-3.

[36] A. Ebrahimi-Ghahnavieh, S. Luo, R. Chiong, Transfer learning for Alzheimer's disease detection on MRI images, in: Proceedings of the IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, 2019, pp. 133–138, https://doi.org/10.1109/ICIAICT.2019.8784845.

[37] X. Zhang, Y. Yang, T. Li, Y. Zhang, H. Wang, H. Fujita, CMC: a consensus multi-view clustering model for predicting Alzheimer's disease progression, Comput. Methods Progr. Biomed. 199 (2020) 105895.

[38] D. Cheng, M. Liu, Combining convolutional and recurrent neural networks for Alzheimer's disease diagnosis using PET images, in: Proceedings of the IEEE International Conference on Imaging Systems and Techniques, 2017, pp. 1–5, https://doi.org/10.1109/IST.2017.8261461.

[39] M. Liu, D. Cheng, W. Yan, Classification of Alzheimer's disease by combination of convolutional and recurrent neural networks using FDG-PET images, Front. Neuroinf. 12 (2018), https://doi.org/10.3389/fninf.2018.00035.

[40] R. Cui, M. Liu, G. Li, Longitudinal analysis for Alzheimer's disease diagnosis using RNN, in: Proceedings of the IEEE 15th International Symposium on Biomedical Imaging, 2018, pp. 1398–1401.

[41] G. Lee, K. Nho, B. Kang, K.-A. Sohn, D. Kim, Predicting Alzheimer's disease progression using multi-modal deep learning approach, Sci. Rep. 9 (1) (2019). Art no. 1952.

[42] L. Gao, et al., Brain disease diagnosis using deep learning features from longitudinal MR images, in: Proceedings of the Joint International Conference on Web and Big Data Asia-Pacific Web (APWeb) and Web-Age Information Management, WAIM), 2018, pp. 327–339, https://doi.org/10.1007/978-3-319-96890-2_27.

[43] M. Liu, D. Cheng, K. Wang, Y. Wang, A.s.D.N. Initiative, Multi-modality cascaded convolutional neural networks for Alzheimer's disease diagnosis, Neuroinformatics 16 (3–4) (Oct 2018) 295–308, https://doi.org/10.1007/s12021-018-9370-4.

[44] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (2017) 221–248, https://doi.org/10.1146/annurev-bioeng-071516-044442.

[45] P.V. Rouast, M. Adam, R. Chiong, Deep learning for human affect recognition: insights and new developments, IEEE Transactions on Affective Computing 12 (2) (2021) 524–543, https://doi.org/10.1109/TAFFC.2018.2890471.

[46] V. Fonov, et al., Unbiased average age-appropriate atlases for pediatric studies, Neuroimage 54 (1) (2011) 313–327, https://doi.org/10.1016/j.neuroimage.2010.07.033.

[47] W.D. Penny, K.J. Friston, J.T. Ashburner, S.J. Kiebel, T.E. Nichols, Statistical Parametric Mapping: the Analysis of Functional Brain Images, Elsevier, 2011.

[48] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition. Artificial Neural Networks–ICANN, Springer, 2010, pp. 92–101.

[49] M. Liu, D. Cheng, K. Wang, Y. Wang, A.s.D.N. Initiative, Multi-Modality Cascaded Convolutional Neural Networks for Alzheimer's Disease Diagnosis, Neuroinformatics, 2018, pp. 1–14.

[50] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256, in: http://proceedings.mlr.press/v9/glorot10a.

[51] A.M. Saxe, J.L. McClelland, S. Ganguli, Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks, 2013 arXiv preprint arXiv:1312.6120.

[52] S. Sarraf, G. Tofighi, Classification of Alzheimer's Disease Using fMRI Data and Deep Learning Convolutional Neural Networks, 2016 arXiv preprint arXiv:1603.08631.

[53] S. Sarraf, G. Tofighi, Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data, in: Proceedings of the Future Technologies Conference, 2016, pp. 816–820.

[54] S. Sarraf, G. Tofighi, DeepAD: Alzheimer's Disease Classification via Deep Convolutional Neural Networks Using MRI and fMRI, 2016, https://doi.org/10.1101/070441 bioRxiv 070441, p. 070441.

[55] Y. Kazemi, S.K. Houghten, A deep learning pipeline to classify different stages of Alzheimer's disease from fMRI data, in: Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, 2018, pp. 1–8, https://doi.org/10.1109/CIBCB.2018.8404980.

[56] J. Islam, Y. Zhang, A novel deep learning based multi-class classification method for Alzheimer's disease detection using brain MRI data, in: Proceedings of the International Conference on Brain Informatics, 2017, pp. 213–222, https://doi.org/10.1007/978-3-319-70772-3_20.