Turkish Journal of Science & Technology
19(1), 223-235, 2024

Research Paper
https://doi.org/10.55525/tjst.1396312

Use of 3D-CAPSNET and RNN models for 4D fMRI-based Alzheimer's Disease Pre-detection

Ali İSMAİL[1], Gonca Gökçe MENEKŞE DALVEREN [2*]
[1] Graduate School of Natural and Applied Sciences, Atilim University, Ankara, Turkiye
[2] Software Engineering Department, Engineering Faculty, Atilim University, Ankara, Turkiye
[1] alykotbb95@gmail.com, [*2] gonca.menekse@atilim.edu.tr

(Geliş/Received: 27/11/2023; Kabul/Accepted: 15/03/2024)

Abstract: Predicting Alzheimer's disease (AD) at an early stage can assist more successfully prevent cognitive decline. Numerous investigations have focused on utilizing various convolutional neural network (CNN)-based techniques for automated diagnosis of AD through resting-state functional magnetic resonance imaging (rs-fMRI). Two main constraints face the methodologies presented in these studies. First, overfitting occurs due to the small size of fMRI datasets. Second, an effective modeling of the 4D information from fMRI sessions is required. In order to represent the 4D information, some studies used the deep learning techniques on functional connectivity matrices created from fMRI data, or on fMRI data as distinct 2D slices or 3D volumes. However, this results in information loss in both types of methods. In order to model the spatiotemporal (4D) information of fMRI data for AD diagnosis, a new model based on the capsule network (CapsNet) and recurrent neural network (RNN) is proposed in this study. To assess the suggested model's effectiveness, experiments were run. The findings show that the suggested model could classify AD against normal control (NC) and late mild cognitive impairment (lMCI) against early mild cognitive impairment (eMCI) with accuracy rates of 94.5% and 61.8%, respectively.

Key words: Alzheimer's disease preliminary diagnosis, machine learning, magnetic resonance imaging, convolutional neural network, recurrent neural network.

4B fMRI Tabanlı Alzheimer Hastalığının Ön Tespiti için 3B-CAPSNET ve RNN Modellerinin Kullanılması

Öz: Alzheimer hastalığının (AH) ilerlemesinin erken tahmini, bilişsel gerilemenin daha etkili bir şekilde yavaşlatılmasına yardımcı olabilmektedir. Dinlenme durumu fonksiyonel manyetik rezonans görüntüleme (dd-fMRG) kullanılarak otomatik AH tanısı için evrişimli sinir ağlarına (ESA) dayalı farklı yöntemlerin uygulanmasına yönelik çeşitli çalışmalar yapılmıştır. Bu çalışmalarda tanıtılan yöntemler iki büyük zorlukla karşılaşmaktadır. Birincisi, fMRG veri kümeleri küçük boyutta olduğundan aşırı uyum gözlemlenebilmektedir. İkincisi, fMRG oturumlarının 4 boyutlu (4B) bilgilerinin verimli bir şekilde modellenmesi gerekmektedir. Çalışmalardan bazıları, derin öğrenme yöntemlerini, 4B bilgiyi modellemek için fMRG verilerinden oluşturulan fonksiyonel bağlantı matrislerine veya ayrı 2B dilimler veya 3B hacimler olarak fMRG verilerine uygulamıştır. Ancak bu durumun her iki yöntem türünde de bilgi kaybına neden olduğu gözlemlenmiştir. Bu çalışmada, AD tanısı için fMRG verilerinin uzay-zamansal (4B) bilgilerini modellemek amacıyla Kapsül ağı (CapsNet) ve tekrarlayan sinir ağını (RNN) temel alan yeni bir model önerilmektedir. Önerilen modelin etkinliğini değerlendirmek için deneyler yapılmıştır. Sonuçlara göre, önerilen modelin AH'na karşı normal kontrol (NK) ve geç hafif bilişsel bozukluk (GHBB) ile erken hafif bilişsel bozukluk (EHBB) sınıflandırma görevlerinde sırasıyla %94.5 ve %61.8 doğruluk elde edebildiği görülmüştür.

Anahtar kelimeler: Alzheimer hastalığı tespiti, makine öğrenmesi, manyetik rezonans görüntüleme, evrişimli sinir ağları, yinelemeli sinir ağı.

1. Introduction

1.1 Preamble

Alzheimer's disease (AD) is the sixth cause of death among older people in the United States [1]. It is expected that the over-60 population will become 2 billion by 2050 which will be 21% of the world's population after it was 8% in 1950 [2]. This reveals a massive increase in the number of people affected by Alzheimer's disease estimated to be 100 million by 2050 [3]. AD is the most common type of dementia, it is a neurodegenerative disease that occurs as a result of changes in amyloid and tau protein levels causing a disability of communication between neurons and cell death in different brain regions resulting in the loss of cognitive abilities. Several brain regions can be damaged due to AD as well as a reduction in some neurotransmitters. As a result, the functionality for which a certain region or a neurotransmitter is responsible for is affected. For example, the hippocampus is responsible for learning and memory, and it is more likely to get damaged before other regions. Also, Acetylcholine (ACh), a neurotransmitter responsible for memory and learning is decreased in concentration. Other

[*] Corresponding author: gonca.menekse@atilim.edu.tr. ORCID Number of authors: [1] 0000-0003-3614-114X, [2] 0000-0002-8649-1909

regions that are affected by AD include the left medial orbital part of the superior frontal gyrus, left caudate nucleus, right middle frontal gyrus, left orbital part of the Inferior frontal gyrus, left triangular part of the inferior frontal gyrus, and left inferior temporal gyrus [4]. Till now, there is no treatment for this disease, but an early prediction of conversion to AD helps slow down the progression of dementia more effectively. Changes in the brain associated with AD begin 20 years before AD symptoms appear [1]. It is more helpful for patients to predict the progression of a patient from mild cognitive impairment (MCI) stage toward AD than distinguishing AD from health normal control (NC) (see Figure 1 [5]). MCI is the cognitive decline stage before AD immediately, where subjects in this stage suffer from mild cognitive ability loss but they are still able to do their daily activities without any sign of dementia. This stage is categorized into early MCI (eMCI) and late MCI (lMCI). Another way of categorizing subjects in this stage is to divide them into stable MCI (sMCI) and progressive MCI (pMCI) based on developing AD or staying stable in a follow-up period [4]. The development of AD always occurs in a time range of 6 to 36 months. However, diagnosis of Alzheimer's disease in its early stages is more challenging than in later ones, as distinguishing between sMCI and pMCI or lMCI and eMCI is very hard due to the high similarities between brain changes in the distinguished cases.

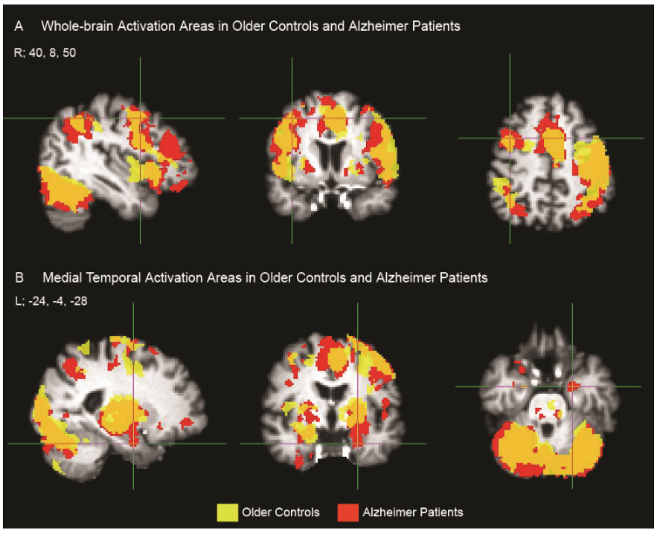
A Whole-brain Activation Areas in Older Controls and Alzheimer Patients
R; 40, 8, 50
B Medial Temporal Activation Areas in Older Controls and Alzheimer Patients
L; -24, -4, -28
Older Controls
Alzheimer Patients

Figure 1. Demonstrates the increase in fMRI activity in neocortical (A) and medial temporal (B) brain areas of Alzheimer's disease (in red) relative to healthy older control subjects (in yellow) [5].

The human brain consists of multiple regions that interact with each other, neurodegenerative diseases such as AD are associated with a change in different brain regions' activity as well as the interaction patterns between different brain regions known as functional connectivity networks [6]. Different from structural magnetic resonance imaging (sMRI) (known traditionally as MRI) which scans the brain's anatomical structure, functional magnetic resonance imaging (fMRI) scans brain activity by capturing the blood oxygenation level-dependent (BOLD) signal as the indicator for brain activity. The information on the brain's functional disturbance provided by fMRI proved to be very helpful with the diagnosis of several brain diseases at an early stage, as brain activity changes happening as a result of AD or any of its previous cognitive decline stages happen earlier than structural (anatomical) changes scanned by sMRI [7]. There are two kinds of fMRI: resting-state fMRI (rs-fMRI) and task-based fMRI. In rs-fMRI, the session is recorded while the subject is at rest. While in task-based fMRI, the subject performs some cognitive tasks during the session. The majority of studies use rs-fMRI for automated AD diagnosis due to being easier to obtain as the subjects are not performing any task which also reduces the motion artifacts making the pre-processing of fMRI data less complicated. Still, fMRI is not yet reliable in the clinical diagnosis of AD due to two major problems. First, sophisticated statistical analysis is required for useful information extraction from raw fMRI. Second, the difficulty of fMRI visualization due to its complicated data structure. Here comes the role of applying deep learning to fMRI images for automated medical diagnosis as a step toward making fMRI applicable in clinical practice. An fMRI session is captured as a time series of brain volumes. In the literature, very few studies investigated applying deep learning in fMRI-based AD diagnosis. The brain activity information recorded by fMRI can be complementary with other neuroimaging biomarkers such as MRI and positron emission tomography (PET) for more accurate diagnosis. However, it is still early to rely on fMRI for clinical diagnosis of AD. Several challenges need to be overcome to make fMRI usage in automated diagnosis reliable and trustable. An fMRI session is captured as a time series of brain volumes. Hence, one of the main challenges is the overfitting problem that goes back to training complex 4D models on fMRI datasets that are always available with small sizes to capture the spatial and temporal features of the fMRI time-series [7].

1.2. Related Work and Contribution

For more than a decade, various machine learning methods have been proposed for automated AD diagnosis based on rs-fMRI data. The first conventional approach introduced in the literature was using Pearson's correlation coefficients between pairs of brain regions to construct a functional connectivity (FC) network (matrix) representing the temporal functional relationships between different brain regions from an fMRI scan after using regions of interest (ROI) template (e.g., AAL) to extract the regions between which the connectivity network is of the analysis interest. Each element in this matrix denotes the Pearson correlation coefficient between the time series of a pair of ROIs. Then, using a feature extractor to extract from the FC matrices the discriminative features to be used in training a machine learning model for classifying different AD stages. For example, Jie et al. applied a graph-kernel method to FC networks to measure the topological similarity between different subjects as a feature extractor to train an SVM for classification [8]. Different from previous studies that used a single SVM, Bi et al. proposed a random SVM cluster to overcome the information loss problem encountered with a single SVM [9]. Each SVM in the cluster is built by randomly selecting a set of samples and their FC features achieving higher feature diversity. To categorize a sample, the prediction by the majority of the multiple SVMs is used. Jie et al. constructed dynamic connectivity networks (DCNs) and extracted both temporal variability features (e.g., the temporal correlation between brain regions) and spatial variability features (e.g., spatial variability within a specific brain region) neglected by most of the studies that existed before [6]. They used a manifold regularized multi-task feature learning model to extract the most important features on which a multi-kernel SVM was trained for disease classification.

However, the aforementioned methods have the disadvantages of feature extraction and the classifier's training being performed as two separate processes in addition to performing feature extraction in a handcrafted manner, which degrades the classification performance. Thanks to deep learning, both processes can be automatically combined into a single training process. Hence, the literature shifted toward applying deep learning methods, particularly convolutional neural networks (CNNs) to FC networks directly leading to significantly better classification accuracy, as feature extraction is directed by the classification feedback through the training process. Also, deep neural networks are capable of learning more hidden disease-related patterns. For example, He et al. constructed an FC matrix by calculating the time series correlation coefficient between each pair of brain regions based on both the variance of the mean value of all voxel time series in the two brain regions over time and the covariance of the time series mean value of the two brain regions [10]. Then, feeding the adjacency matrix of each subject into a simplified instance of 3D-MobileNet [11] architecture. In another study, Duc et al. applied ICA to fMRI data to generate an FC matrix representing the time-series correlations between 16 different independent components (brain regions) for each subject [12]. Then, from these matrices, 3D ICA feature maps were obtained using dual regression and fed into a 3D VGG-Net model. With another approach to decrease the fMRI dataset's small size overfitting effect, Wang et al. utilized PCANet [13] on FC matrices [14]. The choice of PCA was due to its unsupervised nature that eliminates its need for feedback adjustment parameters, making it an efficient choice for small datasets. Lin et al. introduced their end-to-end CRNN framework where they combined a CNN with an LSTM to model the 4D information [15]. They applied CNNs to dynamic FC (dFC) networks for extracting the temporal features based on which an LSTM was used to model the sequential information of the dFC networks. To perform the spatiotemporal analysis avoiding the 4D information complexity and model the 4D features at the same time, Jia et al. used mReHo transformation [16] that represents the whole time series of volumes as a single volume [17]. In the study, a 3D-PCANet was trained on the mReHo volumes as the classifier model. Due to the small size of the fMRI dataset, a disparity in sample distribution is encountered causing overfitting. As a solution for this problem, graph autoencoder (GAE) was applied over a node-edge connectivity graph that represents each node's (brain region) connectivity profile over the time series [18]. Graph theory helps to model the spatiotemporal pairwise correlation between brain regions by encoding the nodes and edges of a graph into a latent vector space to capture hidden information despite the disparity of a dataset. The GAE was trained as the generator part of a generative adversarial network (GAN) [19] that reconstructs the brain connectivity graphs to enforce the GAE to better learn the connectivity graph features benefiting from the discriminator's loss. In [20], a 3D-CNN was trained on fMRI images, and the temporal dimension was neglected. While Wang et al. applied a 2D-CNN to ROIs and then passed the extracted spatial features from sequential fMRI volumes to an LSTM to model the temporal features, they neglected the third spatial dimension [21].

Despite the classification performance improvement achieved by applying CNNs to FC networks or mReHo representations, both representations have the drawback of losing some spatial and temporal information. Also, applying 3D-CNNs to the fMRI session's volumes separately neglects the temporal dimension totally, and applying 2D-CNNs to fMRI volumes' slices neglects the third spatial dimension resulting in information loss in both cases. To preserve both spatial and temporal information, Li et al. introduced a concatenation of a 3D-CNN with an LSTM applied directly to fMRI scans [22]. In this model, the spatial feature maps generated by the 3D-CNN from all volumes along the fMRI time-series are passed sequentially to the LSTM to model the temporal relationship along the fMRI time-series. To model the 4D information, eliminating the complexity of the CNN-RNN combination, Parmar et al. designed a 3D CNN with the first two convolutional layers of a (1 × 1 × 1) kernel to learn the temporal features hierarchically [7], then the spatial features of the volumes in the time-series were learned through the following layers. Although CNN-based methods achieved efficient results being the best in fMRI-based AD diagnosis, an overfitting problem still exists, especially when applying CNNs to small-size datasets, which is always the case for most of the medical datasets, including fMRIs.

In this study, to develop an effective method for early-stage diagnosis of AD we introduce a CapsNet-RNN model to efficiently learn the spatiotemporal features of fMRI data for AD diagnosis tackling the small-size dataset problem that causes overfitting. We propose to utilize an enhanced version of CNNs called Capsule Network (CapsNet) introduced by Sabour et al. [23] as a solution to the overfitting problem encountered with traditional CNNs, combined with a recurrent neural network (RNN) to model the 4D information of fMRI sessions. In other words, to address the issue of overfitting in medical datasets, such as fMRIs, the Capsule Network (CapsNet), a modified version of traditional CNNs created specifically for this use, was combined with an RNN to simulate the spatiotemporal characteristics of fMRI data. The spatial characteristics of every volume in an fMRI time-series were extracted using CapsNet to create a feature vector. Next, to simulate the temporal features along the fMRI volume time-series, the vectors of the time-series are successively input to an RNN. CapsNet was introduced with two main modifications to solve two major problems in traditional CNNs. First, it replaces the scalar feature detectors lacking efficient representation of the spatial correlation between different entities (shapes) produced by a conventional CNN with the capsule concept, dividing output feature maps into several groups and viewing each group of the output feature maps as a grid of capsules. Each capsule outputs a vector and the length of a vector indicates the probability that the entity (shape) represented by its capsule exists at this capsule's location in an image, while the orientation of this vector encodes different properties of the entity such as pose, position, size, orientation, deformation, and texture. Second, instead of the pooling operation that results in information loss, it applies a routing-by-agreement algorithm to alleviate this problem and model the relationship between lower-level entities constructing higher-level ones in the image features hierarchy. In our proposed CapsNet-RNN model, the CapsNet outputs a feature vector representing the spatial features of each volume in the fMRI time series. The vector of each volume is then given to an RNN to model the temporal dependency information between the fMRI volume sequences. The proposed model is able to achieve 94.5% and 61.8% accuracy for the AD vs. NC and lMCI vs. eMCI classification tasks, respectively.

2. Proposed Model

In this section, firstly, CapsNet and RNN models are overviewed to comprehend the structure of the proposed model. Then, the architecture of the proposed model is explained.

2.1. CapsNet

CNNs have proven to be efficient with image classification, especially in the medical domain. However, CNNs still suffer from some drawbacks that are more obvious with small datasets, which is almost the case for medical image datasets. CNNs suffer from two major drawbacks [23]: first, they use scalar feature detectors, which do not efficiently capture the spatial relationships between different entities. This makes CNNs less robust to affine transformations, which means that a CNN needs to be trained on a big dataset containing most, if not all, possible transformations, while medical datasets like fMRIs are usually of small size with respect to a deep learning model's number of parameters, which causes the trained model to overfit the dataset. The second drawback is performing pooling after each convolutional layer to reduce the number of parameters for achieving a lower computational complexity. This reduction of parameters results in precise position information loss. To solve these issues, CapsNet applies two concepts: vector-output capsules and routing-by-agreement as a replacement for scalar feature detectors and max-pooling operation, respectively.

2.1.1. Vector-output capsules

A CapsNet can have one or more convolutional capsule layers; each layer performs convolution on the resulting convolutional capsules (feature maps) from the previous layer. Convolution is used to maintain the shared weight property of CNNs (knowledge replication across space) by learning the reusable feature detectors across the entire image. Each convolutional capsule represents a shape entity, where a convolutional capsule is a group of *N* feature maps; the number of these feature maps is referred to as the number of atoms. This group of *N* feature maps can be viewed as a grid of capsules, each capsule outputting a vector. At each location on this grid, there is a capsule's *N*-dimensional output vector that consists of the feature detectors at the same location across this group of *N* feature maps forming the convolutional capsule. The length of a capsule's output vector represents the probability of the existence of the entity (e.g., rectangle, triangle, etc.) represented by its convolutional capsule at this capsule's location. While the orientation of the capsule's output vector encodes various properties of the entity it represents, including different types of instantiation parameters such as pose, position, size, orientation, deformation, and texture. The capsules of each convolutional capsule layer represent a set of entities at a certain level in a hierarchy of shape complexity, where the entities in a higher layer are constituents of entities from the layer below.

2.1.2. Routing-by-agreement

To assign parts to the wholes, an algorithm called routing-by-agreement is used to couple each capsule representing an entity in a capsule layer with the capsules of the entities (parts) in the layer below. This routing-by-agreement process iteratively takes place between each capsule layer and its previous layer for a defined number of iterations within a single training iteration for the whole network. There is a coupling coefficient between each capsule in the capsule layer and each capsule in the lower layer, indicating the extent to which a lower-level capsule (entity) is probably a part of a higher-level capsule. The input to a capsule $\mathbf{s}_j$ is calculated as a weighted sum over all "prediction vectors" $\hat{\mathbf{u}}_{j|i}$ for this capsule. These prediction vectors are calculated by multiplying the output vectors $\mathbf{u}_i$ of all capsules in the layer below by a weight matrix $\mathbf{W}_{ij}$ that is learned through the training process as in (1) [23]:

$$\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}, \qquad \hat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i. \tag{1}$$

The following squashing function is applied to capsule input vectors to shrink their values slightly below 1 as they are used as probability values as follows in (2):

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}, \tag{2}$$

where a capsule j receives an input $\mathbf{s}_j$ and outputs a vector $\mathbf{v}_j$. The summation of prediction vectors $\hat{\mathbf{u}}_{j|i}$ is weighted by coupling coefficients c_{ij} *and* the coupling coefficients between capsule i and all the capsules in the layer above sum to 1. A coupling coefficient c_{ij} is computed by a "routing softmax" whose initial logits are log prior probabilities b_{ij} that a capsule i should be coupled to capsule j as in (3):

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}. \tag{3}$$

The logits b_{ij} are initially set to zero, and their optimal values are learned iteratively through the dynamic routing-by-agreement process based on measuring the agreement between a capsule output $\mathbf{v}_j$ and the prediction vector $\hat{\mathbf{u}}_{j|i}$ from capsule i. The agreement is calculated as the scalar product $a_{ij} = \mathbf{v}_j \cdot \hat{\mathbf{u}}_{j|i}$, then added to the current logit b_{ij} to calculate the new logits and coupling coefficient values for the following iteration and so on. The routing-by-agreement algorithm is demonstrated in Table 1.

Table 1. Routing algorithm.

Routing algorithm

```
ROUTING($\widehat{U}_{j|i}, r, l$)
 for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l + 1)$: $b_{ij} \leftarrow 0$.
 for $r$ iterations do
     for all capsule $i$ in layer $l$: $C_i \leftarrow \text{softmax}(b_i)$        ▶softmax computes Eq. 3
     for all capsule $j$ in layer $(l + 1)$: $S_j \leftarrow \sum_i C_{ij} \widehat{U}_{j|i}$
     for all capsule $j$ in layer $(l + 1)$: $V_j \leftarrow \text{squash}(S_j)$        ▶squash computes Eq. 1
     for all capsule $i$ in layer $l$ and capsule $j$ in layer $(l + 1)$: $b_{ij} \leftarrow b_{i|j} + \widehat{U}_{j|j}. V_j$
 return $V_j$
```

2.2 Recurrent Neural Networks

RNNs are used to label, classify, or generate sequences. Data is input to an RNN as a sequence of feature vectors [24]. An RNN can be used to predict a class for each feature vector in the sequence or to predict a class for the whole sequence. RNNs are used with any kind of data that contains sequential dependencies. Therefore, RNNs are used for applications like natural language processing and forecasting time series.

2.3 Proposed Model

The proposed model consists of two parts: a CapsNet to capture the spatial features of volumes in the fMRI time series, followed by an RNN to capture the temporal features across the time series. CapsNet first receives an fMRI time series sample of five volumes to extract the spatial features from each volume as a 32D feature vector. Then, these five vectors are fed sequentially into the RNN to model the temporal features of this volume sequence, followed by a fully connected layer for final prediction.

2.3.1. CapsNet Architecture

We designed our CapsNet architecture to start with a convolutional layer with a 3D kernel of size 5, stride 1, valid (zero) padding, and 16 output channels, followed by 2 other convolutional layers with a 3D kernel of size 5, stride 2, valid padding, 32 and 64 output channels, respectively. The convolutional layers are followed by 3 convolutional capsule layers; the output of each is 32 convolutional capsules, each consisting of 8 atoms. Note that the size of a convolutional capsule's grid in a convolutional capsule layer is the size of the feature maps generated from the convolution operation in this layer, and the number of capsules in a convolutional capsule's grid is equal to the size of this grid. Each convolutional capsule layer has a 3D kernel of size 3, stride 2, and padding 1. The architecture of the CapsNet is illustrated in Figure 2. It should be noted that the number of convolutional capsules from the previous layer to the first convolutional capsule layer is equal to 1, as the previous one is a traditional convolutional layer.

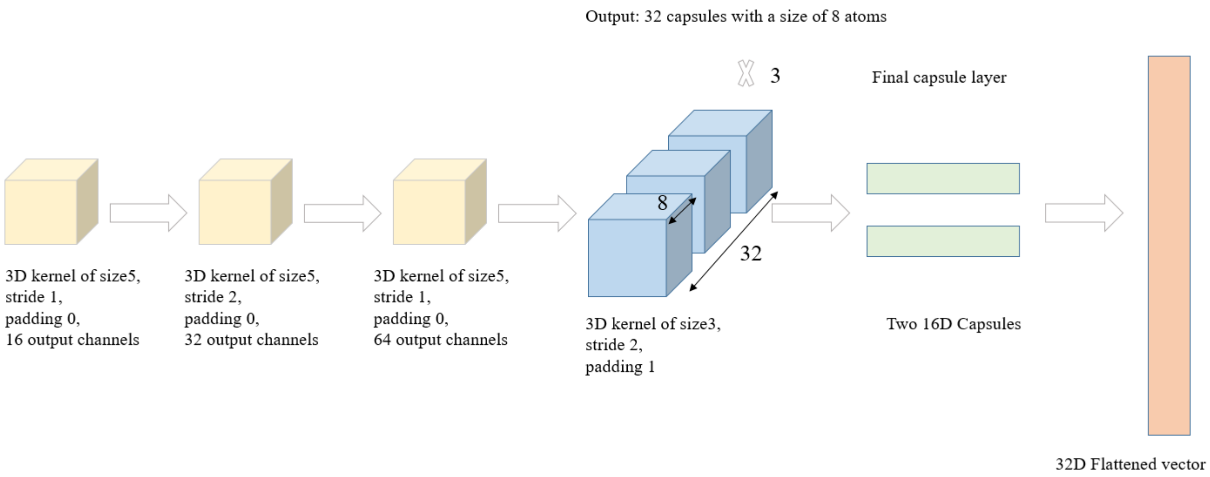

Figure 2. The CapsNet architecture used in the proposed model consisting of a convolutional layer with a 3D kernel of size 5, stride 1, valid (zero) padding, and 16 output channels, followed by 2 other convolutional layers with a 3D kernel of size 5, stride 2, valid padding, 32 and 64 output channels, followed by 3 convolutional capsule layers.

Between every two capsules in two consecutive convolutional capsule layers, there should be a weight matrix, as explained in Section 2.1.2. Thus, it is expected to have a number of weight matrices equal to (number of convolutional capsules in the lower layer × number of convolutional capsules in the higher layer × grid dimension of the lower convolutional capsule layer × grid dimension of the higher convolutional capsule layer), which means an explosive number of parameters. To reduce the number of parameters and replicate knowledge across space, an approach is applied. Based on this approach, a set of filters convolving the convolutional capsules in the lower layer was used, like traditional convolution, which replaces a set of pixels equal to the kernel size by a pixel, a set of capsules is replaced by a capsule. Thus, the weights between the capsules of two convolutional capsule (convolutional capsule) layers are the weights of convolutional filters (playing the role of weight matrices $\mathbf{W}_{ij}$ mentioned in Section 2.1.2).

The final layer in the CapsNet stage of the proposed model is composed of two 16D capsules; each of these two capsules outputs a 16D vector, where two vectors represent the spatial features of an fMRI volume, not acting as class capsules like the original CapsNet architecture introduced in [23]. Here, the capsules in the last convolutional capsule layer use a separate weight matrix $\mathbf{W}_{ij}$ between each capsule i in this layer and each capsule j in the final layer of CapsNet. No convolution is applied between these two layers.

2.3.2 Recurrent Neural Network Architecture

In our model, we used an RNN to model the fMRI time series properties. Therefore, the two output vectors by CapsNet representing the spatial features of each volume are flattened as a single 32D vector, then all the flattened vectors are fed as a sequence to the RNN for modeling the temporal features of the volumes' time series. In the RNN architecture, there is a single layer with 64 hidden cells. Finally, the output of the RNN is passed to a fully connected layer with a single neuron, after which a sigmoid function is applied to perform the binary classification. Overall, the architecture of the proposed model based on CapsNet and RNN is illustrated in Figure 3.

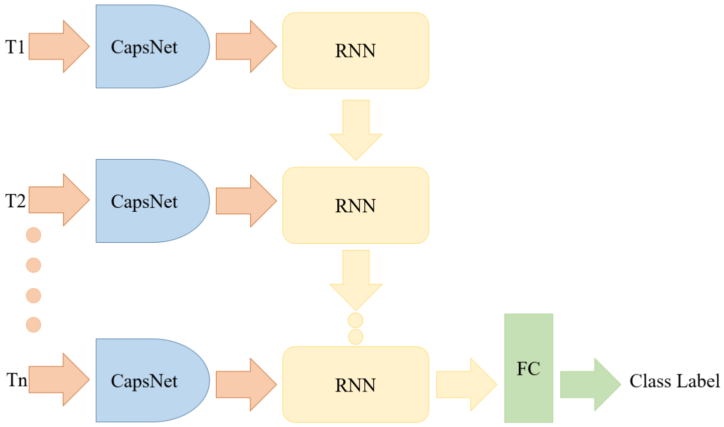

Figure 3. The architecture of the proposed model with an RNN to model the fMRI time series properties.

3. Experiments

3.1. Dataset and Pre-processing

Experiments were performed in order to evaluate the efficiency of the proposed model. In the experiments, rs-fMRI data acquired from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database [25] was used. The rs-fMRI dataset contains four classes of a total of 147 subjects, including 34 AD subjects, 34 NC subjects, and 80 MCI subjects divided into 40 eMCI and 40 lMCI. For each subject, 140 volumes were acquired in a session, with each volume consisting of 36 − 48 slices. Each slice is of size 64 × 64, slice thickness is 3.1 mm, TE (echo time) is 30 ms, and TR (repetition time) is 2200 – 3100 ms. The rs-fMRI dataset was pre-processed using the SPM 12 toolbox [26] and MATLAB 2022b. The pre-processing steps included removing the first three volumes for magnetization equilibrium, head motion correction, slice timing correction, and spatial smoothing by a Gaussian kernel with a full-width-at-half-maximum (FWHM) of 6 mm. Finally, fMRIs were co-registered with T1-weighted MRIs of the same subjects to normalize the fMRIs onto the standard 152 Montreal Neurological Institute (152 MNI) space. By the end of this process, the dimension of fMRI volumes is changed from 64 × 64 × 48 to 61 × 73 × 61 due to the deformations applied by the spatial normalization module to register the images into the standard space.

3.1. Implementation Details

The Python-based Pytorch framework was used to implement and train the proposed model. Besides, Google Colab Pro was used to perform the experiments, utilizing the NVIDIA A-100 GPU with 40 GB of RAM and 83.5 GB of CPU RAM. The proposed model was trained for two binary classification tasks: AD vs. NC and lMCI vs. eMCI classifications. Binary cross-entropy was used as a loss function for the proposed model. The model was trained using the Adam optimizer with a learning rate of 1e-6 and L2 regularization to prevent overfitting with a regularization coefficient of 0.1. The training was completed in 20 and 15 epochs (to avoid overfitting) for the AD vs. NC and lMCI vs. eMCI tasks, respectively. Moreover, the batch size was set to 2 for both classification tasks. The proposed model was trained for nearly 1 hour and 10 minutes. It should be noted that the number of parameters in our model is 4.4 M parameters. Furthermore, the GPU RAM consumption was 1.9 GB, and the CPU RAM consumption used mainly for holding the dataset was 49.5 GB and 43.9 GB for AD vs. NC and lMCI vs. eMCI tasks, respectively.

Before the implementation, the dataset was divided into three categories: 70% as training data, 10% as validation data, and the remaining 20% as testing data. The proposed model was trained on 34 AD, 34 NC, 40 lMCI, and 40 eMCI subjects, taking a maximum of three sessions and two sessions from each subject in AD vs. NC and lMCI vs. eMCI, respectively, due to memory constraints. Following an approach applied in [7], we obtained the first 100 volumes from each session after the pre-processing stage and divided these 100 volumes into 20 samples to have 5-volume time-series samples. Obtaining five samples from a single session increases the number of trainable samples, leading to a higher generalization of the proposed model. Thus, 1680 AD, 1680 NC, 1460 lMCI, and 1460 eMCI samples were obtained to carry out the experiments.

3.2 Results

3.2.1. Classification performance

The classification accuracy achieved by the proposed model for the two binary classification tasks is given in Table 2. For the AD vs. NC classification task, it is clear that the proposed model achieved the accuracy at 100%, 100%, and 94.5% for training, validation, and testing datasets, respectively. For the lMCI vs. eMCI task, it achieved the accuracy at 99%, 64.3%, and 61.8% accuracies for training, validation, and testing datasets, respectively.

The loss and accuracy curves for the AD vs. NC task are plotted in Figure 4 and 5, respectively, and for the lMC vs. eMCI task in Figure 6 and 7, respectively.

Table 2. Classification accuracy of the proposed model.

Accuracy	AD vs. NC	lMCI vs. eMCI
Training	100%	99%
Validation	100%	64.3%
Test	94.5%	61.8%

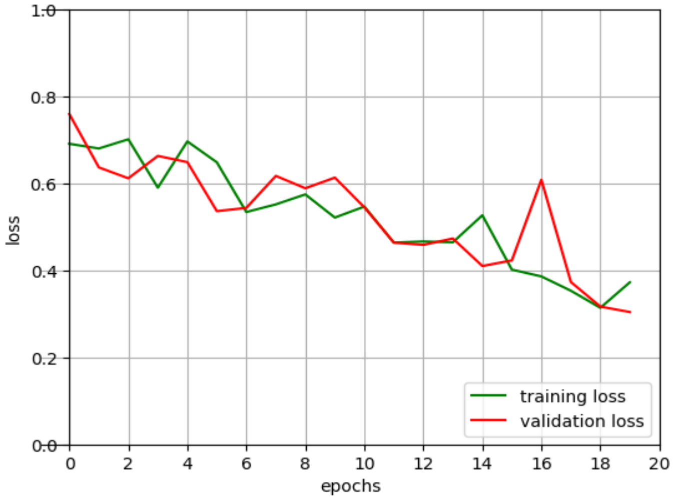

Figure 4. Loss curves of AD vs. NC.

3.2.2. Comparison with State-of-the-Art Methods

In this section, the accuracy of the proposed method is compared with several state-of-the-art methods proposed in the literature. We can see from Table 3 that the proposed model performs comparably well in terms of testing accuracy to the recent studies for the AD vs. NC task. Besides, it can be seen that nearly all the models reach high accuracies with the AD vs. NC task, while there is still a gap between the accuracies achieved by our study and studies [6], [21] when performing lMCI vs. eMCI classification and those achieved by nearly all studies in the AD vs. NC task. Thus, the lMCI vs. eMCI task and its similar sMCI vs. pMCI are reported to be challenging tasks in the literature [4]. Hence, the authors are planning to consider enhancing the accuracy of the lMCI vs. eMCI task in the near future.

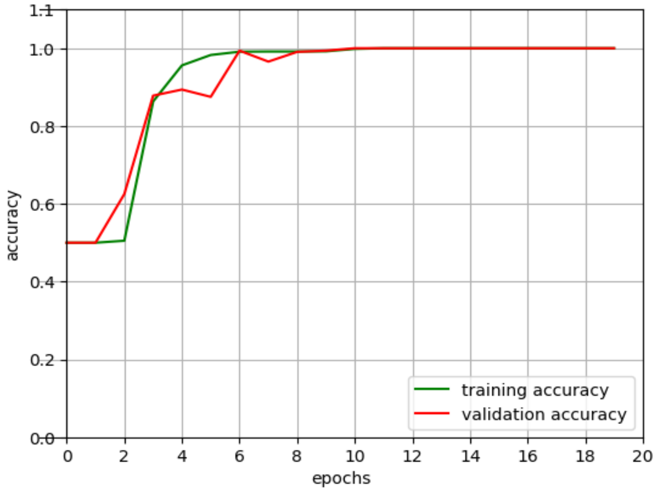

Figure 5. Accuracy curves of AD vs. NC.

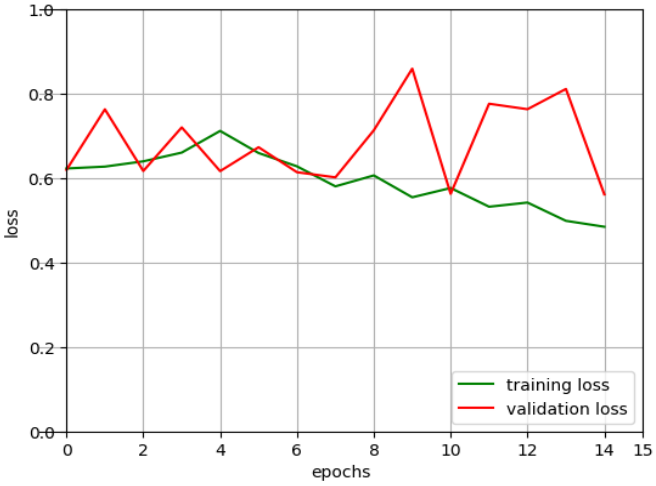

Figure 6. Loss curves of lMCI vs. eMCI.

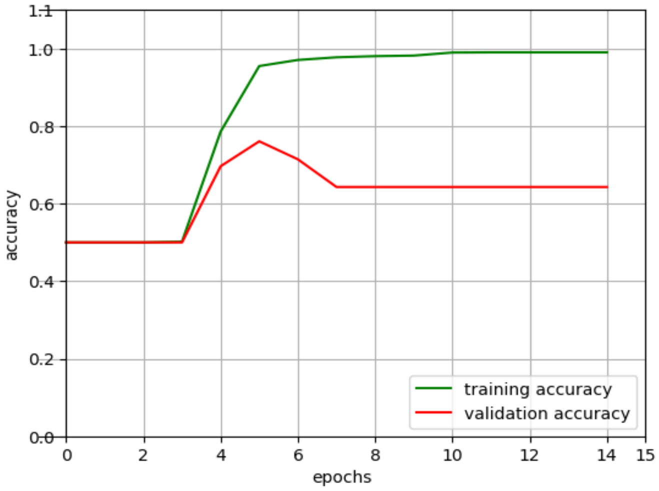

Figure 7. Accuracy curves of lMCI vs. eMCI.

Table 3. Comparison with other studies.

Study	Method	AD vs. NC	lMCI vs. eMCI	pMCI vs. sMCI	Multi-Class
He et al. [10]	2D-CNN applied to FC matrices.	93%	N/A	N/A	N/A
Duc et al. [12]	3D-CNN applied to 3D-ICA feature maps based on FC networks.	85.27%	N/A	N/A	N/A
Jie et al. [6]	Multi-task feature learning+SVM applied to FC networks.	N/A	78.8%	N/A	N/A
Wang et al. [14]	PCANet applied to FC matrices.	88%	100%	N/A	N/A
Jia et al. [17]	PCANet applied to mReHO transformation images.	80%	N/A	N/A	N/A
Mirakhorli et al [18]	Graph-CNN applied to brain connectivity node-edge graph.	97.1%	N/A	N/A	N/A
Lin et al. [15]	CNN+LSTM applied to FC networks	92.8%	N/A	N/A	AD vs eMCI vs lMCI vs NC: 61.7%
Wang et al. [21]	2D-CNN+LSTM applied to overlapping windows of ROIs.	90.28%	79.36%	N/A	AD vs. MCI vs. NC: 71.76% AD vs. eMCI vs. lMCI vs. NC: 60.67%
Li et al. [22]	3D-CNN+LSTM applied to fMRI volumes time-series.	97.37%	N/A	N/A	AD vs MCI vs NC: 89.47%
Parmar et al. [7]	A special configuration of 3D-CNN to model the 4D information applied to fMRI volumes time-series.	N/A	N/A	N/A	AD vs. eMCI vs. lMCI vs. NC: 93%
Our study	CapsNet +RNN applied to fMRI volumes time-series.	94.5%	61.8%	N/A	N/A

As discussed earlier, there are two aims behind combining a CapsNet with an RNN in the same model. The first is to model the 4D information by applying the model directly to the time series of fMRI volumes in order to avoid information loss associated with applying CNNs to FC networks or mReHo transformation images, and the second is to overcome the overfitting problem that occurs with traditional CNNs that was replaced with CapsNet. From the comparison results, it has been observed that the proposed model and the studies introduced in [7] and [22], which model the 4D information from the fMRI volume sequence directly, were able to achieve better accuracies than the models introduced in [10], [12], [14], [15] (which apply CNNs to FC networks), and [17] (that apply CNNs to mReHO images). Moreover, despite CapsNet being designed to achieve better generalization than traditional CNNs, the method which applies CNNs with LSTM directly to fMRI volumes [22] still achieves better accuracy than our model. Hence, we concluded that some enhancement is required in the pre-processing stage to improve the performance of our model, which can be considered as a future work.

4. Discussion

The proposed CapsNet-RNN-based model exhibits a high capability of generalization and learning more hidden patterns of interest from a small dataset and in a relatively small number of epochs. In addition, it efficiently models the 4D information of fMRI sessions, which is unique to fMRI different from other neuroimaging modalities.

Medical datasets generally suffer from being of small size including brain neuroimaging datasets. This is due to several reasons, such as the difficulty of acquiring medical datasets from a wide range of subjects and privacy and data protection concerns. Here, the role of CapsNet comes as a solution to one of the CNNs' drawbacks that becomes more obvious with smaller datasets. CNNs suffer from a lack of capturing sufficient affine transformation features, as discussed earlier resulting in an overfitting problem. CapsNet with its representation of shape entities as vectors and applying the routing-by-agreement algorithm can efficiently learn the relationship features of assigning parts to wholes, which alleviates the need for big datasets to learn all possible affine transformations as in the case of CNNs.

Regarding fMRI data, the CapsNet-RNN models both brain activity dynamic changes within a specific region and dynamic FC between different regions across the fMRI time series. It does not need to construct FC matrices as a pre-processing step, where it learns the spatiotemporal features entirely through the training process performed directly on fMRI images. Thus, it avoids the information loss associated with constructing FC networks or just taking into account the spatial features and neglecting the temporal ones.

This relatively high accuracy reached in less than 20 epochs with the AD vs. NC task reveals the powerfulness of CapsNet to generalize and converge in a relatively small number of epochs with small-size datasets. Similarly, CapsNet converged to the desired classification accuracy in 10 epochs in another study utilizing it in a multi-class brain tumor classification [27]. The promising results of CapsNet in the medical domain and other various domains [28] show that CapsNet is revolutionizing computer-vision-related applications, thus it can be helpful with several medical tasks such as diagnosis, prognosis, and segmentation.

5. Conclusion and Future Work

To capture the 4D information of the rs-fMRI time-series data for AD diagnosis, an end-to-end deep learning model (CapsNet-RNN) is proposed in this study. Specifically, an RNN was concatenated with a version of standard CNNs known as CapsNet, created for this purpose, to represent the spatiotemporal aspects of fMRI data, therefore mitigating the overfitting issue associated with medical datasets, including fMRIs. The spatial characteristics of every volume in an fMRI time series were extracted as a feature vector using CapsNet. Subsequently, an RNN is trained with the time series vectors in order to model the temporal properties along the fMRI volume time series. The robustness of CapsNet to obtain a reasonable generalization level on limited datasets is demonstrated by the reasonably high accuracy (94.5%) reached in fewer than 20 epochs with the AD vs. NC task.

Although the results of applying the suggested model to fMRI data for AD diagnosis have been encouraging, the pre-processing step still needs to be revised. Furthermore, it is also essential to enhance the proposed model to achieve closer accuracy achieved by well-known models for the AD vs. NC task and to attain similar accuracy for the AD vs. NC task along with more challenging tasks like the lMCI vs. eMCI task. As a future work, combining different neuroimaging biomarkers, such as sMRI and PET with fMRI using CapsNets for AD diagnosis needs to be investigated. Combining multi-imaging modalities can lead to better diagnosis accuracy in clinical practice, as different modalities can provide complementary information to each other. Moreover, only binary classification was used in this investigation. The suggested model is anticipated to be used in the near future for multi-class classification of various phases of AD or possibly for differentiating AD from other types of dementia.

Acknowledgements

Publicly available datasets from Alzheimer's Disease Neuroimaging Initiative (ADNI) database were analyzed in this study. Data accessible from: Alzheimer's Disease Neuroimaging Initiative (ADNI) database (https://adni.loni.usc.edu).

References

[1] Alzheimer's Association Report. 2017 Alzheimer's disease facts and figures. Alzheimer's & Dementia 2017; 13(4): 325-373.

[2] Haux R. Health information systems - past, present, future. Int J Med Inform 2006; 75(3-4): 268-281.

[3] Janghel RR, Rathore YK. Deep convolution neural network based system for early diagnosis of alzheimer's disease. IRBM 2021; 42(4): 258-267.

[4] Ebrahimighahnavieh MA, Luo S, Chiong R. Deep learning to detect Alzheimer's disease from neuroimaging: a systematic literature review. Computer Methods and Programs in Biomedicine 2020; 187: 105242.

[5] Kivistö J, Soininen H, Pihlajamaki M. Functional MRI in Alzheimer's Disease. Advanced Brain Neuroimaging Topics in Health and Disease - Methods and Applications. InTech; 2014.

[6] Jie B, Liu M, Shen D. Integration of temporal and spatial properties of dynamic connectivity networks for automatic diagnosis of brain disease. Med Image Anal 2018; 47:81-94.

[7] Parmar H, Nutter B, Long R, Antani S, Mitra S. Spatiotemporal feature extraction and classification of alzheimer's disease using deep learning 3D-CNN for fMRI data. Journal of Medical Imaging 2020; 7(5): 056001.

[8] Jie B, Zhang D, Wee CY, Shen D. Topological graph kernel on multiple thresholded functional connectivity networks for mild cognitive impairment classification. Hum Brain Mapp 2014; 35(7):2876-2897.

[9] Bi XA, Shu Q, Sun Q, Xu Q. Random support vector machine cluster analysis of resting-state fMRI in alzheimer's disease. PLoS One 2018; 13(3): e0194479.

[10] He Y, Wu J, Zhou L, Chen Y, Li F, Qian H. Quantification of cognitive function in alzheimer's disease based on deep learning. Front Neurosci 2021; 15: 651920.

[11] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv 2017; 1704.04861.

[12] Duc NT, Ryu S, Qureshi MNI, Choi M, Lee KH, Lee B. 3D-deep learning based automatic diagnosis of alzheimer's disease with joint mmse prediction using resting-state fMRI. Neuroinformatics 2020; 18(1), 71-86.

[13] Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y. PCANet: a simple deep learning baseline for ımage classification? IEEE Transactions on Image Processing 2015; 24(12): 5017-5032.

[14] Wang Y, Liu X, Yu C. Assisted diagnosis of alzheimer's disease based on deep learning and multimodal feature fusion. Complexity 2021; 2021: 6626728.

[15] Lin K, Jie P, Dong P, Ding X, Bian W, Liu M. Convolutional recurrent neural network for dynamic functional mrı analysis and brain disease ıdentification. Front Neurosci 2022; 16: 933660.

[16] Jiang L, Zuo XN. Regional homogeneity: a multimodal, multiscale neuroimaging marker of the human connectome. Neuroscientist 2016; 22(5): 486-505.

[17] Jia H, Lao H. Deep learning and multimodal feature fusion for the aided diagnosis of alzheimer's disease. Neural Comput Appl 2022; 34(22): 19585-19598.

[18] Mirakhorli J, Amindavar H, Mirakhorli M. A new method to predict anomaly in brain network based on graph deep learning. Rev Neurosci 2020; 31(6): 681-689.

[19] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. ArXiv 2014; 1406.2661.

[20] Ghafoori S, Shalbaf A. Predicting conversion from MCI to AD by integration of rs-fMRI and clinical information using 3D-convolutional neural network. Int J Comput Assist Radiol Surg 2022; 17(7): 1245-1255.

[21] Wang M, Lian C, Yao D, Zhang D, Liu M, Shen D. Spatial-temporal dependency modeling and network hub detection for functional MRI analysis via convolutional-recurrent network. IEEE Trans Biomed Eng 2020; 67(8): 2241-2252.

[22] Li W, Lin X, Chen X. Detecting alzheimer's disease based on 4D fMRI: an exploration under deep learning framework. Neurocomputing 2020; 388: 280-287.

[23] Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; 4-9 December 2017; Long Beach, California, USA: Curran Associates Inc. pp. 3859–3869.

[24] Schmidt RM, Recurrent neural networks (RNNs): a gentle introduction and overview. ArXiv 2019; 1912.05911.

[25] Mueller SG, Weiner MW, Thal LJ, Petersen RC, Jack CR, Jagust W, Trojanowski JQ, Toga AW, et al. Ways toward an early diagnosis in alzheimer's disease: the alzheimer's disease neuroimaging ınitiative (ADNI). Alzheimers Dement. 2005; 1(1): 55-66.

[26] The FIL Methods Group and honorary members, SPM12 Manual. Functional Imaging Laboratory, Institute of Neurology, UCL, 2015, http://www.fil.ion.ucl.ac.uk/spm/doc/manual.pdf.

[27] Afshar P, Mohammadi A, Plataniotis KN. Brain tumor type classification via capsule networks. In: 25th IEEE International Conference on Image Processing (ICIP): 07-10 October 2018; Athens, Greece: IEEE. pp. 2381-8549.

[28] Goceri E. Analysis of capsule networks for image classification. In: International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing: IADIS. pp. 53-60.